

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1613

**POMAGALO ZA UPRAVLJANJE  
SCRUM PROJEKTIMA**

Jan Kelemen

ZAGREB, lipanj 2018.

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1613

**POMAGALO ZA UPRAVLJANJE SCRUM  
PROJEKTIMA**

Jan Kelemen

ZAGREB, lipanj 2018.

Zagreb, 7. ožujka 2018.

Predmet: **Razvoj informacijskih sustava**

## DIPLOMSKI ZADATAK br. 1613

Pristupnik: **Jan Kelemen (0036479753)**  
Studij: Računarstvo  
Profil: Programsko inženjerstvo i informacijski sustavi

Zadatak: **Pomagalo za upravljanje Scrum projektima**

### Opis zadatka:

Proučiti dostupnu programsku podršku za praćenje projekata razvijanih Scrum metodologijom te ustanoviti ključne funkcionalnosti aplikacija ovog tipa. Oblikovati bazu podataka te nad njom izraditi programsko pomagalo koje će omogućiti evidenciju osoba i njihovih uloga na projektu, podršku sastancima, evidenciju artefakata, izvođenje sprintova i zadataka te vrednovanje rezultata i praćenje performansi. Ugraditi prikladno grafičko korisničko sučelje i izvješća. Ilustrirati i verificirati vlastito rješenje na primjeru reprezentativnog projekta. Cjeloviti tekst rada, dokumentirani programski kod te korištene komponente priložiti na optičkom disku.

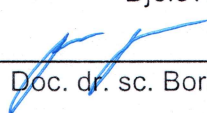
Zadatak uručen pristupniku: 16. ožujka 2018.

Rok za predaju rada: 29. lipnja 2018.

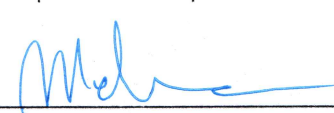
Mentor:

  
Prof. dr. sc. Krešimir Fertalj

Djelovođa:

  
Doc. dr. sc. Boris Milašinović

Predsjednik odbora za  
diplomski rad profila:

  
Izv. prof. dr. sc. Igor Mekterović



# Sadržaj

1.	Uvod .....	1
2.	Upravljanje projektima razvoja informacijskih sustava .....	2
2.1.	Scrum.....	3
2.1.1.	Scrum tim .....	3
2.1.2.	Životni ciklus razvoja sustava Scrum metodologijom .....	3
2.1.3.	Artefakti Scrum metodologije .....	5
3.	Programska podrška za upravljanje Scrum projektima .....	7
3.1.	ScrumDesk .....	7
3.2.	Jira .....	12
3.3.	Usporedba postojećih rješenja .....	14
4.	Zahtjevi na vlastito rješenje .....	15
4.1.	Konceptualni model podataka .....	16
5.	Arhitektura razvijenog sustava .....	19
5.1.	Struktura razvijenog rješenja .....	20
5.2.	Entiteti .....	21
5.3.	Slučajevi korištenja .....	24
5.4.	Perzistencija podataka .....	26
5.4.1.	Implementacija perzistencije podataka.....	27
5.5.	Prezentacija podataka .....	31
5.6.	Primjer izvršavanja jednog slučaja korištenja .....	33
6.	Opis funkcionalnosti razvijenog sustava .....	35
6.1.	Evidencija korisnika .....	35
6.2.	Evidencija timova .....	41
6.3.	Evidencija projekata .....	44
6.3.1.	Upravljanje radnim zadacima .....	47
6.3.2.	Upravljanje sprintevima .....	52
6.3.3.	Upravljanje sastancima.....	59
6.3.4.	Pregled dokumenata .....	60
6.4.	Kritički osvrt na razvijeni sustav .....	63
7.	Korištene tehnologije i alati.....	64
8.	Zaključak .....	65

Literatura .....	66
Pomagalo za upravljanje Scrum projektima.....	68
Scrum Project Management Tool.....	69
Dodatak: MongoDB baza podataka.....	70
Obnova baze podataka.....	70
Dodatak: Popis slika .....	71
Dodatak: Popis tablica .....	74
Dodatak: Popis isječaka koda .....	75

# 1. Uvod

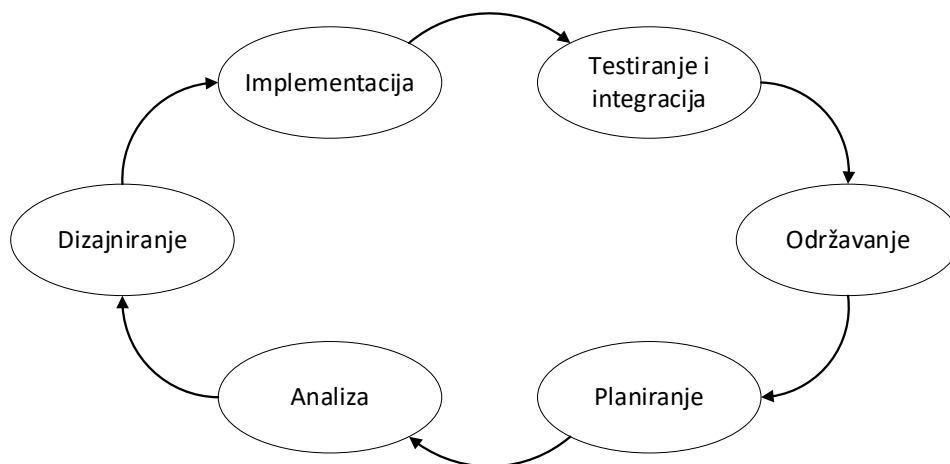
Tema ovog rada je pomagalo za upravljanje Scrum projektima, to jest, realizacija programske podrške koja će omogućiti praćenje razvoja sustava razvijanih Scrum metodologijom. Scrum je agilna metodologija razvoja kompleksnih produkata koja se najčešće koristi za razvoj informacijskih sustava [1]. U radu je dan pregled procesa razvoja programske podrške te detaljniji opis procesa razvoja sustava Scrum metodologijom. Analizirana su dva postojeća rješenja koja omogućuju upravljanje Scrum projektima. Definiran je skup funkcionalnosti kojeg bi takvo pomagalo trebalo podržavati iz kojeg je napravljen model podataka. Nakon specifikacije zahtjeva razvijenog alata, opisana je arhitektura i funkcionalnosti razvijenog alata. U zadnjem poglavlju rada navedene su tehnologije i alati korišteni za razvoj ovog alata.

## 2. Upravljanje projektima razvoja informacijskih sustava

Upravljanje projektima razvoja informacijskih sustava je proces praćenja svih aktivnosti projekta tokom cijelog životnog ciklusa razvoja sustava. Životni ciklus razvoja sustava (engl. systems development life cycle - SDLC) je naziv za općeniti proces razvoja sustava, koji opisuje proces za planiranje, analizu, izradu i održavanje informacijskih sustava. Ovaj ciklus najčešće se dijeli na sljedeće faze:

- planiranje – inicira se projekt te se određuje poslovni cilj i plan sustava,
- analiza – određuje se poslovni model i prijedlog sustava,
- oblikovanje – izrađuje se tehnička specifikacija sustava,
- izrada – izrađuje se funkcionalni sustav,
- testiranje i integracija – provjerava se da li izrađeni sustav zadovoljava prethodno napravljene specifikacije, te se vrši integracija s postojećim sustavima,
- održavanje – izrađeni sustav se nadgleda te se vrše potrebne prilagodbe.

Navedene faze nisu nužno slijedne i diskretne aktivnosti, već ovise o metodologiji razvoja sustava. Ukoliko se na projektu koristi klasični vodopadni model razvoja sustava koji se zasniva na slijednom napredovanju iz faze u fazu, one će biti slijedne i diskretne, dok se kod neke druge metodologije razvoja faze mogu preklapati [2] [3]. Slika 2.1 prikazuje tipičan odnos faza u životnom ciklusu razvoja sustava.



Slika 2.1 Životni ciklus razvoja sustava [4]



## 2.1. Scrum

Scrum je agilna metodologija inkrementalnog razvoja produkata, najčešće korištena za razvoj informacijskih sustava, koja omogućava brzu prilagodbu na promjene zahtjeva. Scrum se zasniva na tri stupa koji omogućuju kontrolu procesa razvoja:

- transparentnost – svi aspekti procesa moraju biti vidljivi onima odgovornima za rezultat procesa, odnosno postoji zajednička definicija što znači kada je neki posao završen,
- inspekcija – uključene osobe trebaju vršiti inspekciju nad artefaktima procesa i napretku prema cilju, kako bi na vrijeme mogle reagirati na neželjene promjene,
- adaptacija – ukoliko je inspekcijom utvrđeno da proces ili obavljen posao odskaka od prihvatljivih granica, potrebno je izvršiti adaptaciju [1].

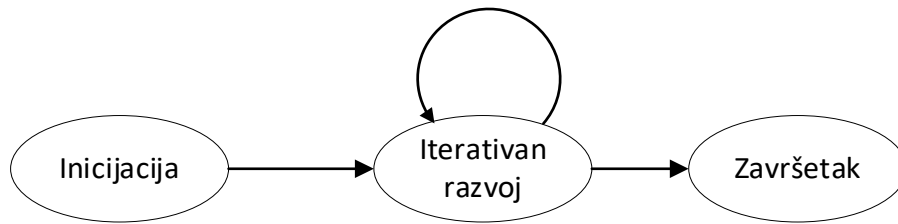
### 2.1.1. Scrum tim

Scrum tim je skupina ljudi koja sudjeluje u razvoju sustava Scrum metodologijom, a sastoji se od vlasnika proizvoda (engl. product owner), razvojnog tima (engl. development team) i od Scrum majstora (engl. Scrum master). Vlasnik proizvoda zadužen je za maksimizaciju vrijednosti posla koju obavlja razvojni tim i održavanje radne liste proizvoda (engl. product backlog). Razvojni tim sastoji se od tri do devet osoba, od kojih članovi posjeduju sve vještine potrebne za razvoj sustava, a zaduženi su za izvršavanje zadataka potrebnih kako bi se prilikom svake iteracije mogla izdati nova verzija razvijanog sustava. Scrum majstor je osoba koja brine i koordinira proces, ali ne donosi poslovne ni tehničke odluke. Uloga Scrum majstora prema vlasniku proizvoda je osiguravanje da su cilj, doseg i domena poznati i jasni svim članovima Scrum tima, te održavanje događaja Scrum metodologije kada za to postoji potreba. Uloga Scrum majstora prema razvojnog timu je pomoć u organizaciji i dijeljenju znanja u timu, te uklanjanje prepreka tokom razvoja [1] [2].

### 2.1.2. Životni ciklus razvoja sustava Scrum metodologijom

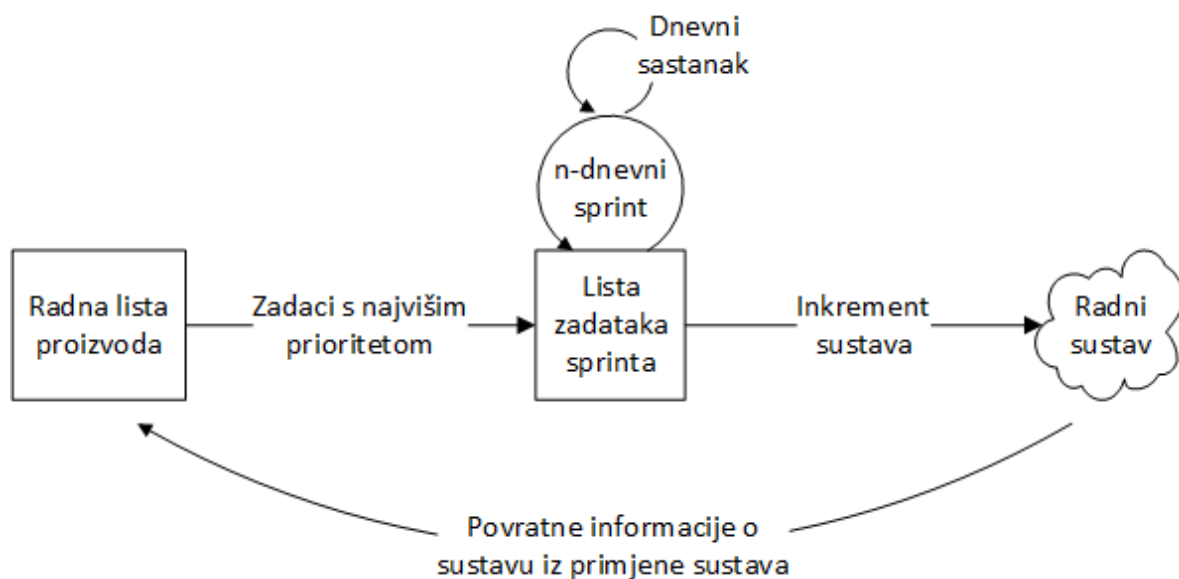
Razvoj sustava Scrum metodologijom zasniva se na iterativnim ciklusima, koji se nazivaju sprintevi, čije je trajanje fiksno te najčešće iznosi deset do petnaest radnih dana. Ovim iteracijama načelno su pokrivenne faze analize, oblikovanja, izrade i testiranja, predstavljene u poglavlju 2, dok inicijacija i završetak projekta nisu uključeni u ovu metodologiju. Stoga

se stvarni životni ciklus razvoja ovakvih sustava sastoji od inicijacije projekta, iterativnog razvoja sustava i završetka projekta. Tokom inicijacije projekta identificira se poslovni cilj i njegova izvršivost, stvaraju se inicijali zahtjevi i arhitektura sustava, te se stvara razvojni tim. Završetak projekta može uključivati migraciju korisnika na drugi sustav i isključivanje razvijenog sustava iz produkcije [2] [5]. Slika 2.2 prikazuje cijeli životni ciklus sustava razvijanog Scrum metodologijom.



Slika 2.2 Životni ciklus sustava razvijanog Scrum metodologijom

Iteracija u Scrum metodologiji, odnosno sprinteva, ima proizvoljno mnogo te se održavaju tako dugo dok razvoj nije završen. Svaki sprint započinje planiranjem kada cijeli Scrum tim sudjeluje u dogovoru što će biti napravljeno u narednom sprintu. Po završetku sprinta održava se pregled (engl. review) i retrospektiva sprinta. Tokom razvoja razvojni tim održava dnevne sastanke (engl. daily meeting) u trajanju od 15 minuta, kada tim međusobno komunicira što je od zadataka riješeno, a što će krenuti rješavati. Pri planiranju sprinta Scrum tim odgovara na dva pitanja pomoću kojih se određuje što će biti napravljeno u narednom sprintu, prvo pitanje je: „Što može biti napravljeno u ovome sprintu?“. Odgovorom na ovo pitanje, odabiru se zadaci iz radne liste proizvoda s najvišim prioritetom, te se stavljaju u listu zadataka sprinta (engl. sprint backlog). Drugo pitanje na koje se odgovara tokom planiranja sprinta je: „Kako će odabrani zadaci biti riješeni?“. U ovome dijelu sastanka, pojedini zadaci iz liste zadataka sprinta dijele se na manje zadatke koji su prikladne veličine da bi ih jedna osoba mogla obaviti. Pregled sprinta služi kako bi se demonstrirali rezultati sprinta vlasniku proizvoda i ostalim zainteresiranim osobama. Retrospektiva sprinta je događaj u kojem Scrum tim iskazuje pozitivna i negativna iskustva iz prethodnog sprinta te pokušava definirati akcije kako bi se povećao utjecaj pozitivnih iskustava, a uklonili utjecaj negativnih iskustava [1]. Slika 2.3 prikazuje radni tok tokom jednog sprinta.



Slika 2.3 Jedna iteracija u Scrum metodologiji [5]

### 2.1.3. Artefakti Scrum metodologije

U Scrum metodologiji razvoja sustava, postoje tri glavna artefakta koji se koriste tokom razvoja: radna lista proizvoda, lista zadataka sprinta i definicija završenog (engl. definition of “Done”). Radna lista proizvoda je popis svih funkcionalnosti, zahtjeva, poboljšanja i popravaka koje je potrebno napraviti na razvijanom sustavu tokom nekih od budućih verzija sustava. Ova lista je dinamička te ju najčešće održavaju vlasnik proizvoda i razvojni tim, koji konstantno rade na poboljšanju zadataka dodavanjem detaljnijih opisa, procjena ili mijenjanjem prioriteta zadataka. Konačna odluka o procjeni težine zadataka odgovornost je razvojnog tima. Lista zadataka sprinta je podskup zadataka iz radne liste proizvoda te dodatni zadaci koje razvojni tim smatra potrebnima kako bi se ostvario planirani cilj sprinta. Ovaj artefakt vidljiv je svima te se tokom iteracije redovito održava od strane razvojnog tima, kako bi cijeli Scrum tim imao prikaz realnog stanja i napretka sprinta. Listu zadataka sprinta razvojni tim najčešće održava u obliku ploče zadataka sprinta (engl. task board, sprint board). Definicija završenog koristi se kako bi cijeli Scrum tim imao zajedničku definiciju što znači kada je neki zadatak završen, koju svi razumiju i pridržavaju se te definicije. Definicija završenog najčešće je popis minimalnih zahtjeva koji moraju biti zadovoljeni da bi se zadatak smatrao završenim [1]. Tablica 2.1 prikazuje primjer definicije završenog za sustav razvijan Scrum metodologijom.

Tablica 2.1 Primjer definicije završenog za sustav razvijan Scrum metodologijom

<b>Kriterij</b>
Napravljen dizajn slučajeva testiranja za sve manualne testove
Postoje manualni ili automatski testovi koji pokrivaju sve stavke kriterija prihvaćanja (engl. acceptance criteria)
Svi automatski testovi prolaze
Razvijena funkcionalnost integrirana je u glavnu razvojnu granu

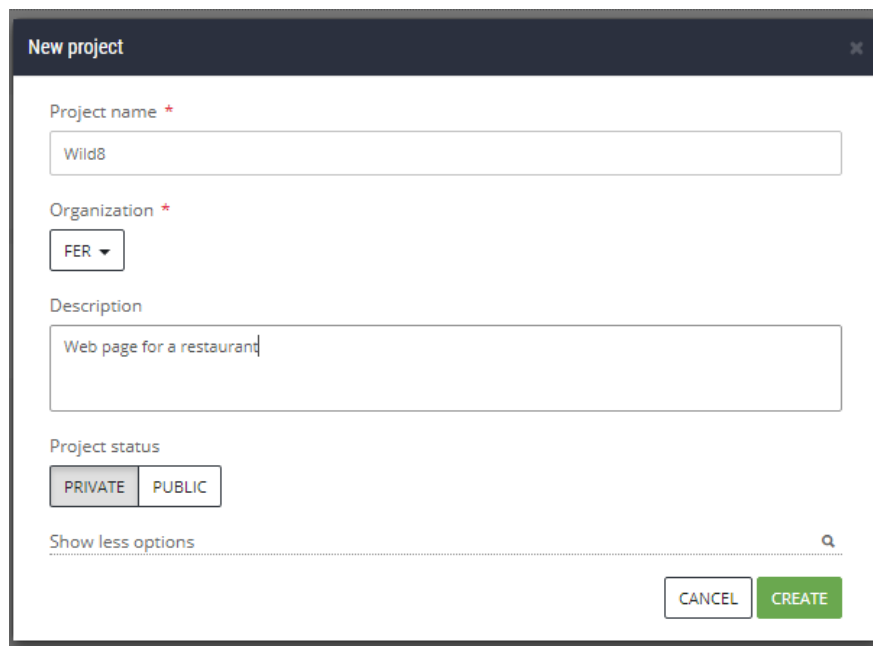
Osim navedenih artefakata, Scrum timovi često koriste i neka proširenja, poput grafa sagorijevanja (engl. burndown chart) koji grafički prikazuje nezavršeni posao unutar sprinta i predikciju, kad bi mogao biti završen i artefakt brzine tima (engl. velocity) koji se koristi kod planiranja novog sprinta kao granica posla koju tim procjenjuje da može obaviti u sprintu, a temelji se na povijesnim podacima iz prošlih sprinteva.

### 3. Programska podrška za upravljanje Scrum projektima

Analiza programske podrške radi se kako bi se dobilo bolje razumijevanje problemskog područja koje programska podrška treba riješiti. Programska podrška koja se analizira može biti sustav koji je potrebno unaprijediti ili neka druga programska podrška koja rješava sličan problem. Za područje upravljanja Scrum projektima razvoja informacijskih sustava postoji nekoliko već postojećih rješenja, među kojima su ScrumDesk [6] i Jira [7]. Mogućnosti ove dvije aplikacije testirane su na primjeru projekta razvoja web stranice za restoran.

#### 3.1. ScrumDesk

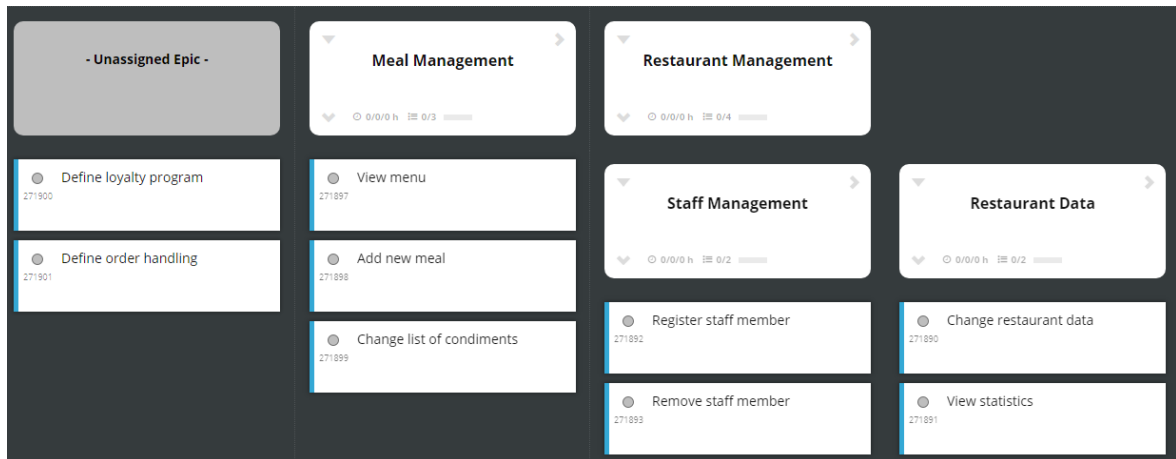
ScrumDesk je online alat koji je besplatan za timove do 4 osobe, realiziran u obliku web aplikacije, a omogućuje praćenje više projekata unutar organizacije. Ovaj alat podržava privatne i javne projekte. O pojedinom projektu bilježi se ime i opis projekata, a uz projekt veže se nekoliko postavka, između kojih su trajanje sprinta i podrazumijevani radni dani. Postavke vezane uz projekt podešavaju se naknadno nakon kreiranja projekta. Slika 3.1 prikazuje ekran za kreiranje novog projekta.



The image shows a 'New project' form in the ScrumDesk application. The form has a dark header bar with the title 'New project' and a close button. The main content area is white and contains several input fields and buttons. The 'Project name' field is required (marked with a red asterisk) and contains the text 'Wild8'. The 'Organization' field is also required and is a dropdown menu currently showing 'FER'. The 'Description' field is a larger text area containing 'Web page for a restaurant'. Below these fields are two buttons for 'Project status': 'PRIVATE' (which is selected and highlighted) and 'PUBLIC'. At the bottom of the form, there is a link 'Show less options' and two buttons: 'CANCEL' and 'CREATE' (which is green).

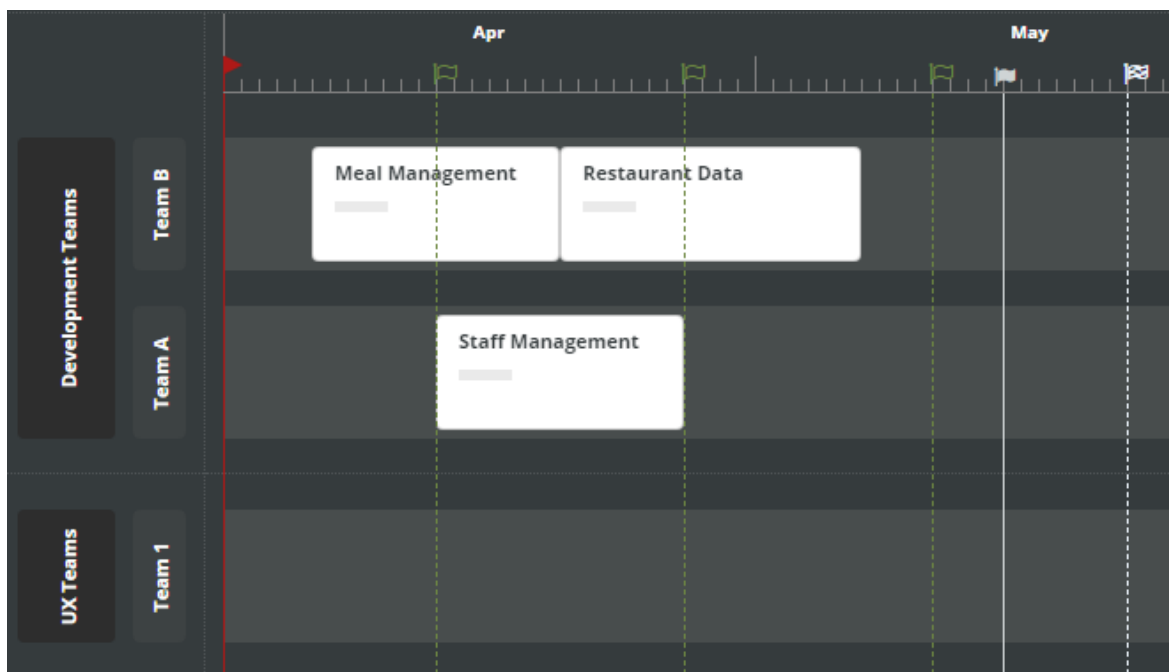
Slika 3.1 Prozor za stvaranje novog projekta u alatu ScrumDesk

U ovome alatu postoji nekoliko kartica koje pokrivaju aktivnosti planiranja i razvoja: *Story Map*, *Backlog*, *Roadmap*, *Plan*, *Work*, *Retro* i *Reports*. Kartica *Story Map* namijenjena je primarno za vlasnika proizvoda i razvojni tim kao podrška za planiranje sustava na način da se korisničke priče grupiraju u epove (engl. epic) koji predstavljaju dio funkcionalnosti sustava. Slika 3.2 prikazuje karticu *Story Map* u kojem je su funkcionalnosti sustava podijeljene na upravljanje jelima i upravljanje restoranom. Na kartici *Story Map* uvijek se nalazi i stupac za zadatke koji trenutno nisu pridruženi ni jednom epu.



Slika 3.2 Kartica *Story Map* u alatu ScrumDesk

Uz karticu *Story Map*, kao podrška za planiranje razvoja sustava, važna je i kartica *Roadmap*, koja omogućuje dugoročno planiranje razvoja sustava odnosno rasporeda kada će se raditi na nekom epu. Ova kartica omogućava vremensko planiranje rada po timovima ili grupama timova, uz označavanje nekih važnih događaja tokom razvoja, poput izdavanja nove verzije razvijenog sustava, kraj pojedinog sprinta ili kontrolnih točaka (engl. milestone). Slika 3.3 prikazuje karticu *Roadmap* alata ScrumDesk, za tri tima i njima pridijeljeni epovi i funkcionalnosti (engl. feature). Crvena zastavica označava trenutni datum, zelene zastavice označavaju krajeve sprinteva, sivom zastavicom označena je kontrolna točka, a bijelo-crna zastavica označava datum izdavanja nove verzije razvijanog sustava.



Slika 3.3 Kartica *Roadmap* u alatu ScrumDesk

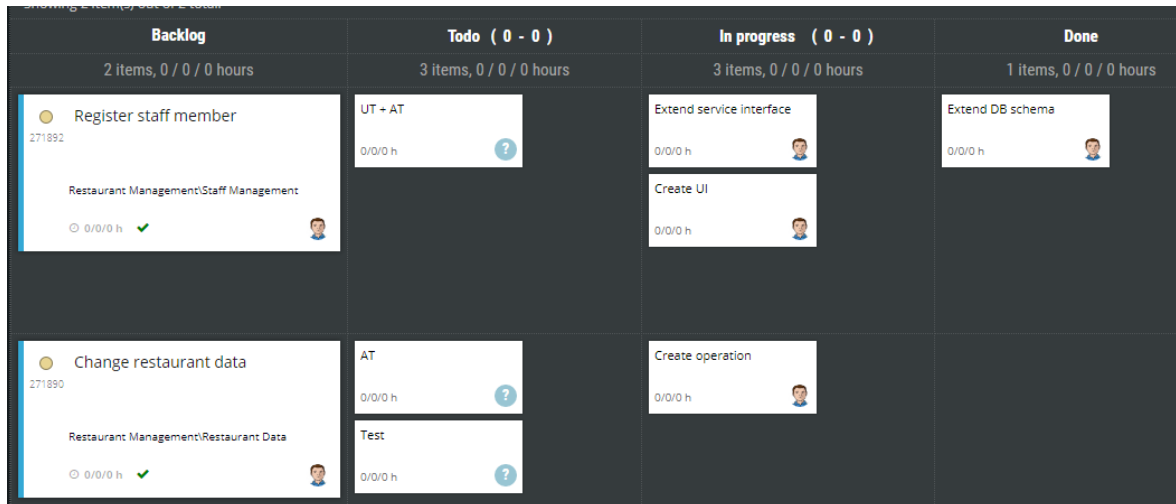
Kartica *Backlog* služi za pregled radne liste proizvoda, na njemu su vidljivi svi nezavršeni zadaci na projektu, moguće ih je grupirati i sortirati po različitim kriterijima, primjerice kojem su sprintu pridruženi ili prema njihovom prioritetu. Postoji nekoliko unaprijed određenih tipova zadataka. U radnoj listi proizvoda nalaze se svi tipovi zadataka svi tipovi osim epova i funkcionalnosti. Na ovoj kartici vidljiv je i status pojedinog zadatka, koji može označavati da je zadatak spreman i čeka implementaciju, čeka doradu ili ga nije trenutno nije moguće izvršiti zbog nekog drugog razloga. Slika 3.4 prikazuje karticu *Backlog* na kojoj su vidljive tri korisničke priče, jedna greška (engl. bug) i jedno poboljšanje. Tip pojedinog zadatka može se prepoznati po boji kojom je označen. Svijetloplavom bojom označene su korisničke priče, crvenom bojom označen je defekt, a poboljšanje je označeno zelenom bojom.

#271892	Register staff member	Staff Mana...	V10	Sprint 01		●	0/0/0 h	✓	🗨	ⓘ
#271890	Change restaurant data	Restauran ..	V10	Sprint 01		●	0/0/0 h	✓	🗨	ⓘ
#271913	User can't login		V10	Sprint 02		●	0/0/0 h	✓	🚩	ⓘ
#271901	Define order handling		V10	Sprint 03		●	0/0/0 h	✓	🚩	ⓘ
#271914	Enable multiple condime...		V10	Sprint 03		●	0/0/0 h	✓	🗨	ⓘ

Slika 3.4 Kartica *Backlog* u alatu ScrumDesk

Za planiranje pojedinih verzija projekata koristi se kartica *Plan*, u njoj je moguće stvoriti nove verzije sustava, pridruživati nove sprinteve kreiranim verzijama sustava, te pridijeliti

sprintevima zadatke iz radne liste proizvoda koji pripadaju toj verziji. Kada je sprint aktivan, osobama koje sudjeluju u razvoju najvažnija je kartica *Work*, na njoj se nalazi ploča zadataka sprinta, na kojoj se prikazuju korisničke priče. Korisničkoj priči pridruženo je nekoliko zadataka, za pojedini zadatak prati se trenutno stanje koje može biti: čeka na izvršavanje, izvršava se ili je završen. Slika 3.5 prikazuje ploču zadataka sprinta kojem su pridružene dvije korisničke priče s nekoliko zadataka. Jedan zadatak je završen, tri zadatka se izvršavaju, a ostali čekaju na izvršavanje.



Slika 3.5 Prikaz ploče zadataka u alatu ScrumDesk

Kartica *Retro* zamišljena je kao podrška za retrospektivu na kojoj članovi tima mogu staviti pozitivne ili negativne dojmove iz prošlog sprinta te ih ocijeniti kako bi se odlučile stavke o kojima će se detaljnije razgovarati tokom retrospektive. U kartici *Reports* nalaze se izvještaji i grafovi sa statistikama o razvoju sustava. Dostupni grafovi su graf sagorijevanja, kumulativni graf statistika radne liste proizvoda, histogram brzine, graf ostvarene vrijednosti i graf kompleksnosti. Osim navedenih grafova, u kartici *Reports* može se kreirati izvještaj za pojedini sprint. Na kojem su vidljive osobe koje su sudjelovale u sprintu i vrijeme koje su utrošile, usporedba koliko je priča bilo planirano, a koliko završeno, graf sagorijevanja sprinta, te lista zadataka sprinta s ostvarenim rezultatom, odnosno stanjem zadatka po završetku sprinta. Slika 3.6 prikazuje izvještaj jednog sprinta kreiranog alatom ScrumDesk.



## Sprint: S#10 ▾

What we did. What we can. Get context. A law.

From 06/07/2015 11:19 to 11/09/2015 17:53

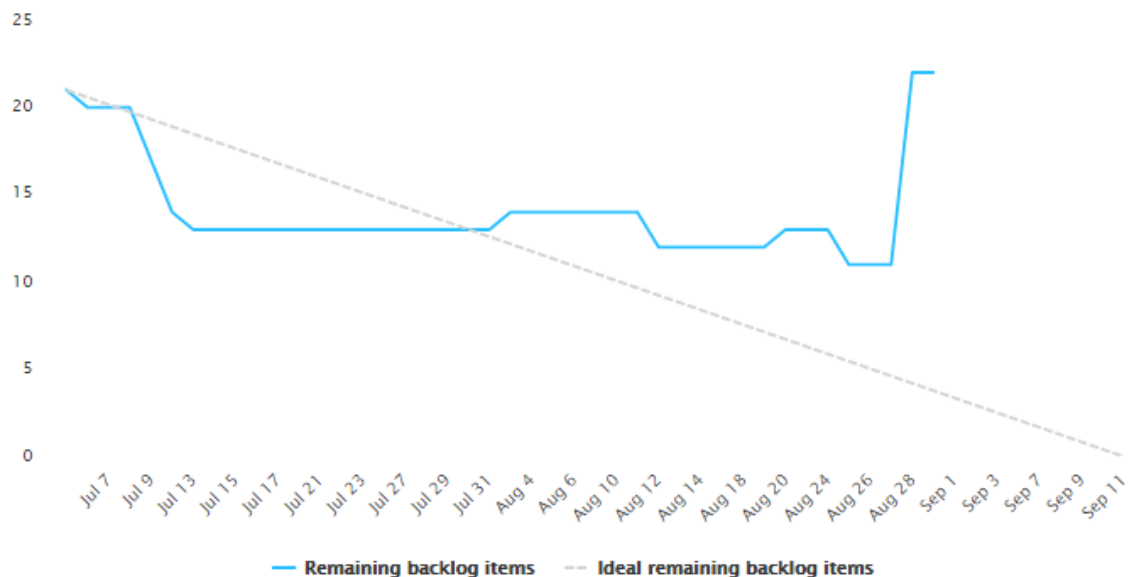
### Team

Role	Name	Hours (Spent/Est)
PRODUCT OWNER	Dusan Kocurek	0 / 0
TEAM MEMBER	Juraj Pikora	61.5 / 138
SCRUM MASTER	Ondrej Proksa	11 / 49
SCRUM MASTER	Vladimir Oleksak	0 / 0

### Sprint Overview

Planned			Completed		
Items	Effort	Time	Items	Effort	Time
31	12.5	187h	9	4	22h

### Bumdown Chart



### BACKLOG ITEMS

### EFFORT

### TIME

### Sprint Backlog

*	ID	Type	Title	Status	Estimate	Spent
	29967	Bug	Defects S#10	IN PROGRESS	5h	1.5h
	29969	User story	Unplaced backlog items II	DONE	2h	2h
	24862	User story	Terms of policy and privacy agreement	DONE	2h	2h
	6937	User story	Cookies warning	DONE	3h	2h
	29272	User story	Location of new card	DONE	1h	1h

Slika 3.6 Izvještaj jednog sprinta kreiranog alatom ScrumDesk [6]

## 3.2. Jira

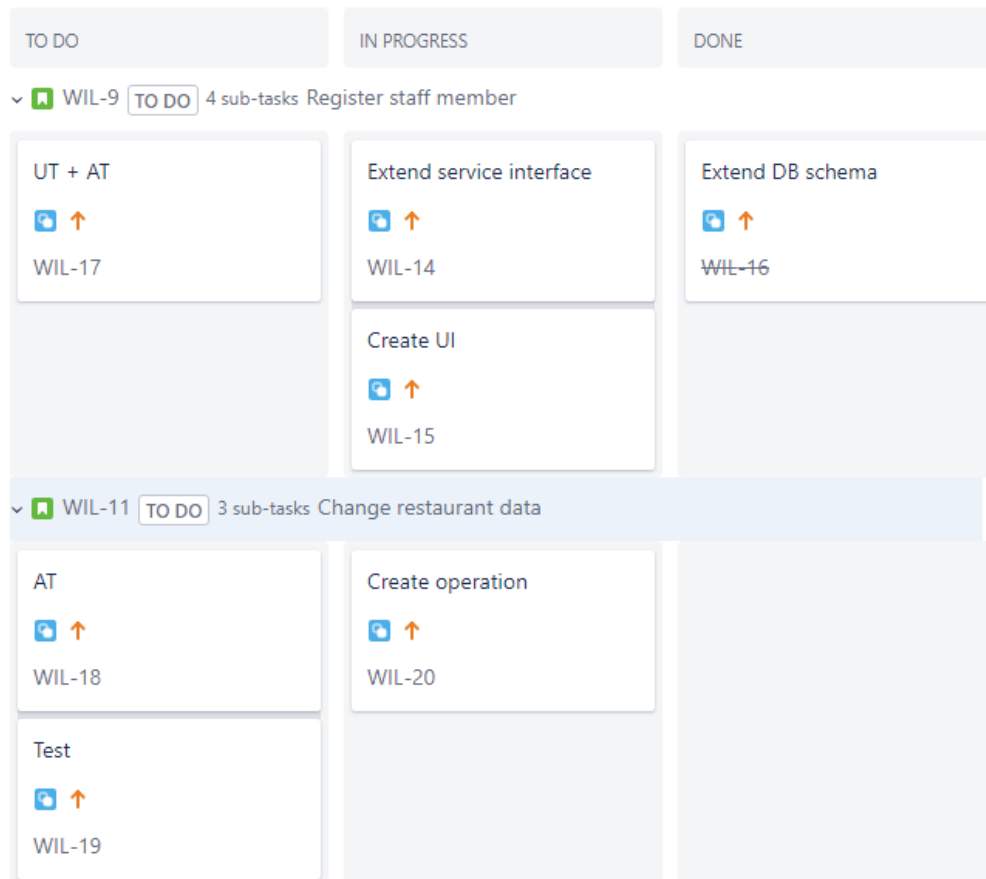
Jira je komercijalan online alat, kao i ScrumDesk realiziran je u obliku web stranice. Za razliku od ScrumDesk alata koji ima ugrađenu podršku isključivo za Scrum projekte, Jira podržava različite tipove projekata među kojima su Scrum i Kanban projekti. Jira također omogućava praćenje više projekata unutar iste organizacije. Kao i kod prethodnog alata, različite funkcionalnosti za praćenje projekata podijeljene su u nekoliko kartica: *Backlog*, *Active sprints*, *Reports*, *Releases*, *Issues*, *Pages* i *Components*. Kartica *Backlog* služi za pregled radne liste proizvoda, planiranje sprinteva, a moguće je i kreirati nove zadatke u radnoj listi proizvoda ili u sprintu. Slika 3.7 prikazuje izgled kartice *Backlog* gdje su planirana dva sprinta s pridruženim zadacima i radna lista proizvoda.

The screenshot displays the Jira Backlog interface. At the top, there are two sprint cards: 'WIL Sprint 1' with 2 issues and 'WIL Sprint 2' with 2 issues. Each sprint card has a 'Start sprint' button and a 'Plan sprint' dropdown. Below the sprints is a 'Backlog' card with 6 issues. Each issue card includes a status icon (green square with a white 'A' or red square with a white 'X'), the issue title, the component name in a colored box, the issue key, and a priority indicator (orange arrow up or down, and a grey circle with a minus sign). A '+ Create issue' link is present below each sprint and the backlog. A 'Create sprint' button is located at the bottom right of the backlog card.

Sprint	Issue Title	Component	Key	Priority
WIL Sprint 1	Register staff member	Restaurant Management	WIL-9	↑
	Change restaurant data	Restaurant Management	WIL-11	↑
WIL Sprint 2	User can't login		WIL-13	↑
	Define order handling		WIL-8	↑
Backlog	View menu	Meal Management	WIL-3	↑
	Add new meal	Meal Management	WIL-4	↑
	Change list of condiments	Meal Management	WIL-5	↑
	Define loyalty program		WIL-7	↑
	Remove staff member	Restaurant Management	WIL-10	↑

Slika 3.7 Kartica *Backlog* u alatu Jira

Kartica *Active sprints* ima istu funkcionalnost kao i kartica *Work* u alatu ScrumDesk, na ovoj kartici nalazi se ploča zadataka sprinta, a zadaci su grupirani prema korisničkim pričama kojima pripadaju. Ploča je podijeljena na tri stupca koji prikazuju u kojem se stanju nalazi pojedini zadatak, stanja pojedinih zadataka inicijalno su ista kao i u alatu ScrumDesk, međutim u ovome alatu imena i broj ovih stanja može se konfigurirati po želji. Slika 3.8 prikazuje ploču zadataka sprinta u alatu Jira, gdje su stanja zadataka definirana prema inicijalnim postavkama.



Slika 3.8 Prikaz ploče zadataka sprinta u alatu Jira

Kartica *Releases* omogućava kreiranje novih verzija razvijanog sustava za koje se bilježe dan početka razvoja i dan izdavanja, a na kartici *Issues* nalazi se popis svih zadataka gdje ih je moguće pretraživati prema različitim kriterijima, poput datuma kreiranja, osobe koja je kreirala zadatak, statusu ili tipu zadatka. Kartica *Pages* služi kao kolekcija dokumenata vezanih uz projekt, poput dokumenata sa specifikacijom zahtjeva ili objašnjenjem nekih arhitekturnih odluka koje su donesene tokom razvoja sustava. Kartica *Components* namijenjena je korištenju u velikim projektima kada se projekt sastoji od više komponenata

koje je moguće razvijati nezavisno. Na kartici *Reports* moguće je stvoriti različite vrste izvještaja, podržane su sve vrste izvještaja kao i u alatu ScrumDesk. Također postoje i izvještaji za analizu zadataka gdje je moguće analizirati statistike zadataka poput prosječne starosti zadataka. Izvještaj o sprintu u ovome alatu sadrži graf sagorijevanja i izvještaj o statusu zadataka iz liste zadataka sprinta.

### **3.3. Usporedba postojećih rješenja**

Oba rješenja prikladna su za upravljanje Scrum projektima i sadrže sve potrebne funkcionalnosti potrebne za upravljanje takvim projektima. Alat ScrumDesk primarno je namijenjen za upravljanje Scrum projektima, dok Jira ima podršku za više metodologija razvoja, više mogućnosti za konfiguraciju i prilagodbu vlastitom radnom toku.

## 4. Zahtjevi na vlastito rješenje

Zahtjevi na programsku podršku u detalje upisuju što razvijani sustav treba raditi. Eksplicitni set zahtjeva pomaže u osiguravanju da korisnik, a ne programer, odlučuje o funkcionalnosti sustava. Najčešće se zahtjevi dijele na zahtjeve korisnika i na zahtjeve sustava. Zahtjevi korisnika specificirani su u prirodnom jeziku te opisuju što korisnik očekuje od sustava [8] [9]. Ova vrsta zahtjeva prikuplja se intervjuiranjem stvarnih korisnika programske podrške, analizom dokumentacije ili promatranjem poslovnog procesa. U Scrum projektima korisničke zahtjeve definira vlasnik proizvoda koji prema ostatku Scrum tima komunicira želje korisnika u obliku zadataka i korisničkih priča u radnoj listi proizvoda [1] [2]. Korisnički zahtjevi za razvijani sustav su:

- korisnik mora moći upravljati s više projekata,
- korisnik mora moći definirati uloge osoba na projektu,
- korisnik mora moći evidentirati rezultate sastanaka,
- korisnik mora moći planirati izvođenje sprinteva,
- korisnik mora moći kreirati izvještaje.

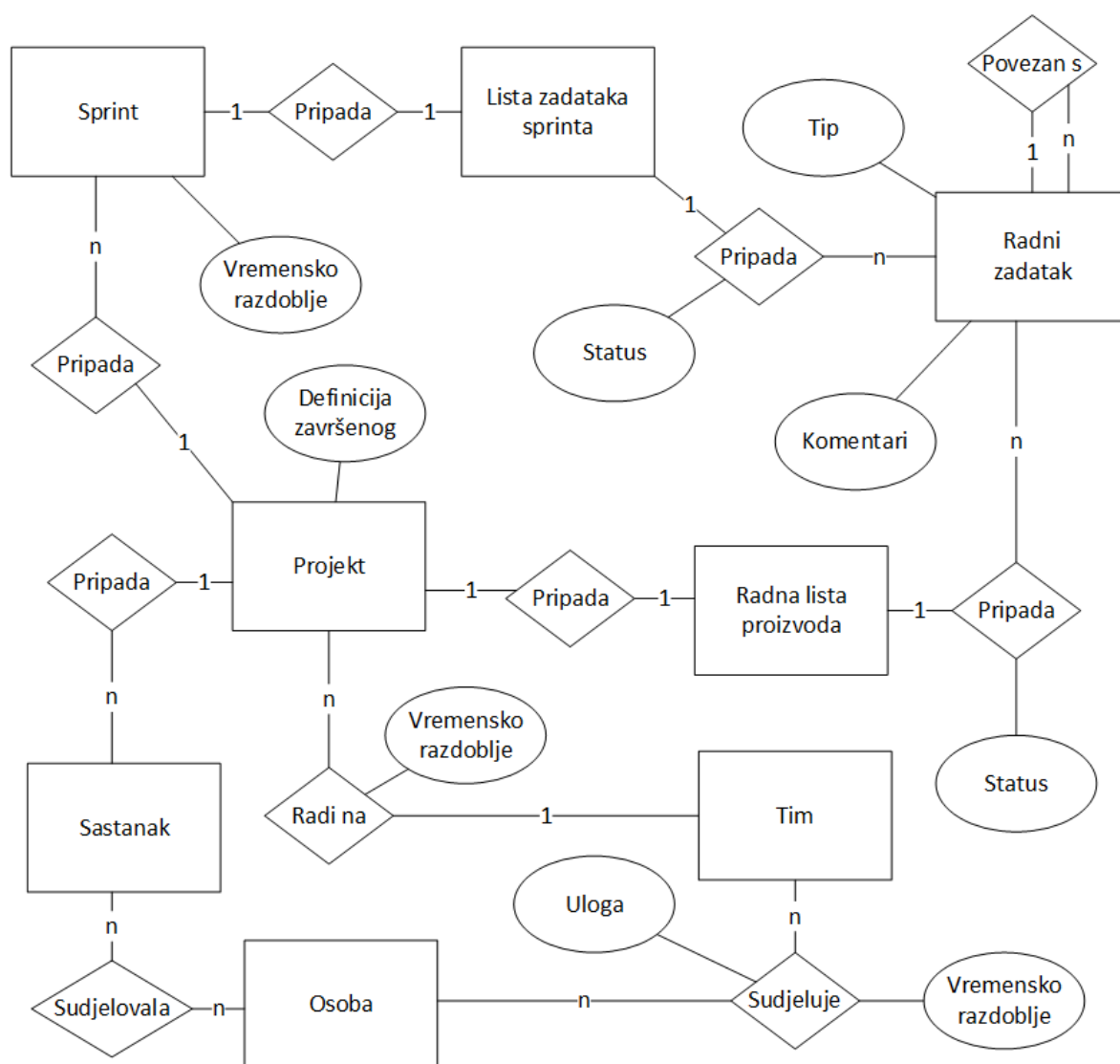
Zahtjevi sustava su detaljni opis funkcionalnosti i ograničenja sustava, ovi zahtjevi trebali bi detaljno definirati što će biti napravljeno. Govore o tome kako sustav reagira na određene ulaze ili kako se ponaša u nekim situacijama, ponekad mogu specificirati i ono što sustav ne bi trebao učiniti [9]. Ova vrsta zahtjeva prikuplja se detaljnom analizom korisničkih zahtjeva i analizom postojećih rješenja. Zahtjevi sustava na razvijen sustav su:

- sustav mora omogućiti upravljanje s barem tri projekta unutar iste organizacije,
- sustav mora omogućiti evidenciju najmanje tri osobe na projektu s ulogama korištenim u Scrum metodologiji,
- sustav mora omogućiti definiranje zadataka različitih tipova, među kojima su: ep, korisnička priča, zadatak i defekt,
- sustav mora omogućiti evidenciju artefakata Scrum metodologije, među kojima su radna lista proizvoda, lista zadataka sprinta i definiciju završenog,
- sustav mora omogućiti evidenciju retrospektiva sprinta i predloženih odlučениh akcija

- sustav mora omogućiti definiranje pojedinog sprinta, te evidenciju statusa pojedinog zadatka tokom izvršavanja sprinta,
- sustav mora omogućiti pregled izvještaje za pojedini sprint.

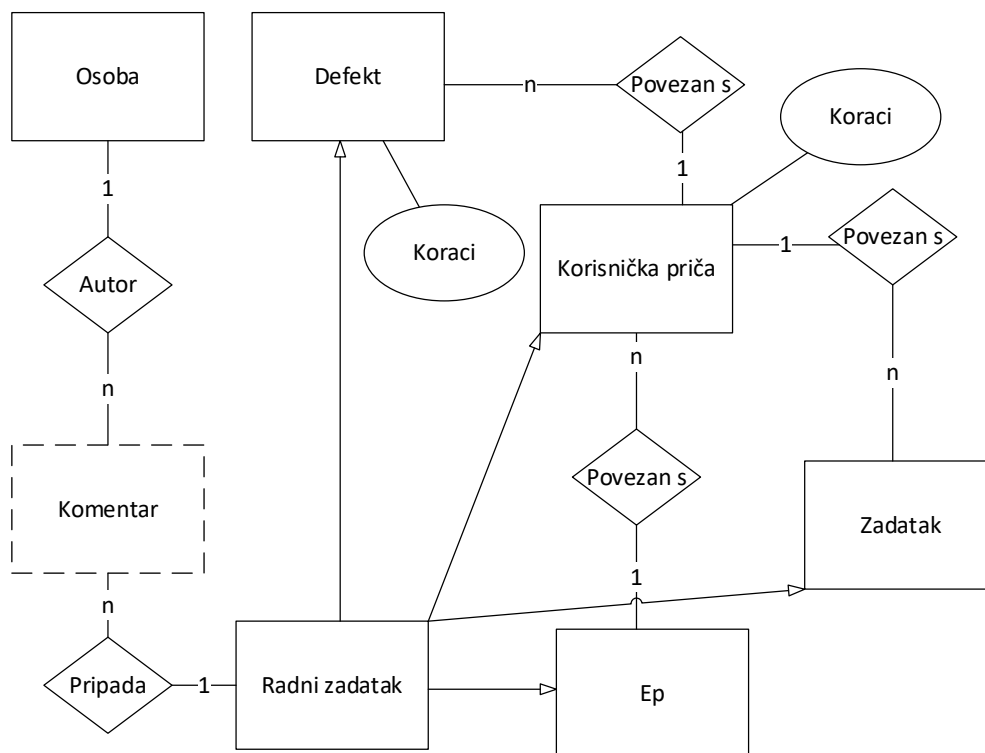
## 4.1. Konceptualni model podataka

Temeljem analize korisničkih zahtjeva, zahtjeva sustava i analize postojećih rješenja, identificirani su entiteti koji se nalaze u domeni sustava: osoba, tim, projekt, zadaci različitih tipova, sprint i sastanak. Slika 4.1 prikazuje konceptualni model podataka s entitetima i osnovnim vezama između njih.



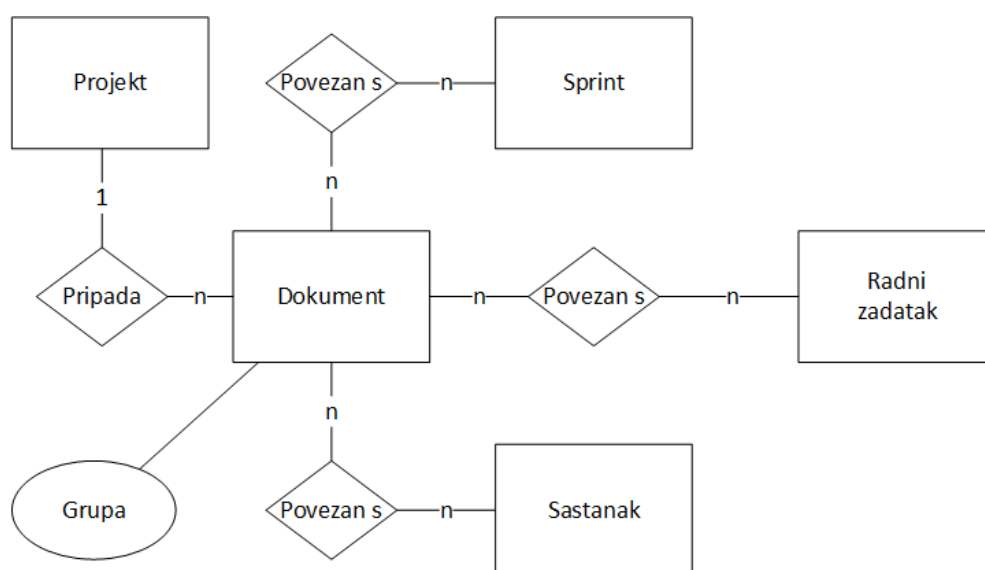
Slika 4.1 Konceptualni model podataka

Entitet osobe predstavlja jednu osobu unutar organizacije, za koju se bilježi ime, prezime i kontakt podaci. Osoba može sudjelovati u više timova u isto vrijeme i s različitim uloga, ali ukoliko osoba ima ulogu vlasnika proizvoda u timu, ona ne može biti u nijednoj drugoj ulozi u tom timu u tom vremenskom razdoblju. Za tim se bilježi naziv tima i vrijeme dnevnog sastanka i trenutni članovi tima. Moguće uloge osoba u timu su: vlasnik proizvoda, Scrum majstor i član razvojnog tima. U timu uvijek mora biti jedna osoba koja je vlasnik proizvoda, jedna osoba koja je Scrum majstor i jedna osoba koja je član razvojnog tima. Entitet projekta predstavlja proizvod koji se razvija, za kojeg se bilježi naziv, opis i razvojni tim. Naziv projekta mora biti jedinstveni unutar organizacije. Za projekt postoje artefakti Scrum metodologije: radna lista proizvoda i definicija završenog, te mu pripadaju sastanci, sprintevi i razvojni zadaci. Na projektu unutar određenog vremenskog razdoblja radi isključivo jedan tim. Radna lista proizvoda sadrži sve do sada neriješene epove, korisničke priče, zadatke i defekte za taj projekt. Za sastanke se bilježi naziv sastanka, datum održavanja i trajanje, opis, osobe koje bi trebale sudjelovati na tom sastanku i zapisnik sastanka. Za sprinteve se bilježi vremensko razdoblje u kojem se održavaju, naziv i cilj sprinta, te lista zadataka sprinta. Vremensko razdoblje održavanja sprinta je jedinstveno na projektu, odnosno ne može postojati preklapanje među sprintevima. Lista zadataka sprinta sadrži sve korisničke priče na kojima se radi u tom sprintu zajedno s njihovim trenutnim statusom. Entitet razvojnog zadatka predstavlja jedan zadatak na projektu, koji imaju različite tipove te se međusobno mogu povezivati. Podržani tipovi radnih zadataka su ep, korisnička priča, zadatak i defekt. Za svaki tip zadatka bilježi se naziv, procijenjena težina i opis. Kao podrška kolaboraciji sve tipove razvojnih zadataka moguće je komentirati. Ep može imati povezane korisničke priče, dok one mogu biti povezane sa zadacima i defektima. Korisničke priče i defekti uz ostale attribute imaju i popis koraka, odnosno scenarij izvršavanja korisničke priče. Slika 4.2 prikazuje detaljan model radnih zadataka.



Slika 4.2 Konceptualni model radnih zadataka

Uz prethodno navedene entitete postoji i pomoćni entitet koji predstavlja dokumente, za dokumente se bilježi naslov, sadržaj i lista vanjskih poveznica. Dokumenti se mogu vezati uz sam projekt, sastanak, sprint ili radni zadatak. Dokumenti imaju svoju grupu koja označava uz koje se entitete pojedini dokument može vezati, pa tako postoje dokumenti koji se mogu povezati samo s projektom, sprintevima, sastancima, radnim zadacima ili sa svim navedenim entitetima. Slika 4.3 prikazuje konceptualni model dokumenata.



Slika 4.3 Konceptualni model dokumenata



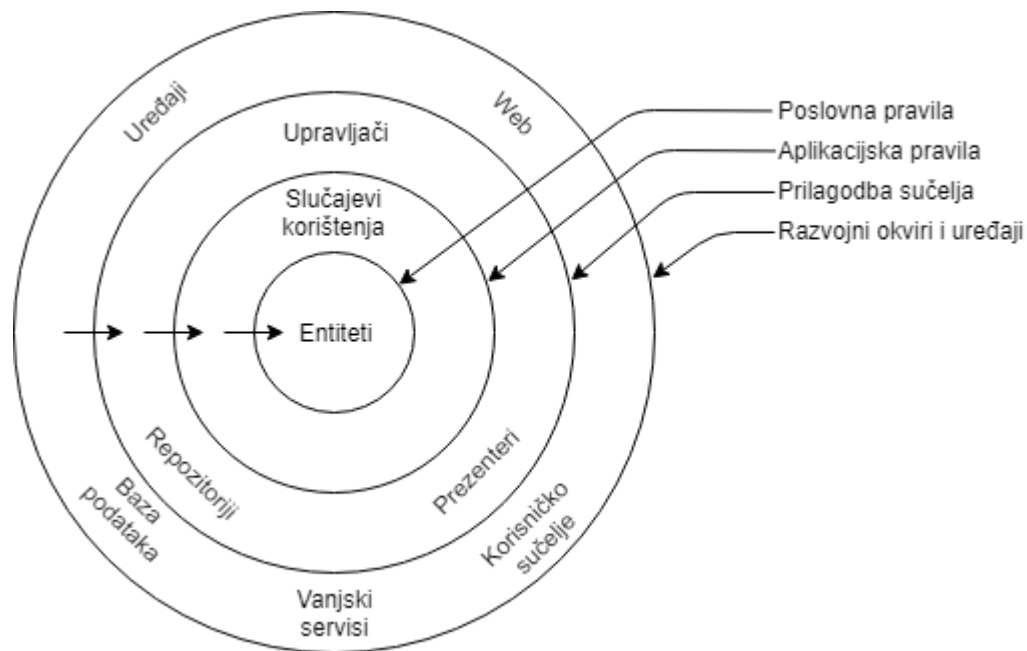
## 5. Arhitektura razvijenog sustava

Arhitektura programske podrške opisuje sustav s visoke razine u kojoj se definiraju komponente sustava i interakcija među njima. Odluke u arhitekturi mogu utjecati na svojstva sustava poput performansi, sigurnosti i dostupnosti sustava [9]. Arhitektura razvijenog sustava rađena je po predlošku „čiste“ arhitekture (engl. clean architecture), čija je osnovna motivacija izolacija domene sustava od vanjskih tehnologija i biblioteka. Izolacija se postiže razdvajanjem komponenata sustava u slojeve od kojih svaki ima određenu ulogu, te strogim pridržavanjem pravila o smjeru ovisnosti unutar sustava. Ovakva arhitektura omogućava izradu sustava koji ima sljedeće karakteristike:

- nezavisnost o razvojnim okvirima – arhitektura ne ovisi o postojanju ili ograničenjima neke biblioteke,
- ispitljivost (engl. testability) – poslovna pravila moguće je provjeriti automatskim testovima, bez korištenja korisničkog sučelja, baze podataka i vanjskih servisa,
- nezavisnost poslovnih pravila – poslovna pravila nezavisna su o korisničkom sučelju, bazi podataka i vanjskim servisima, te ih je moguće zamijeniti drugima bez utjecaja na poslovna pravila [10].

Prema razini apstrakcije komponente sustava mogu se grupirati u komponente koje realiziraju poslovna pravila, komponente koje realiziraju aplikacijska pravila, komponente za prilagodbu sučelja, te razvojne okvire i uređaje. Najapstraktnije komponente su entiteti koji enkapsuliraju poslovna pravila, te također predstavljaju i poslovne objekte aplikacije ukoliko se razvija samo jedan informacijski sustav koji ih koristi. Zajedno s entitetima, slučajevi korištenja (engl. use case) predstavljaju domenu aplikacije. Slučajevi korištenja sadrže aplikacijska pravila te upravljaju entitetima kako bi se ostvario cilj slučaja korištenja. Sloj za prilagodbu sučelja je skup adaptera koji ostvaruje komunikaciju između vanjskih servisa ili uređaja prema slučajevima korištenja i entitetima. Smjer ovisnosti unutar sustava uvijek je u smjeru apstrakcije. Najviše apstraktne komponente ne smiju ovisiti o manje apstraktnim komponentama [10]. Na primjer, entiteti kao najapstraktnija komponenta ne smiju ovisiti o slučajevima korištenja. Ukoliko je potrebna komunikacija iz unutarnjeg sloja prema vanjskom, ona mora biti ostvarena preko sučelja definiranog u unutarnjem sloju, a implementiranog u vanjskom sloju. Slika 5.1 prikazuje komponente sustava u čistoj

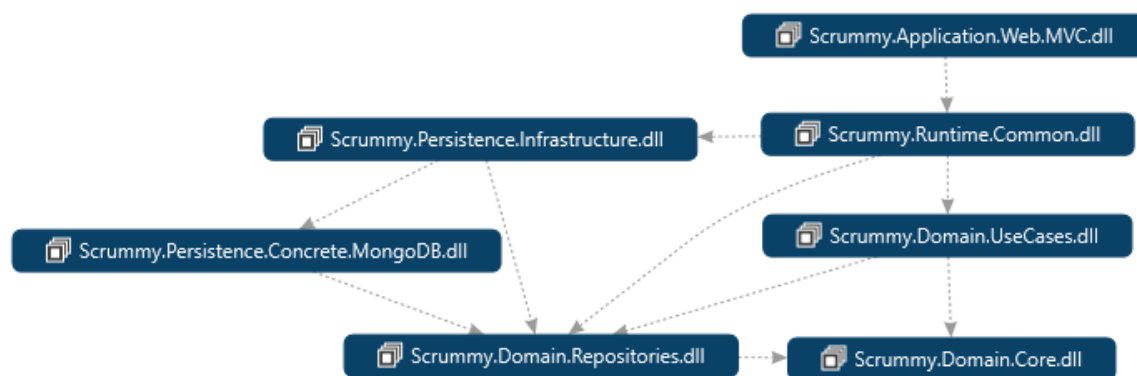
arhitekturi te kojem sloju pripadaju prema razini apstrakcije. Strelicama s lijeve strane slike naznačen je smjer u kojem je dopuštena ovisnost između komponenata.



Slika 5.1 Komponente sustava u čistoj arhitekturi [10]

## 5.1. Struktura razvijenog rješenja

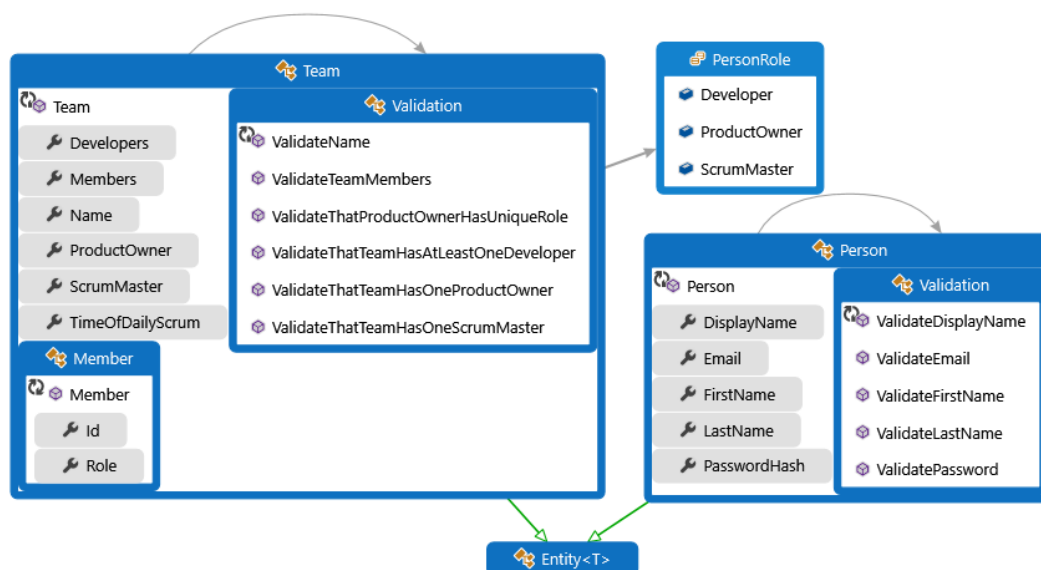
Razvijeni sustav sastoji se od sedam komponenti i baze podataka. Komponenta „Domain.Core“ sadrži entitete i iznimke (engl. exceptions) koje entiteti koriste za javljanje pogrešaka, komponenta „Domain.UseCases“ sadrži slučajeve korištenja i iznimke koje oni koriste, komponenta „Domain.Repositories“ sadrži definiciju sučelja prema perzistenciji podataka, komponente „Persistence.Infrastructure“ i „Persistence.Concrete.MongoDB“ realiziraju perzistenciju podataka, komponenta „Application.Web.MVC“ u kojem se nalazi implementacija web aplikacije, te pomoćna komponenta „Runtime.Common“ koji se koristi za inicijalizaciju komponenata prilikom pokretanja sustava. O bazi podataka ovisi jedino komponenta „Persistence.Concrete.MongoDB“. Slika 5.2 prikazuje komponente razvijenog rješenja i ovisnosti između njih. Ovisnost između komponenata prikazana je strelicom, a ukoliko strelica pokazuje prema nekoj komponenti, to je oznaka da ta komponenta ovisi o komponenti na kojeg pokazuje.



Slika 5.2 Komponente razvijenog rješenja

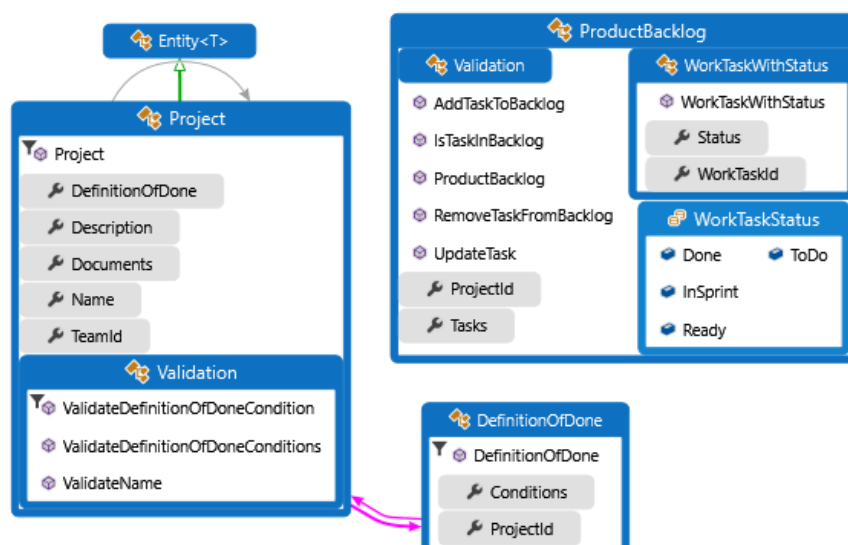
## 5.2. Entiteti

Entiteti su poslovni objekti razvijenog sustava, kao što je navedeno u poglavlju 4.1 postoje entiteti koji predstavljaju osobe, timove, projekte, radne zadatke, sprinteve i sastanke. Sve klase koje realiziraju entitete naslijeđene su iz klase „Entity“ koja je bazna klasa modela domene, a sadrži svojstvo (engl. property) „Id“ koje predstavlja jedinstveni identifikator entiteta, te par pomoćnih metoda za usporedbu entiteta i validaciju. Entitet osobe realiziran je klasom „Person“ koja preko javnog sučelja otkriva podatkovne članove koji se bilježe za osobe i pomoćnu klasu za validaciju poslovnih pravila tih podatkovnih članova. Entitet tima realiziran je klasom „Team“ i klasom „Member“ koja predstavlja zapis o jednom članu tima i enumeracijom „PersonRole“ gdje su definirane moguće uloge osoba unutar tima, a kao i klasa „Person“, klasa „Team“ sadrži pomoćnu klasu za validaciju. Trenutni članovi tima pamte se unutar kolekcije „Members“ koja sadrži parove entiteta i uloge osobe u timu. Slika 5.3 prikazuje javno sučelje klase koje implementiraju entitete osoba i timova.



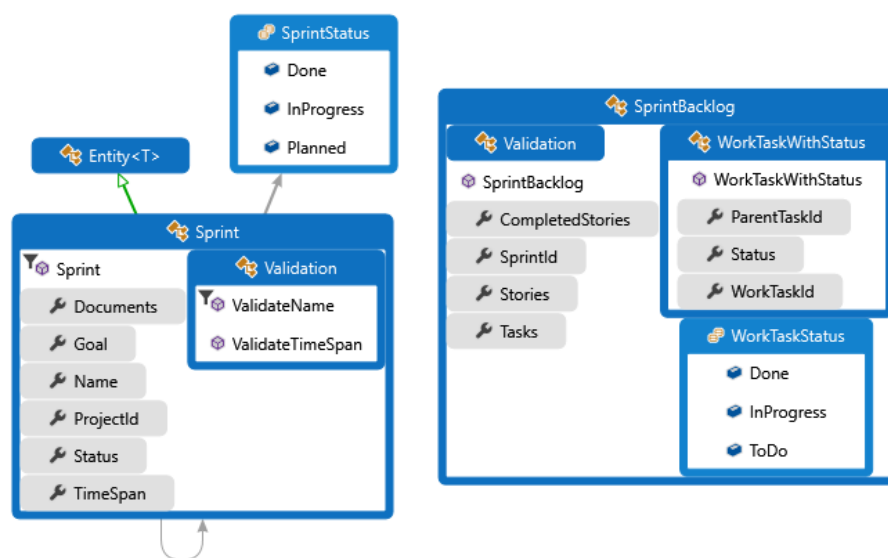
Slika 5.3 Model entiteta timova i osoba

Entitet projekta realiziran je klasom „Project“ i klasom „DefinitionOfDone“ koja predstavlja artefakt Scrum metodologije definicije završenog. Drugi artefakt Scrum metodologije, radna lista proizvoda, realiziran je klasom „ProductBacklog“ koja prikazuje trenutno stanje radne liste proizvoda, dok su moguća stanja zadatka definirana enumeracijom „WorkTaskStatus“. U radnoj listi proizvoda zadaci pamte se unutar kolekcije „Tasks“ koja sadrži parove identiteta i stanja zadatka. Radna lista proizvoda povezana je s projektom pomoću jedinstvenog identifikatora koji se bilježi u svojstvu „ProjectId“. Slika 5.4 prikazuje javno sučelje entiteta projekta i radne liste proizvoda.



Slika 5.4 Model entiteta projekta i radne liste proizvoda

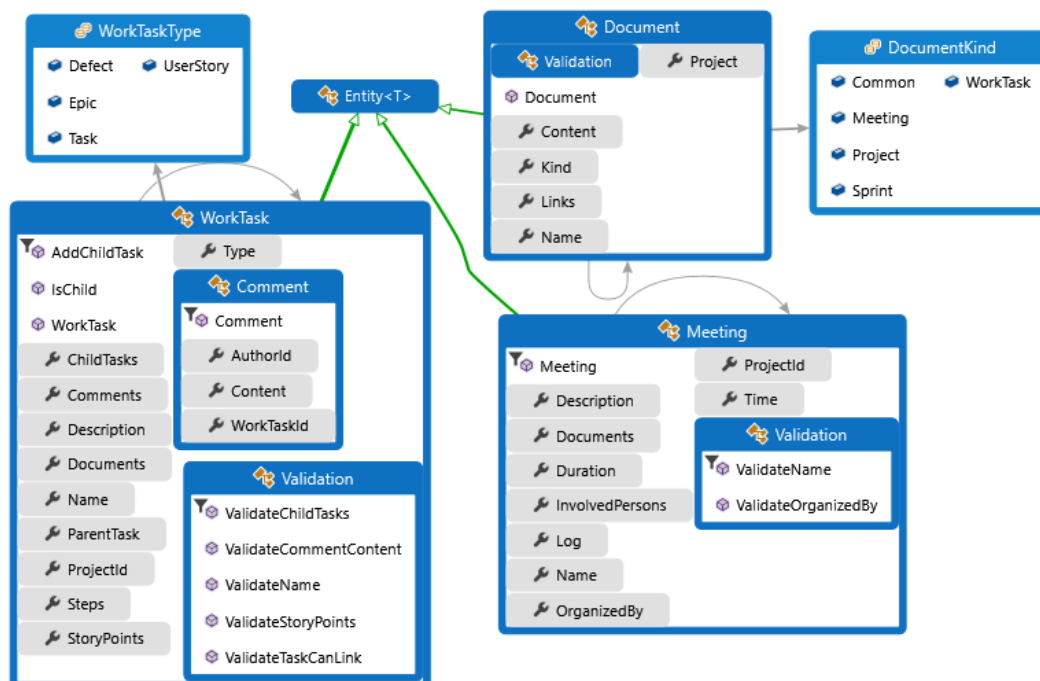
Entitet sprinta realiziran je klasom „Sprint“, a lista zadataka sprinta implementirana je na sličan način kao i radna lista proizvoda, klasom „SprintBacklog“. Sprint je povezan s projektom pomoću svojstva „ProjectId“, dok su moguća stanja sprinta definirana u enumeraciji „SprintStatus“. Lista zadataka sprinta sadrži kolekcije planiranih korisničkih priča u kolekciji „Stories“, kolekciju završenih korisničkih priča „CompletedStories“ te listu zadataka „Tasks“ koja sadrži radne zadatke planiranih korisničkih priča i njihovo trenutno stanje. Moguća stanja radnog zadatka definirana su u enumeraciji „WorkTaskStatus“. Lista zadataka sprinta povezana je sa sprintom preko svojstva „SprintId“. Slika 5.5 prikazuje javno sučelje entiteta sprinta i liste zadataka sprinta.



Slika 5.5 Model entiteta sprinta i liste zadataka sprinta

Entitet radnog zadatka realiziran je klasom „WorkTask“, koji predstavlja sve vrste radnih zadataka, a tip radnog zadatka definiran je enumeracijom „WorkTaskType“, tip radnog zadatka dostupan je preko svojstva „Type“ klase „WorkTask“. Ova klasa sadrži poveznice na povezane zadatke, komentare koji pripadaju tom zadatku, a ukoliko objekt predstavlja korisničku priču ili defekt, koristi se i kolekcija „Steps“ koja sadrži korake korisničke priče ili korake za reproduciranje defekta. Komentari su definirani ugniježđenom klasom „Comments“ koja sadrži sadržaj komentara, jedinstveni identifikator autora i radnog zadatka. Entitet sastanaka realiziran je klasom „Meeting“, dok je entitet dokumenata realiziran klasom „Document“, a grupa kojoj dokument pripada definirana je enumeracijom „DocumentKind“ dostupne preko svojstva „Kind“ klase „Document“. Entiteti koji mogu biti povezani s dokumentima vezu s dokumentima pamte unutar kolekcije „Documents“ koja

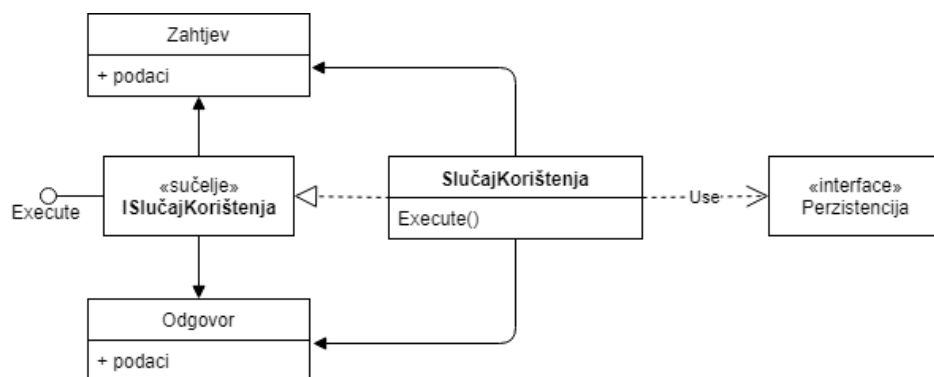
sadrži identifikatore povezanih dokumenata. Slika 5.6 prikazuje javno sučelje entiteta radnih zadataka, dokumenata i sastanaka.



Slika 5.6 Model entiteta radnih zadataka, dokumenata i sastanaka

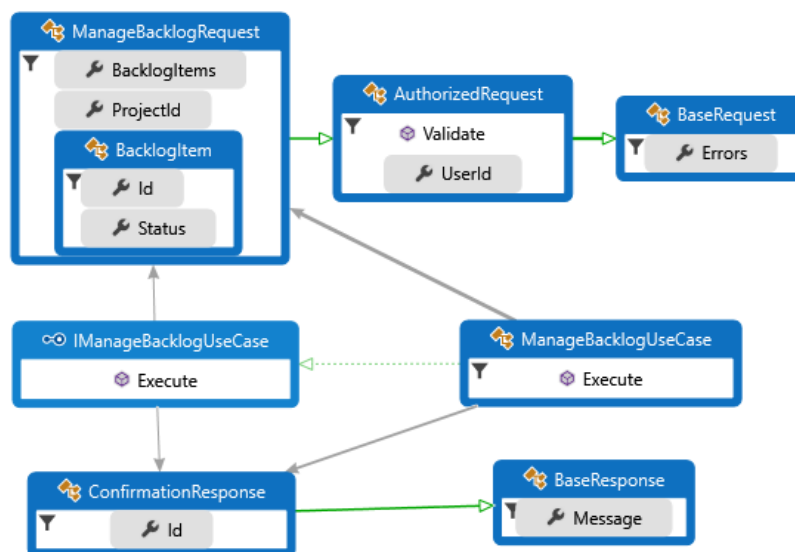
### 5.3. Slučajevi korištenja

Slučajevi korištenja sadrže aplikacijska pravila te upravljaju s entitetima, odnosno sadrže logiku kako i kada koristiti poslovna pravila definirana u entitetima. Slučaj korištenja oblikovan je oblikovnim obrascem transakcijske skripte (engl. transaction script), te definira ulazne podatke, a kao rezultat vraća izlazne podatke [11]. Ulazni i izlazni podaci u slučaj korištenja su podatkovne strukture koje ne ovise o drugim komponentama ili bibliotekama, a također ne ovise ni o entitetima, iako često dijele jednake podatkovne članove. Ove podatkovne strukture i entiteti koriste se u različite svrhe, te se mijenjaju zbog različitih razloga, dodatno kada bi sadržavale reference na entitete, ti entiteti trebali bi omogućiti stanja koja nisu dopuštena [10]. Primjer nedopuštenog stanja entiteta je radni zadatak čija je težina procijenjena na negativnu. Kako bi slučajevi korištenja mogli učitati ili spremati entitete s kojima rade, koriste sučelja za pristup perzistenciji opisana u poglavlju 5.4. Slika 5.7 prikazuje općeniti model jednog slučaja korištenja. Klijenti pojedinog slučaja korištenja ne znaju za konkretnu implementaciju slučaja korištenja, već ga koriste preko definiranog javnog sučelja.



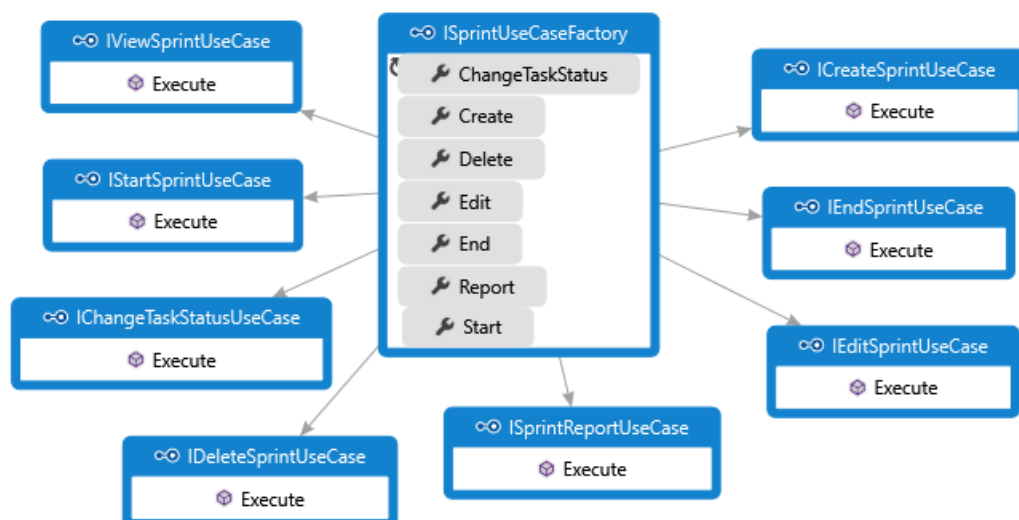
Slika 5.7 Dijagram klasa slučaja korištenja

U razvijenom rješenju sve klase koje predstavljaju ulazne podatke u slučaj korištenja nasljeđuju od zajedničke klase „BaseRequest“ ili od klase „AuthorizedRequest“, ukoliko se radi o slučaju korištenja za koji je potrebna autorizacija. Klase koje predstavljaju izlazne podatke iz slučaja korištenja naslijeđene su od klase „BaseResponse“. Slika 5.8 prikazuje model slučaja korištenja za upravljanje radnom listom proizvoda. Ostali slučajevi korištenja realizirani su na jednak način kao i prikazani slučaj korištenja.



Slika 5.8 Model slučaja korištenja za upravljanje radnom listom proizvoda

Objekte koji implementiraju sučelje slučaja korištenja, klijenti dohvaćaju preko tvornica (engl. factory), pa su na taj način skrivene stvarne implementacije slučajeva korištenja od klijenata. Slika 5.9 prikazuje sučelje tvornice koja kreira slučajeve korištenja za upravljanje sprintevima.

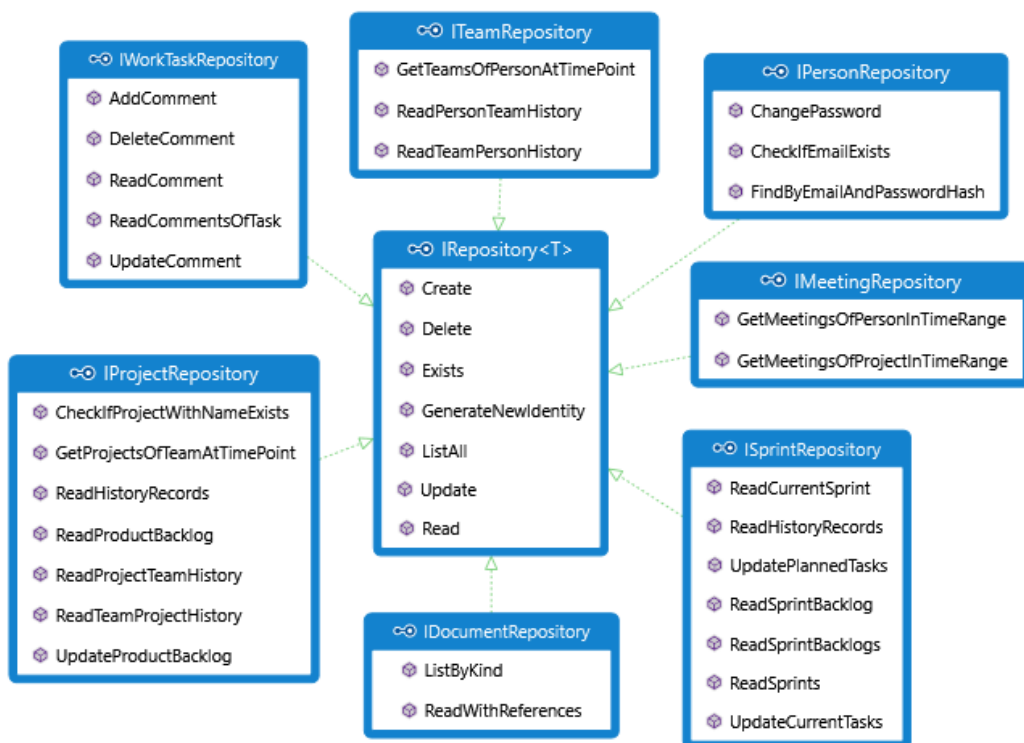


Slika 5.9 Sučelje tvornice za kreiranje slučajeva korištenja za upravljanje sprintevima

## 5.4. Perzistencija podataka

Sučelje prema perzistenciji podataka realizirano je oblikovnim obrascem repozitorija (engl. repository), čija je namjena izolacija modela domene od implementacije pristupa bazi podataka i mapiranju podataka iz baze podataka u objekte koji predstavljaju entitete. Na taj način ostvarena je jednosmjerna ovisnost između modela domene i pristupa bazi podataka [11]. U razvijenom rješenju za svaki entitet postoji odgovarajuće sučelje repozitorija, koje definira moguće upite prema perzistenciji podataka. Svi repozitoriji nasljeđuju od baznog sučelja „IRepository“ koje definira najčešće operacije i upite koje se obavljaju s entitetima. Slika 5.10 prikazuje sučelja repozitorija u razvijenom rješenju.

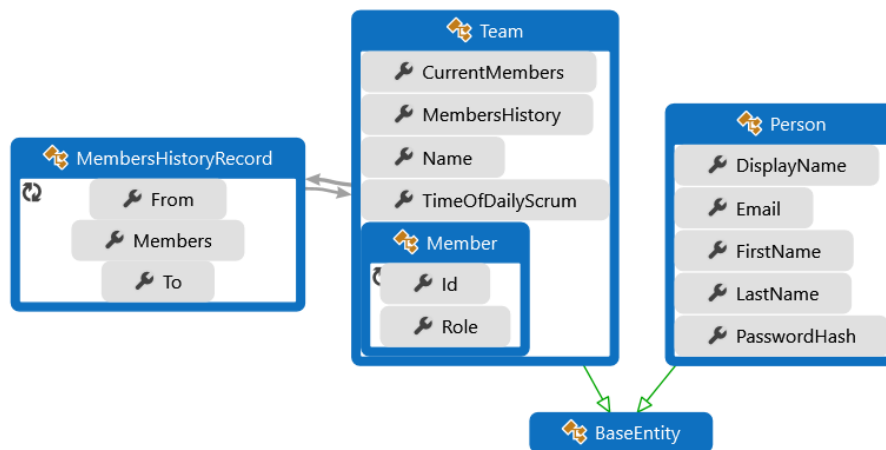




Slika 5.10 Sučelja repozitorija za pristup perzistenciji podataka

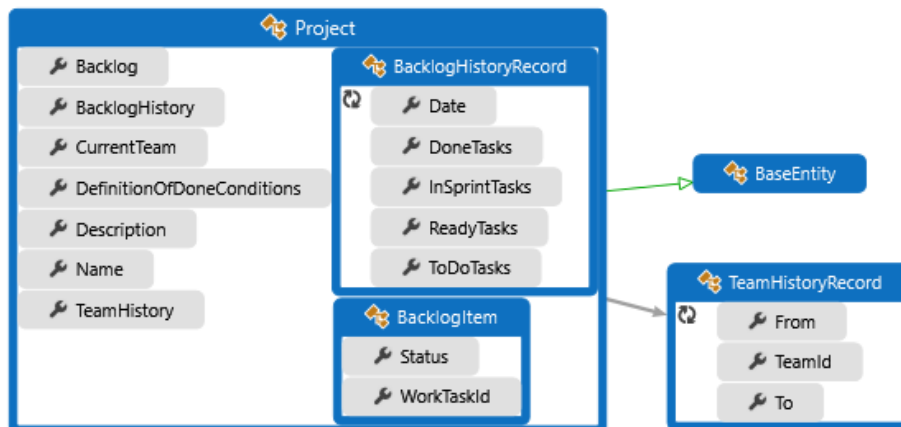
### 5.4.1. Implementacija perzistencije podataka

Perzistencija podataka implementirana je nad MongoDB [12] dokument bazom podataka. U sloju za perzistenciju podataka definiran je model dokumenata baze podataka, odnosno preslike entiteta u objekte koji se spremaju u bazu podataka. Ti objekti su čisti podatkovni objekti (engl. plain old data) te ne sadrže ni poslovnu ni aplikacijsku logiku, a popunjavaju se direktno vrijednostima koje su zapisane u bazi podataka. Slika 5.11 prikazuje model dokumenata za entitete osoba i timova, prema podacima koje sadrže model dokumenata tima razlikuje se od entiteta tima po tome da sadrži i povijest promjena članova u timu uz informaciju o trenutnim članovima.



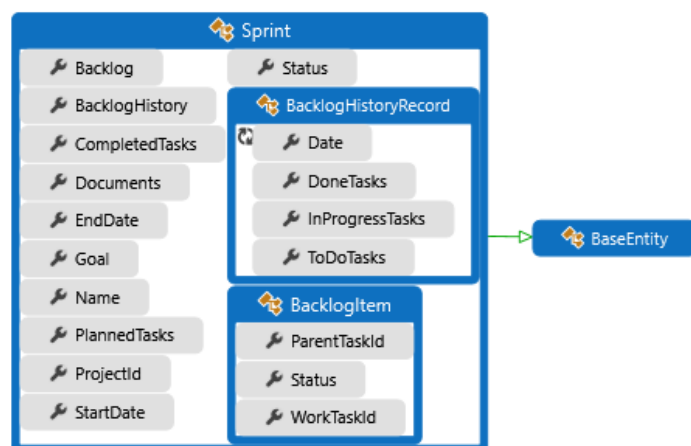
Slika 5.11 Model dokumenata osoba i timova

Slika 5.12 prikazuje model dokumenata za entitet projekta u kojem se ujedno nalaz i radna lista proizvoda i njeni statistički podaci, kao i povijest promjene timova na projektu.



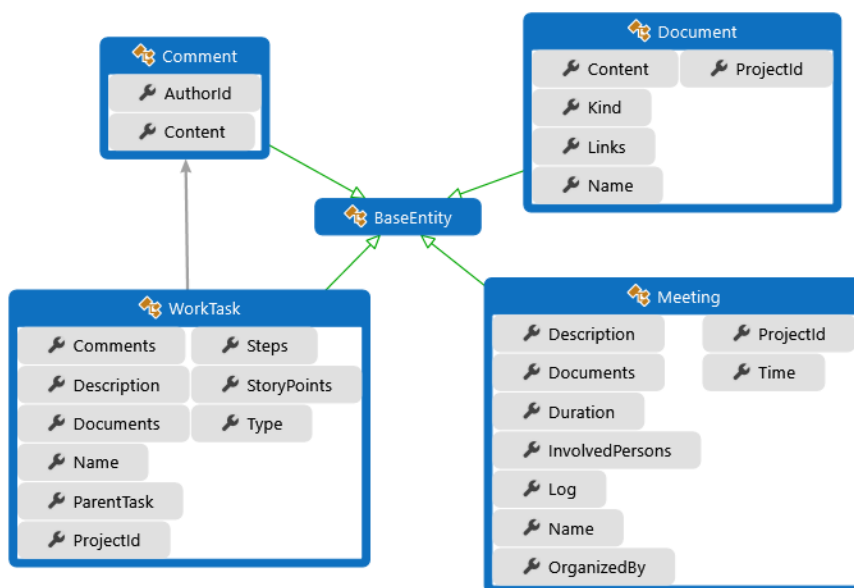
Slika 5.12 Model dokumenta projekta

Slika 5.13 prikazuje model dokumenata za entitet sprinta u kojem se uz podatke u sprintu ujedno nalaze i lista zadataka sprinta, kao i statistika o povijesti statusa zadataka iz liste zadataka.



Slika 5.13 Model dokumenata sprinta

Model dokumenata sastanaka i model dokumenta entiteta dokument jednaki su kao i modeli domene, dok se model radnih zadataka razlikuje u odnosu na model entiteta po tome što ne sadrži listu zadataka koji su povezani na taj zadatak već je spremljena samo informacija na kojeg je pojedini zadatak vezan u atributu „ParentTask“. Slika 5.14 prikazuje modele dokumenata navedena tri entiteta.



Slika 5.14 Model dokumenata radnih zadataka, sastanaka i dokumenata

Dokumenti se u bazi podataka spremaju u JSON (engl. JavaScript Object Notation – JSON) formatu. Isječak koda 5.1 prikazuje primjer dokumenta entiteta projekta u obliku u kojem je on spremljen u bazu podataka.

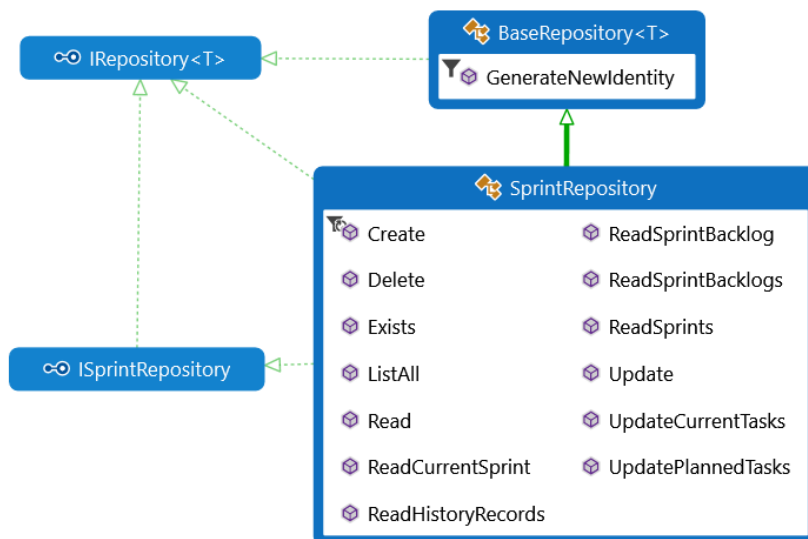
```

{
  "_id" : ObjectId("5b16246a3a80583444263ffe"),
  "Name" : "Element14",
  "Description" : null,
  "DefinitionOfDoneConditions" : [
    "Features must be tested manually before merging",
    "All automatic tests are passing",
  ],
  "CurrentTeam" : {
    "From" : ISODate("2018-06-05T05:57:22.035Z"),
    "To" : Date(253402300799999),
    "TeamId" : ObjectId("5b1626383a8058344426400e")
  },
  "TeamHistory" : [
    {
      "From" : Date(-62135596800000),
      "To" : ISODate("2018-06-05T05:57:22.034Z"),
      "TeamId" : ObjectId("5b16240f3a80583444263ffd")
    }
  ],
  "Backlog" : [
    ...
    {
      "WorkTaskId" : ObjectId("5b1624ab3a80583444264002"),
      "Status" : "Done"
    }
    ...
  ],
  "BacklogHistory" : [
    ...
    {
      "Date" : "2018-06-09",
      "ToDoTasks" : 2,
      "ReadyTasks" : 1,
      "InSprintTasks" : 0,
      "DoneTasks" : 1
    },
    ...
  ]
}

```

Isječak koda 5.1 Dokument entiteta projekta

U bazi podataka MongoDB dokumenti se spremaju u kolekcije, svaki model dokumenata ima vlastitu kolekciju koje se zovu „Persons“, „Projects“, „Teams“, „Meetings“, „Sprints“, „WorkTasks“ i „Documents“. Podacima iz tih kolekcija upravlja se u implementacijama repozitorija prikazanih na slici Slika 5.10. Klijenti sloja za perzistenciju podataka ne koriste izravno konkretne implementacije repozitorija, već ih koriste isključivo kroz definirana sučelja, a objekte koji ih implementiraju dohvaćaju na sličan način kao i objekte slučajeva korištenja preko tvornice. Slika 5.15 prikazuje model repozitorija za entitete sprinteva.



Slika 5.15 Model repozitorija za entitete sprinteva

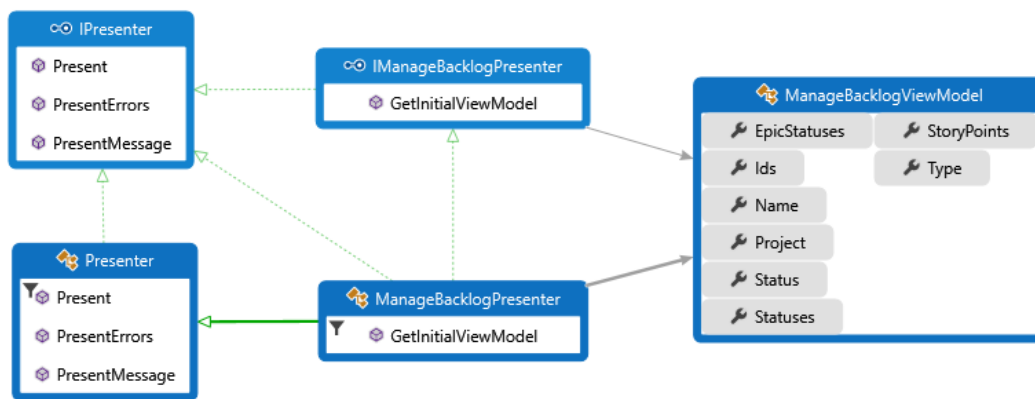
## 5.5. Prezentacija podataka

Prezentacijski sloj realiziran je nad arhitekturnim obrascem model-pogled-upravljač (engl. model-view-controller, MVC). Ovaj arhitekturni obrazac razdvaja komponente za korisničku interakciju na komponente s određenim ulogama. Model je objekt koji predstavlja neku informaciju o domeni. Pogled je komponenta odgovorna za prikazivanje informacija iz modela korisniku na korisničkom sučelju, dok je upravljač komponenta koja prihvaća korisničke zahtjeve, manipulira modelom i ažurira pogled [11]. U razvijenom sustavu, model ovog arhitekturnog obrasca realiziran je modelom domene i slučajevima korištenja. Upravljači sadrže metode koje predstavljaju moguće akcije koje korisnik može napraviti. U razvijenom sustavu postoji osam upravljača, koji grupiraju akcije prema vrsti entiteta nad kojim rade. Tako se „ProjectController“ koristi za akcije vezane uz projekte, a „SprintController“ za akcije nad sprintevima. Postoji također i „HomeController“ koji se koristi za autorizaciju korisnika na sustav. Svi upravljači u sustavu nasljeđuju od „BaseController“ klase koja sadrži pomoćne metode za javljanje poruka korisniku. Slika 5.16 prikazuje model upravljača razvijenog sustava.



Slika 5.16 Model upravljača

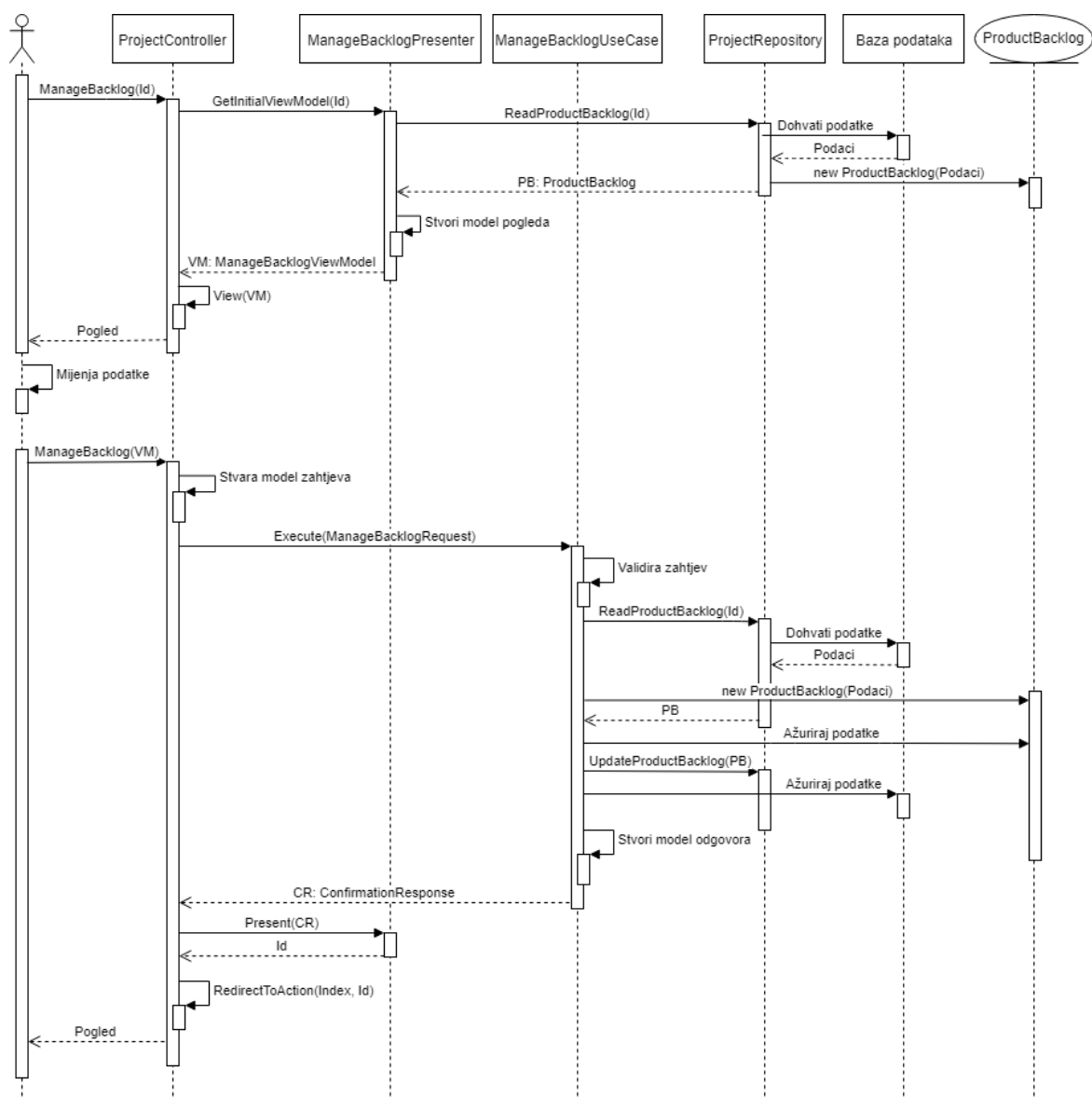
Kada upravljač zaprimi zahtjev od korisnika, te zahtjeve pretvara u model zahtjeva slučaja korištenja i poziva odgovarajući slučaj korištenja, koji su opisani u poglavlju 5.3. Nakon što upravljač dobije odgovor od slučaja korištenja prosljeđuje ga *prezenteru* (engl. presenter) koji iz dobivenog odgovora stvara model pogleda (engl. view model), te nakon toga upravljač stvar pogled od tog modela pogleda. *Prezenteri* su komponente koje prilagođavaju model odgovora slučajeva korištenja u model pogleda, u njima se nalazi logika za primjerice format datuma potreban za prikaz korisniku ili pretvorbu modela domene u model prikladan za prezentaciju korisniku. Modeli pogleda su komponente koje sadrže sve podatke koji će biti prikazani korisniku na pogledu. Većina akcija upravljača, odnosno slučajeva korištenja, ima odgovarajući par *prezentera* i modela pogleda. *Prezenteri* imaju odgovarajuće sučelje koje implementiraju, upravljači komuniciraju s *prezenterima* preko tog sučelja, kako bi se izbjegla ovisnost između upravljača i konkretne implementacije *prezentera*, a objekte koje implementiraju ta sučelja upravljači dobivaju koristeći tvornice, na sličan način kao što to rade i slučajevi korištenja s repozitorijima. Slika 5.17 prikazuje model *prezentera* i modela pogleda za slučaj korištenja upravljanja radnom listom proizvoda.



Slika 5.17 Model *prezentera* i modela pogleda za jedan slučaj korištenja

## 5.6. Primjer izvršavanja jednog slučaja korištenja

Prema dijagramu prikazanom na slici 5.1, izvršavanje jednog slučaja korištenja započinje u vanjskim slojevima prema unutarnjim te nakon toga nazad prema vanjskim slojevima. U primjeru je prikazano izvršavanje slučaja korištenja za upravljanje radnom listom proizvoda. Izvršavanje započinje tako da korisnik preko korisničkog sučelja zatraži ekran za upravljanje tom listom, nakon čega upravljač prima zahtjev s identifikatorom projekta čiju listu korisnik želi ažurirati. Upravljač tada prosljeđuje identifikator projekta *prezenteru* te zatraži da se stvori inicijalni pogled modela. *Prezenter* komunicira s repozitorijem od kojeg dobiva entitet radne liste proizvoda te ga pretvara u model pogleda koji upravljač prikazuje korisniku, nakon čega korisnik mijenja podatke na korisničkom sučelju prema tome što želi ostvariti. Kada je korisnik promijenio podatke, šalje zahtjev za spremanje. Upravljač tada prima model pogleda s promijenjenim podacima iz kojih stvara model zahtjeva za slučaj korištenja te ga prosljeđuje slučaju korištenja. Slučaj korištenja tada radi validaciju nad modelom zahtjeva, te ponovo učitava radnu listu proizvoda preko repozitorija. Slučaj korištenja upravlja podacima iz modela zahtjeva i komunicira s entitetom radne liste proizvoda koje promjene je potrebno napraviti. Nakon što slučaj korištenja promijeni sve podatke u radnoj listi proizvoda, prosljeđuje repozitoriju zahtjev za ažuriranje liste, repozitorij tada sprema ažuriranu listu u bazu podataka. Slučaj korištenja tada stvara model odgovora i vraća ga upravljaču. Upravljač dobiveni odgovor prosljeđuje *prezenteru* koji prikazuje potrebne poruke i vraća upravljaču identifikator projekta. Upravljač tada preusmjerava korisnika na pogled za pregled projekta. Slika 5.18 prikazuje dijagram slijeda izvršavanja slučaja korištenja za upravljanje radnom listom proizvoda.



Slika 5.18 Dijagram slijeda izvršavanja jednog slučaja korištenja

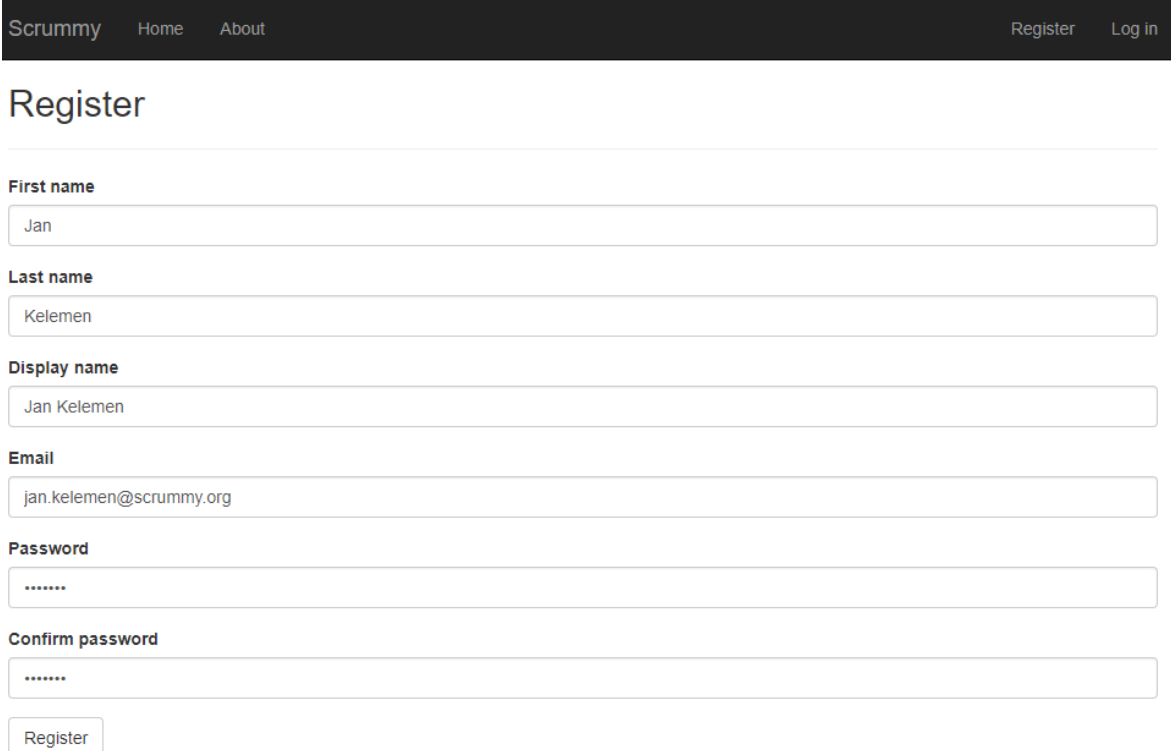


## 6. Opis funkcionalnosti razvijenog sustava

U ovome poglavlju prikazane su sve funkcionalnosti razvijenog sustava. U poglavlju 6.1 prikazana je funkcionalnost evidencije korisnika, poglavlje 6.2 prikazuje funkcionalnost evidencije timova, dok poglavlje 6.3 prikazuje funkcionalnost evidencije projekata, kao i funkcionalnosti za upravljanje sastancima, sprintevima i ostalim entitetima blisko vezanim uz entitet projekta.

### 6.1. Evidencija korisnika

Kako bi korisnik mogao početi koristiti razvijenu aplikaciju potrebno je izvršiti registraciju. Stranica za registraciju dostupna je iz navigacijske trake klikom na *Register*. Na stranici za registraciju ispunjava podatke o sebi, željeno ime koje će biti korišteno za prikazivanje, adresu elektroničke pošte i željenu lozinku. Kada korisnik završi s upisom željenih podataka klikom na *Register* predaje zahtjev za registraciju na sustav. Slika 6.1 prikazuje stranicu za registraciju korisnika. Zbog različitih uloga u Scrum timovima u sustavu trebaju postojati barem tri korisnika da bi se sustav mogao koristiti.



Scrummy Home About Register Log in

## Register

**First name**

**Last name**

**Display name**

**Email**

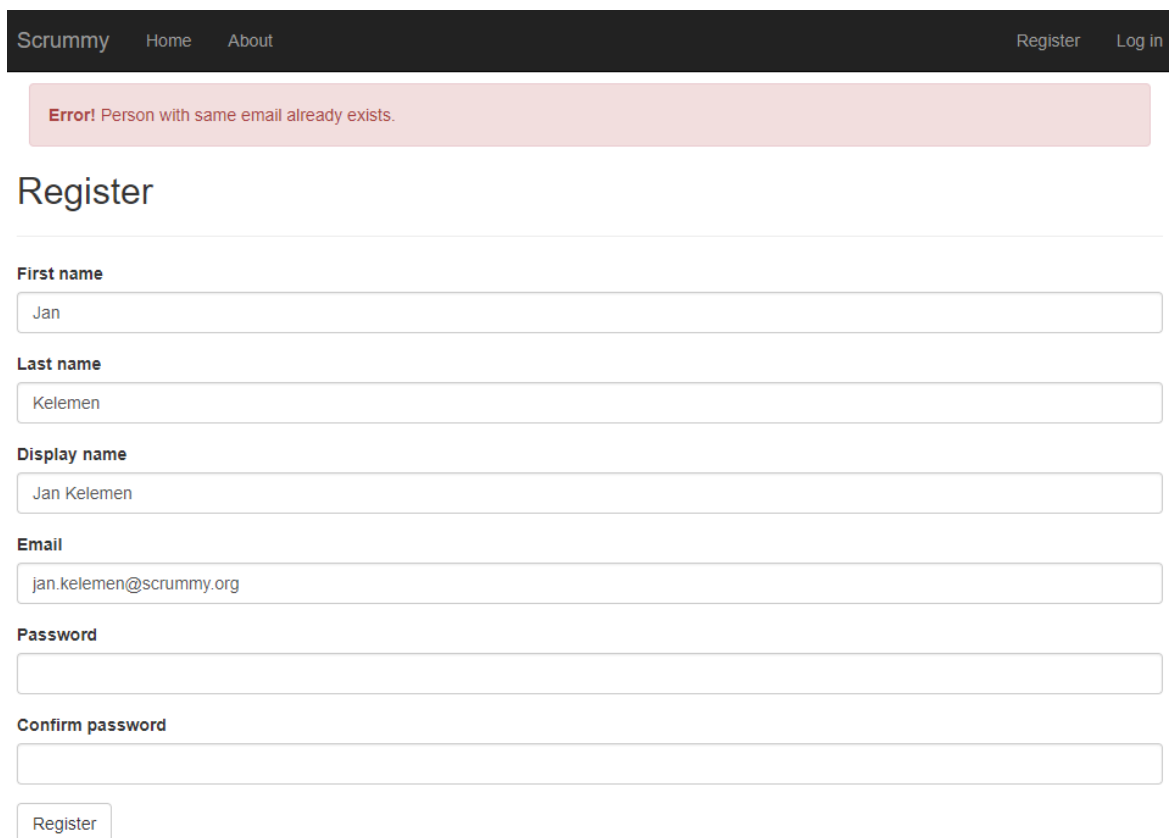
**Password**

**Confirm password**

Register

Slika 6.1 Stranica za registraciju korisnika

Korisnik neće biti u mogućnosti registrirati se, ukoliko je iskoristio adresu elektroničke pošte koju koristi neki već registrirani korisnik u sustavu, a ukoliko dođe do tog slučaja, korisnik će biti obaviješten o pogrešci. Slika 6.2 prikazuje prikaz greške tokom registracije, a u ostalim slučajevima korištenja pogreške se dojavljuju na sličan način kao što je prikazano na ovoj slici.



Scrummy Home About Register Log in

**Error!** Person with same email already exists.

## Register

**First name**

**Last name**

**Display name**

**Email**

**Password**

**Confirm password**

Register

Slika 6.2 Pokušaj registracije korisnika s već postojećom adresom elektroničke pošte

Po uspješnoj registraciji korisniku će biti prikazana poruka da je korisnik uspješno registriran te će biti preusmjeren na stranicu za prijavu na sustav. Na stranici za prijavu na sustav korisnik upisuje svoju adresu elektroničke pošte i lozinku koju je odabrao. Klikom na *Log in* korisnik se prijavljuje na sustav. Do stranice za prijavu na sustav korisnik može doći i odabirom opcije *Log in* iz navigacijske trake. Slika 6.3 prikazuje stranicu na prijavu za sustav nakon uspješne registracije korisnika, na ovoj slici vidljiv je izgled notifikacije korisniku koja potvrđuje da je neka radnja uspješno obavljena.

Scrummy Home About Register Log in

Success! Person Jan Kelemen successfully created.

## Login

Email

jan.kelemen@scrummy.org

Password

\*\*\*\*\*

Log in

Slika 6.3 Stranica za prijavu na sustav

Kada se korisnik prijavi na sustav, bit će preusmjeren na stranicu na kojoj je vidljiv njegov aktualni posao, a to je stranica na kojoj se prikazuje popis projekata na kojem rade timovi kojima on trenutno pripada i popis budućih i trenutnih sastanaka za te projekte. Na popisu projekata korisnik ima prečace do stranice za pregled radne liste proizvoda klikom na prvu ikonu, stranicu za upravljanje sprintevima projekta klikom na ikonu s kalendarom, stranicu za upravljanje sastancima projekta klikom na ikonu s pješčanim satom i stranicu za pregled detalja projekta klikom na ikonu s tri točke. Na popisu sastanaka korisniku se prikazuje ime, vrijeme početka i trajanje sastanka, a ima prečace za stranicu za ažuriranje podataka sastanka klikom na ikonu s kvadratom i olovkom i stranicu za pregled detalja sastanka klikom na ikonu s tri točke. Na svim stranicama u razvijenom rješenju postoji konvencija da ikona s tri točke vodi na pregled detalja za taj entitet, a ikona s kvadratom i olovkom vodi do stranice za uređivanje tog entiteta. Također iz navigacijske trake koja je dostupna na svim stranicama nakon prijave na sustav ima mogućnost kreiranja novih projekta i pregled popisa svih projekata u kartici *Projects*, kreiranje novih timova i pregled popisa svih timova u kartici *Teams*, pregled popisa svih korisnika u kartici *Users*, pregled vlastitog profila klikom na svoje ime i odjavu sa sustava klikom na *Log out*. Slika 6.4 prikazuje stranicu s aktualnim poslom korisnika na kojoj su vidljiva dva projekta i dva sastanka. Timovi koji je korisnik trenutno član, rade na projektima „Wild 8 Open 2“ i „ReqTrack“, a zabilježena su dva buduća sastanka, sastanak „Planning 1“ i sastanak „Review“.

## Current work

Project						
Wild 8 Open 2						
ReqTrack						
Meeting	Time	Duration				
Planning 1	2018-06-12 09:30	02:00				
Review	2018-06-11 16:00	05:00				

Slika 6.4 Stranica s aktualnim poslom nekog korisnika

Na stranici za pregled profila korisnika prikazuju se podaci o korisniku koje je unio prilikom registracije i popis trenutnih timova. Sa strane korisniku se nude opcija za promjenu lozinke klikom na *Change password*, promjenu podataka klikom na *Edit details* i izvještaj o povijesti timova u kojima je taj korisnik bio tokom vremena klikom na *Team history*. *Contact* je opcija za slanje elektroničke pošte korisniku koristeći trenutnu adresu korisnika. Opcija za promjenu lozinke i promjenu podataka vidljive su isključivo tada kada se korisnik nalazi na stranici za pregled vlastitog profila. Slika 6.5 prikazuje stranicu za pregled profila korisnika koji se trenutno nalazi u dva tima, timu „The Dalton Brothers“ i u timu „Unimatrix 01“.

Scrummy

Home

Projects ▾

Teams ▾

Users

About

Jan Kelemen

Log out

Actions

Change password

Edit details

Contact

Team history

Person - Jan Kelemen

First name

Jan

Last name

Kelemen

Display name

Jan Kelemen

Email

jan.kelemen@scrummy.org

Current teams

Team

The Dalton Brothers

Unimatrix 01

Slika 6.5 Stranica za pregled profila korisnika

Na stranici za uređivanje podataka o korisniku korisnik može uređivati svoje podatke koje je unio prilikom registracije, također kao i kod registracije, korisnik neće moći postaviti

svoju adresu elektroničke pošte na adresu koja već postoji u sustavu zabilježena za nekog drugog korisnika. Slika 6.6 prikazuje stranicu za uređivanje podataka korisnika.

The screenshot shows the 'Edit person' form for 'Jan Kelemen'. The form is titled 'Edit person - Jan Kelemen' and has a 'Cancel' button on the left. The form fields are: 'First name' (Jan), 'Last name' (Kelemen), 'Display name' (Jan Kelemen), and 'Email' (jan.kelemen@scrummy.org). There is a 'Save' button at the bottom right.

Slika 6.6 Stranica za uređivanje podataka korisnika

Podatak koji ne može promijeniti na stranici za uređivanje podataka korisnika je lozinka. Na stranici za promjenu lozinke potrebno unijeti trenutnu lozinku i dva puta ponoviti željenu novu lozinku. Slika 6.7 prikazuje stranicu za promjenu lozinke korisnika.

The screenshot shows the 'Change password' form in the Scrummy application. The form is titled 'Change password' and has a 'Cancel' button on the left. The form fields are: 'Old password' (masked with dots), 'New password' (masked with dots), and 'Confirm password' (masked with dots). There is a 'Save' button at the bottom right.

Slika 6.7 Stranica za promjenu lozinke korisnika

Na stranici za uređivanje podataka ili promjenu lozinke korisnik potvrđuje promjenu podataka klikom na *Save* ili odustaje od promjena klikom na *Cancel*, što će ga vratiti natrag na pregled profila korisnika. Stranica za pregled povijesti timova korisnika sadrži popis svih timova i uloga koje je korisnik imao od trenutka kad se podaci bilježe u sustavu. Slika 6.8

prikazuje stranicu s izvještajem o povijesti timova korisnika. Prikazani korisnik bio je član dva tima, „Unimatrix 01“ i „The Dalton Brothers“. U timu „Unimatrix 01“ bio je član razvojnog tima od 11.6.2017. do 11.6.2018., kada je dobio ulogu Scrum majstora, te i sada ima tu ulogu, dok je u timu „The Dalton Brothers“ od 25.5.2018. kao član razvojnog tima.

<div>Scrummy</div> <div> <a href="#">Home</a> <a href="#">Projects ▾</a> <a href="#">Teams ▾</a> <a href="#">Users</a> <a href="#">About</a> </div> <div> <div>Jan Kelemen</div> <div>Log out</div> </div>				
<div>Actions</div> <div>Team history of - Filip Gulan</div>				
<div>Person</div>				
Team	Roles	From	To	
Unimatrix 01	Scrum master	2018-06-11	Currently	...
The Dalton Brothers	Developer	2018-05-25	Currently	...
Unimatrix 01	Developer	2017-06-11	2018-06-11	...

Slika 6.8 Stranica s izvještajem o povijesti timova korisnika

Profili svih korisnika u sustavu dostupni su sa stranice za pregled svih korisnika kojoj se pristupa iz navigacijske trake odabirom opcije *Users*. Slika 6.9 prikazuje stranicu s popisom korisnika.

<div>Scrummy</div> <div> <a href="#">Home</a> <a href="#">Projects ▾</a> <a href="#">Teams ▾</a> <a href="#">Users</a> <a href="#">About</a> </div> <div> <div>Jan Kelemen</div> <div>Log out</div> </div>				
List of persons				
Person				
Jan Kelemen				...
Luka Križan				...
Filip Gulan				...
Hrvoje Butina				...
Glorija Markač				...
Marko Matijaščić				...
Marina Brebrić				...

Slika 6.9 Stranica s popisom korisnika

## 6.2. Evidencija timova

Timovi se koriste kako bi se stvorila grupa korisnika s njihovim ulogama koji zajedno surađuju na nekom projektu. Stranica za kreiranje timova dostupna je iz navigacijske trake odabirom opcije *Create* iz kartice *Teams*. Na stranici za kreiranje timova korisnik unosi naziv tima, vrijeme dnevnog sastanka tima te članove tima. Korisnik dodaje nove članove klikom na ikonu sa simbolom „+“ u zaglavlju tablice, a pojedini redak dodan u tablicu može maknuti klikom na ikonu sa simbolom „x“ s desne strane tog retka. Slika 6.10 prikazuje stranicu za kreiranje tima naziva „The Dalton Brothers“ koji ima vrijeme dnevnog sastanka u 9:30 u kojem se nalaze četiri člana od kojih su dva člana razvojnog tima.

Member	Role	
Jan Kelemen	Product owner	x
Luka Križan	Scrum master	x
Filip Gulan	Developer	x
Hrvoje Butina	Developer	x

Slika 6.10 Stranica za kreiranje tima

Nakon što korisnik uspješno kreira tim, bit će preusmjeren na stranicu za pregled tima. Na stranici za pregled tima dostupni su podaci u timu, popis trenutnih članova tima, te popis projekata na kojima taj tim trenutno radi. Na popisu projekata nalaze se prečaci do stranica projekata jednaki onima sa stranice za aktualni posao. Sa strane korisniku su dostupne opcije za uređivanje podataka tima klikom na *Edit*, izvještaj o povijesti promjene članova tima klikom na *Member history* i izvještaj o povijesti projekata timova klikom na *Project history*. Tim je moguće obrisati ukoliko tim nikada nije sudjelovao ni na jednom projektu ili su

projekti na kojima je taj tim sudjelovao obrisani iz sustava. Kada korisnik pristupi stranici za pregled takvog tima, bit će mu ponuđena i opcija brisanja tima koju ostvaruje klikom na *Delete*. Stranica za uređivanje podataka o timu ista je kao i stranica za kreiranje tima uz razliku što je inicijalno popunjena trenutnim podacima o timu. Slika 6.11 prikazuje stranicu za pregled detalja tima, u kojem se trenutno nalaze četiri člana i tim trenutno radi na jednom projektu.

The screenshot shows the 'Team - The Dalton Brothers' page. At the top, there is a navigation bar with links: Scrummy, Home, Projects, Teams, Users, About, and a user profile for Jan Kelemen with a 'Log out' button. The main content area is divided into two sections: 'Actions' on the left and 'Team - The Dalton Brothers' on the right. The 'Actions' section contains three buttons: 'Edit', 'Member history', and 'Project history'. The 'Team - The Dalton Brothers' section displays the team's name, 'The Dalton Brothers', and the 'Time of daily scrum', '09:30'. Below this is a table of team members with columns for 'Member' and 'Role'. The members listed are Jan Kelemen (Product owner), Luka Križan (Scrum master), Filip Gulan (Developer), and Hrvoje Butina (Developer). Each member row has a three-dot menu icon to its right. Below the members table is a section for 'Projects' with a table showing one project, 'Wild 8 Open 2'. At the bottom right of the projects table, there are four icons: a list icon, a calendar icon, a clock icon, and a three-dot menu icon.

Member	Role
Jan Kelemen	Product owner
Luka Križan	Scrum master
Filip Gulan	Developer
Hrvoje Butina	Developer

Project
Wild 8 Open 2

Slika 6.11 Stranica za pregled tima

Stranica sa izvještajem o promjeni članova tima sadrži popis svih promjena članova i njihovih uloga. Slika 6.12 prikazuje stranicu za pregled izvještaja o promjeni članova tima „Unimatrix 01“ gdje je vidljivo da je tokom povijesti tima došlo do jedne promjene, 28.6.2018. članovi Jan Kelemen i Filip Gulan su zamijenili uloge unutar tima.



Scrummy

Home

Projects

Teams

Users

About

Jan Kelemen

Log out

Actions

Member history of - Unimatrix 01

Team

Time	Member	Role	
2018-06-11 - Currently			
	Jan Kelemen	Developer	...
	Luka Križan	Developer	...
	Filip Gulan	Scrum master	...
	Hrvoje Butina	Product owner	...
2017-06-11 - 2018-06-11			
	Jan Kelemen	Scrum master	...
	Luka Križan	Developer	...
	Filip Gulan	Developer	...
	Hrvoje Butina	Product owner	...

Slika 6.12 Stranica sa izvještajem o promjeni članova tima

Stranica s izvještajem o projektima tima sadrži popis svih projekata na kojem je taj tim radio tokom vremena. Slika 6.13 prikazuje povijest projekata tima „Unimatrix 01“. Prikazani tim je od početka razvoja sustava „ReqTrack“ do sada radio na tom projektu, dok od 28.6.2018. radi i na projektu „Wild 8 Open 2“.

Scrummy

Home

Projects ▾

Teams ▾

Users

About

Jan Kelemen

Log out

Actions

Project history of - Unimatrix 01

Team

Project	From	To	
Wild 8 Open 2	2018-06-11	Currently	...
ReqTrack	Beginning	Currently	...

Slika 6.13 Stranica sa izvještajem o povijesti projekata tima

Stranica za pregled popisa svih timova dostupna je iz navigacijske trake odabirom opcije *View all* iz kartice *Teams*. Slika 6.14 prikazuje stranicu za pregled svih timova na kojoj su vidljiva tri tima, „The Dalton Brothers“, „Grupa TNT“ i „Unimatrix 01“.

## List of teams

Team	
The Dalton Brothers	<div><div></div><div>...</div></div>
Grupa TNT	<div><div></div><div>...</div></div>
Unimatrix 01	<div><div></div><div>...</div></div>

Slika 6.14 Stranica s popisom timova

### 6.3. Evidencija projekata

Korisnik može kreirati projekt odabirom opcije *Create* iz navigacijske trake pod izbornikom *Projects*. Tada će mu se prikazati stranica za kreiranje projekata na kojoj korisnik unosi ime projekta, odabire tim koji će razvijati sustav, opis projekta i unosi uvjete definicije završenog. Upravljanje uvjetima definicijom završenog slično je kao i upravljanje članovima tima na stranicama za kreiranje ili uređivanje tima, uz dodatak da je moguće promijeniti redoslijed pojedinih zapisa klikom na ikone sa strelicama koje se nalaze pored pojedinog retka. Naziv projekta mora biti jedinstven, a definicija završenog za projekt mora sadržavati barem jedan uvjet. Slika 6.15 prikazuje stranicu za kreiranje projekta, na kojoj se kreira projekt imena „Wild 8 Open 2“, koji će razvijati tim „The Dalton Brothers“ i ima četiri uvjeta u definiciji završenog.

Scrummy

Home

Projects ▾

Teams ▾

Users

About

Jan Kelemen

Log out

## Create project

Name

Wild 8 Open 2

Team

The Dalton Brothers ▾

Description

Web page for a restaurant

Definition of Done

Condition

+

Test cases for manual testing are defined

↑

↓

×

Acceptance criteria is covered by manual or automatic tests

↑

↓

×

All automatic tests are passing

↑

↓

×

Work is integrated in main development branch

↑

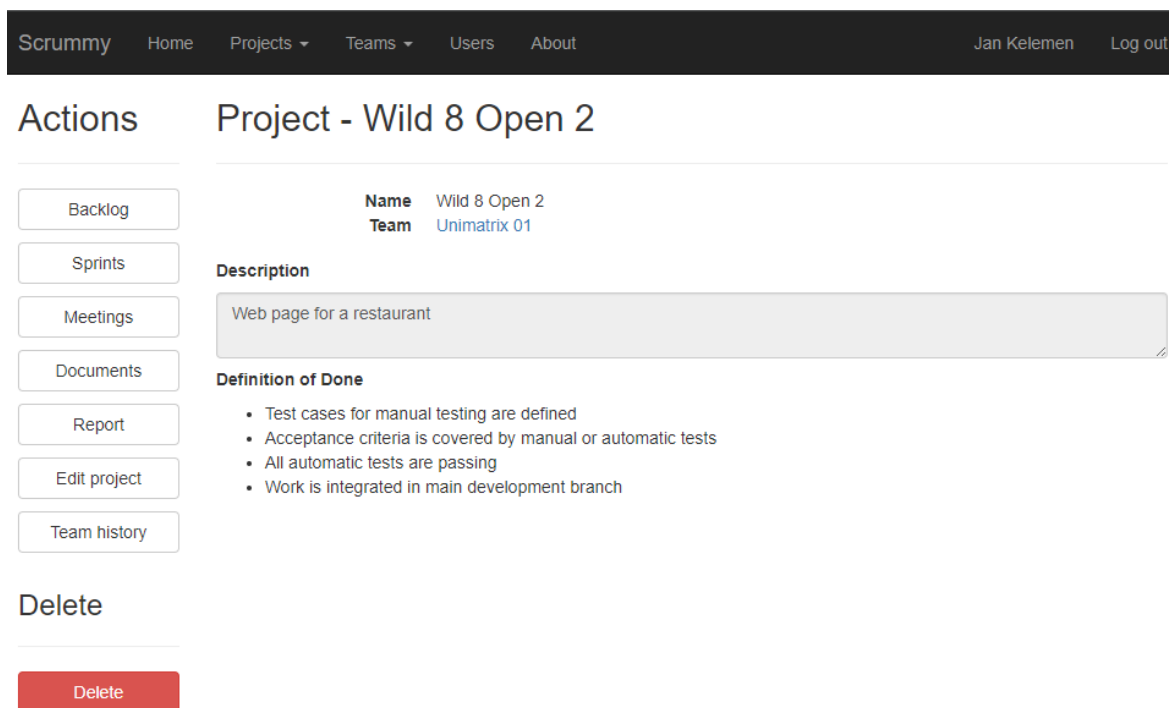
↓

×

Create

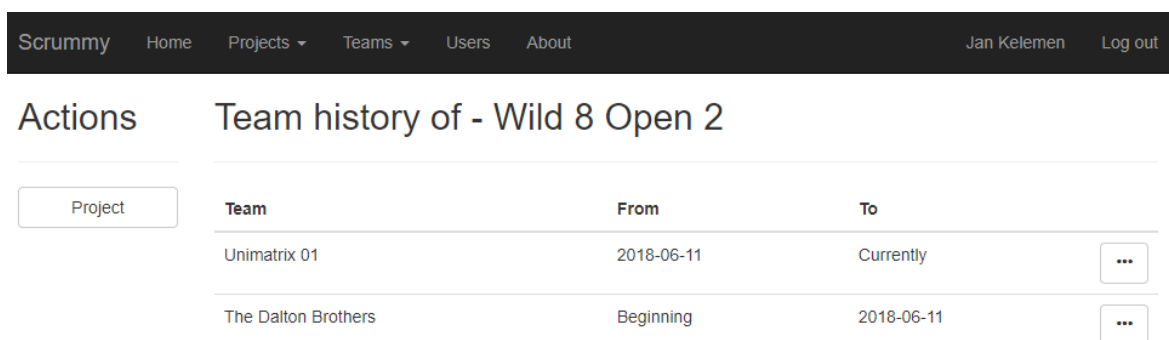
Slika 6.15 Stranica za kreiranje projekta

Kad korisnik uspješno kreira projekt, bit će preusmjeren na stranicu za pregled detalja projekta. S ove stranice korisniku su vidljivi detalji o projektu te akcije koje je moguće izvršavati za taj projekt. Dostupne su opcije: pregled radne liste proizvoda klikom na *Backlog*, upravljanje sprintevima klikom na *Sprints*, upravljanje sastancima projekta klikom na *Meetings*, upravljanje dokumentima projekta klikom na *Documents*, izvještaj za projekt klikom na *Report*, uređivanje podataka projekta klikom na *Edit project* i izvještaj o povijesti promjene timova na projektu klikom na *Team history*. Korisnik također može i obrisati projekt klikom na *Delete*, kad će mu se prikazati upozorenje želi li stvarno obrisati projekt, što će morati potvrditi prije nego što se projekt obriše. Ukoliko postoji aktualan sprint, na ovoj stranici nalazi se i ploča zadataka sprinta gdje se može upravljati statusom zadataka u i opcija za završavanje sprinta. Slika 6.16 prikazuje stranicu za pregled projekta. Stranica za uređivanje podataka o projektu ista je kao i stranica za kreiranje projekta prikazana na slici 6.15.



Slika 6.16 Stranica za pregled detalja projekta

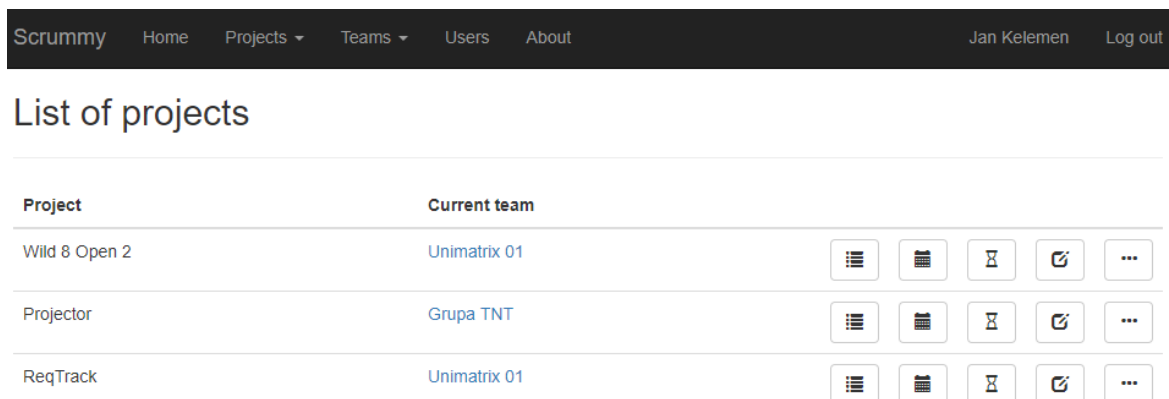
Stranica za pregled povijesti timova projekta sadrži popis svih timova koji su sudjelovali na razvoju sustava. Slika 6.17 prikazuje stranicu za pregled povijesti timova projekta „Wild 8 Open 2“ na kojem je razvoj započeo tim „The Dalton Brothers“, a 11.6.2018. razvoj je preuzeo tim „Unimatrix 01“. Tim koji radi na sustavu moguće je promijeniti preko stranice za uređivanje projekta.


















Slika 6.17 Stranica za pregled povijesti timova projekta

Stranica za pregled popisa svih projekata dostupna je iz navigacijske trake odabirom opcije *View all* iz kartice *Projects*. Uz prečace vezane uz projekt koji se prikazuju na stranici aktualnog posla, na ovoj stranici postoji i prečac do stranice za uređivanje projekta. Na ovoj stranici prikazuje se ime projekta i tim koji trenutno radi na razvoju sustava. Slika 6.18 prikazuje stranicu za pregled svih projekata na kojoj su vidljiva tri projekta „Wild 8 Open

2“ na kojemu radi tim „Unimatrix 01“, projekt „Projector“ na kojemu radi tim „Grupa TNT“ i projekt „ReqTrack“ na kojemu radi tim „Unimatrix 01“.



The screenshot shows the 'List of projects' page in the Scrummy application. The header includes navigation links (Home, Projects, Teams, Users, About) and user information (Jan Kelemen, Log out). The main content is a table with two columns: 'Project' and 'Current team'. Each row represents a project and includes a set of action icons on the right.

Project	Current team	
Wild 8 Open 2	Unimatrix 01	    
Projector	Grupa TNT	    
ReqTrack	Unimatrix 01	    

Slika 6.18 Stranica s popisom svih projekata

### 6.3.1. Upravljanje radnim zadacima

Do stranice za pregled radne liste proizvoda, korisnik dolazi klikom na *Backlog* sa stranice za pregled detalja projekta ili odabirom prečaca do te stranice koji se nalazi na stranicama za pregled aktualnog posla ili s popisa svih projekata. Na ovoj stranici vidljivo je trenutno stanje radne liste proizvoda, koja sadrži popis zadataka poredan po prioritetu. Za svaki zadatak unutar ove liste vidljivo je ime zadatka, tip zadatka, procijenjena težina, status zadatka, te u slučaju da se radi o korisničkoj priči, ep kojem je ta korisnička priča dodijeljena, ukoliko je ona definirana. Pored popisa korisniku se nudi mogućnost upravljanja radnom listom proizvoda klikom na *Manage backlog*, odnosno promjena prioriteta ili promjena statusa i mogućnost pregleda popisa završenih radnih zadataka projekta klikom na *Completed*. Ovdje se također nalaze i poveznice na stranice za kreiranje pojedine vrste radnog zadatka. Stranica za popis završenih radnih zadataka jednaka je stranici za pregled radne liste proizvoda, samo što se na njoj prikazuju završeni radni zadaci. Slika 6.19 prikazuje radnu listu proizvoda u kojoj se nalaze dva epa, sedam korisničkih priča, jedan defekt i jedan zadatak. Dvije korisničke priče imaju status *Ready*, odnosno spreman, a ostali radni zadaci imaju status *ToDo*, što znači da ih je potrebno detaljnije definirati.

Scrummy
Home
Projects ▾
Teams ▾
Users
About
Jan Kelemen
Log out

Manage backlog

Filter

Completed

Create

Epic

Story

Task

Defect

Backlog of Wild 8 Open 2

Task	Type	Story Points	Status	Parent	
Register staff member	User Story	5	Ready	Restaurant Management	...
Change restaurant data	User Story	3	Ready	Restaurant Management	...
Restaurant Management	Epic		ToDo		...
User can't login	Defect		ToDo		...
Meal management	Epic		ToDo		...
Define order handling	User Story		ToDo		...
View menu	User Story		ToDo		...
Add new meal	User Story		ToDo	Meal management	...
Change list of condiments	User Story		ToDo	Meal management	...
Define loyalty program	Task		ToDo		...
Remove staff member	User Story		ToDo	Restaurant Management	...

Slika 6.19 Stranica za pregled radne liste proizvoda

Klikom na *Manage backlog* korisnik odabire opciju upravljanja radnom listom proizvoda, na ovoj stranici moguće mijenjati status radnog zadatka odabirom statusa iz padajućeg izbornika i mijenjati prioritet klikom na ikone sa strelicama. Opcija za upravljanje radnom listom proizvoda vidljiva je samo korisnicima koji su članovi tima koji trenutno razvija taj sustav. Završetak uređivanja korisnik potvrđuje klikom na *Save* koji se nalazi na dnu stranice, a može i odustati od uređivanja klikom na *Cancel* koji se nalazi pored popisa, u tom slučaju korisnik će biti vraćen natrag na pregled radne liste proizvoda. Slika 6.20 prikazuje stranicu za upravljanje radnom listom proizvoda.

Actions

Manage backlog - Wild 8 Open 2

Cancel

Task	Type	Story Points	Status		
Register staff member	User Story	5	Ready ▾	↑	↓
Change restaurant data	User Story	3	Ready ▾	↑	↓
Restaurant Management	Epic		ToDo ▾	↑	↓
User can't login	Defect		ToDo ▾	↑	↓
Meal management	Epic		ToDo ▾	↑	↓
Define order handling	User Story		ToDo ▾	↑	↓
View menu	User Story		ToDo ▾	↑	↓
Add new meal	User Story		ToDo ▾	↑	↓
Change list of condiments	User Story		ToDo ▾	↑	↓
Define loyalty program	Task		ToDo ▾	↑	↓
Remove staff member	User Story		ToDo ▾	↑	↓

Save

Slika 6.20 Stranica za upravljanje radnom listom proizvoda

Odabirom jedne od opcije za kreiranje radnog zadatka korisnika se preusmjerava na stranicu za kreiranje tog tipa radnog zadatka, stranice za kreiranje pojedinih tipova radnih zadataka su međusobno slične uz razliku da neki tipovi radnih zadataka imaju dodatne podatke koje je moguće unijeti. Tako primjerice za ep nije moguće definirati povezani radni zadatak jer u hijerarhiji zadataka ne postoji tip radnog zadatka koji obuhvaća veći opseg od epa, ili za zadatke i defekte nije moguće unijeti povezane zadatke jer ne postoji tip radnih zadataka koji obuhvaća manji opseg od ta dva tipa. Dok za korisničke priče i defekte postoji mogućnost unošenja koraka potrebnih za realizaciju korisničke priče ili reprodukciju defekta. Konačna procjena težine zadatka odgovornost je razvojnog tima, a prilikom kreiranja radnog zadatka to polje može ostati neispunjeno. Slika 6.21 prikazuje stranicu za kreiranje epa naziva „Restaurant Management“ koji ima procijenjenu težinu od 42 boda, jednu povezanu korisničku priču „Register staff members“ i jedan povezanu dokument „SRS“.

Scrummy

HomeProjectsTeamsUsersAbout

Jan KelemenLog out

## Create Epic - Wild 8 Open 2

Name

Restaurant Management

Story points

42

Description

Epic for grouping stories which handle restaurant management

Linked tasks

Task

Register staff member

+

×

Documents

Document

SRS

+

×

Create

Slika 6.21 Stranica za kreiranje epa

Po uspješnom kreiranju radnog zadatka korisnik se preusmjerava na stranicu za pregled radnog zadatka, na kojoj se nalaze detalji o radnom zadatku uneseni prilikom kreiranja radnog zadatka i komentari vezani uz taj radni zadatak. Pored pregleda radnog zadatka korisnik ima mogućnost odlaska na stranicu za uređivanje radnog zadatka klikom na *Edit*. Opcije za kreiranje novih radnih zadataka koji su na neki način povezani s tim zadatkom i opcija za kreiranje radnog zadatka istog tipa kao i onaj koji se trenutno pregledava. Ovdje je dostupna i opcija za brisanje radnog zadatka klikom na *Delete*. Opcije za kreiranje povezanih radnih zadataka, uređivanje ili brisanje radnog zadatka nisu uvijek dostupne, nekima od njih nije moguće pristupiti ukoliko je taj ili roditeljski zadatak već završen tokom razvoja sustava ili je već planiran u nekome od sprinteva. Ograničenje na uređivanje i brisanje zadataka se koristi kako se ne bi narušila povijest razvoja sustava. Slika 6.22 prikazuje stranicu za prikaz korisničke priče koja ima četiri koraka, tri povezana zadatka, jedan povezani dokument, te jedan komentar. Korisnik može dodati komentar na svaki tip radnog zadatka na dnu stranice za pregled radnog zadatka klikom na *Add*, nakon čega će se njegov komentar prikazivati tokom pregleda tog radnog zadatka.



## Actions

Edit

## Create

Parent Epic

Linked Task

Linked Defect

Related Story

Story

## Delete

Delete

## User Story - Register staff member

Name	Register staff member
Project	Wild 8 Open 2
Parent task	Restaurant Management
Story points	5

### Description

Register a new staff member and store his results to database

### Steps

#### Step

Owner fills out a form with staff member's data

Owner confirms the entered data

Server checks validity of given data

Server stores the results to the database and notifies the member via email

### Linked tasks

#### Task

Support persistence structure

...

Support businesses checks

...

Create UI

...

### Documents

#### Document

SRS



...

### Comments

#### Author

Jan Kelemen



✕

It should be investigated if every exceptional case is covered

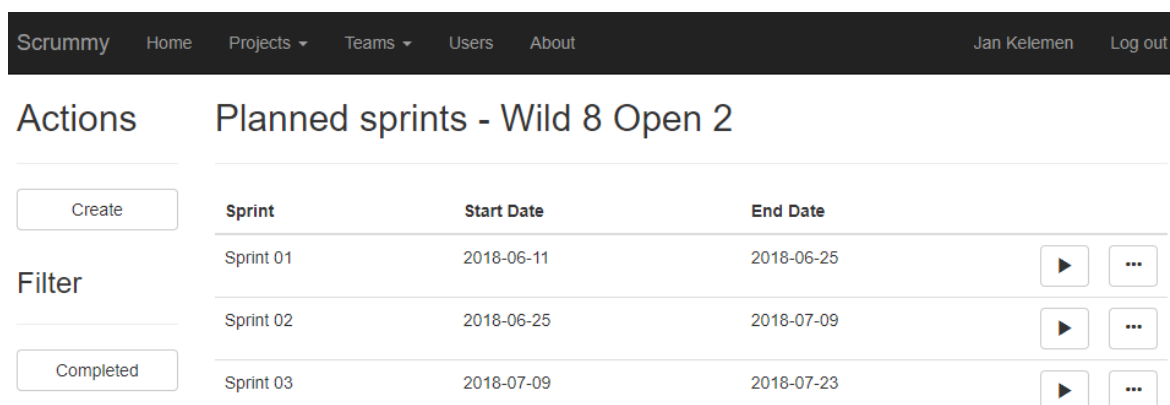
## Add comment

Add

Slika 6.22 Stranica za prikaz korisničke priče

### 6.3.2. Upravljanje sprintevima

Sprintevima se upravlja na sličan način kao i s radnom listom proizvoda, korisnik može pristupiti stranici za upravljanje sprintevima sa stranice za pregled detalja projekta klikom na *Sprints* ili odabirom prečaca sa stranice s aktualnim poslom ili popisa svih projekata. Korisniku se tada prikazuje popis planiranih sprinteva za određeni projekt. U popisu sprinteva za svaku stavku prikazuje se naziv, planirani datum početka i planirani datum završetka sprinta. Korisnik može pokrenuti pojedini sprint klikom na ikonu s „▶“ simbolom. Pokretanje sprinta znači da će se odabrani sprint postaviti kao sprint koji je trenutno u toku za taj projekt, ova opcija dostupna je isključivo kada za projekt trenutno nije definiran aktualan sprint. Na ovoj stranici nalazi se i mogućnost pregleda popisa završenih sprinteva projekta klikom na *Completed*. Stranica za pregled popisa završenih sprinteva slična je stranici za pregled planiranih sprinteva. Slika 6.23 prikazuje stranicu za pregled planiranih sprinteva na kojoj su vidljiva tri planirana sprinta, naziva „Sprint 01“, „Sprint 02“ i „Sprint 03“.



Actions		Planned sprints - Wild 8 Open 2		
Create		<b>Sprint</b>	<b>Start Date</b>	<b>End Date</b>
Filter		Sprint 01	2018-06-11	2018-06-25
		Sprint 02	2018-06-25	2018-07-09
		Sprint 03	2018-07-09	2018-07-23
Completed				

Slika 6.23 Stranica za pregled planiranih sprinteva

Do stranice za kreiranje novih sprinteva korisnik dolazi klikom na *Create* sa stranice za pregled planiranih ili završenih sprinteva. Na ovoj stranici korisnik unosi željeni naziv, planirani datum početka, planirani datum završetka, cilj sprinta, planirane korisničke priče i dokumente povezane uz taj sprint. Prilikom odabira korisničkih priča korisniku su dostupne samo one korisničke priče koje su u radnoj listi proizvoda bile proglašene spremnima, odnosno imale status *Ready*. Korisničke priče unutar sprinta moguće je poredati po prioritetu. Slika 6.24 prikazuje stranicu za kreiranje sprinta s dvije planirane korisničke priče i jednim povezanim dokumentom.

Scrummy

Home

Projects ▾

Teams ▾

Users

About

Jan Kelemen

Log out

## Create sprint - Wild 8 Open 2

**Name**

Sprint 01

**Start date**

2018-06-11

**End date**

2018-06-25

**Goal**

Create initial infrastructure and demo of functionality

**Planned stories**

**User Story**

+

Register staff member

▾

↑

↓

×

Change restaurant data

▾

↑

↓

×

**Documents**

**Document**

+

Iteration calculation - 10.06.2018.

▾

×

Create

Slika 6.24 Stranica za kreiranje sprinta

Kada korisnik uspješno kreira sprint, preusmjerava se na stranicu za pregled sprinta, na toj stranici nalaze se svi podaci o sprintu, kao i popis korisničkih priča tog sprinta s pločom zadataka. Opcija uređivanja sprinta dostupna je ukoliko korisnik pregledava stranicu planiranog sprinta, a ako pregledava stranicu aktualnog ili završenog sprinta projekta, dostupan mu je izvještaj o tom sprintu klikom na *Report*. Pored pregleda sprinta dostupna mu je i opcija za uređivanje sprinta. Stranica za uređivanje sprinta slična je stranici za kreiranje sprinta. Slika 6.25 prikazuje stranicu za pregled aktualnog sprinta projekta koji ima dvije korisničke priče od koja svaka ima po tri zadatka. Prva korisnička priča ima jedan zadatak koji čeka izvršavanje i dva koji se trenutno izvršavaju, a druga ima jedan zadatak koji čeka izvršavanje, jedan koji se trenutno izvršava i jedan koji je završen.

Scrummy
Home
Projects
Teams
Users
About
Jan Kelemen
Log out

Actions

Sprint - Sprint 01

Report

Name
Project
Start date
End date

Sprint 01
Wild 8 Open 2
2018-06-11
2018-06-25

Goal

Create initial infrastructure and demo of functionality

Stories

Story	To Do	In Progress	Done	Completed
Register staff member (5)				No
	Support persistence structure	Support businesses checks		
		Create UI		
Change restaurant data (3)				No
	AT	Create operation	Test	

Documents

Document

Iteration calculation - 10.06.2018.

Slika 6.25 Stranica za pregled sprinta

Stranica s izvještajem sprinta sadrži podatke o sprintu, prikazuje statističke podatke o korisničkim pričama i pridruženim zadacima, graf sagorijevanja na kojem se nalaze podaci o nezavršenim zadacima te graf koji prikazuje podatke o završenim zadacima tokom sprinta. Slika 6.26 prikazuje izvještaj aktualnog sprinta projekta koji ima dvije korisničke priče od kojih svaka ima tri zadatka. Prema statistikama trenutno je završen jedan zadatak, a dva još nisu započeta (zadaci koji se trenutno izvršavaju nisu uključeni u ove statistike).

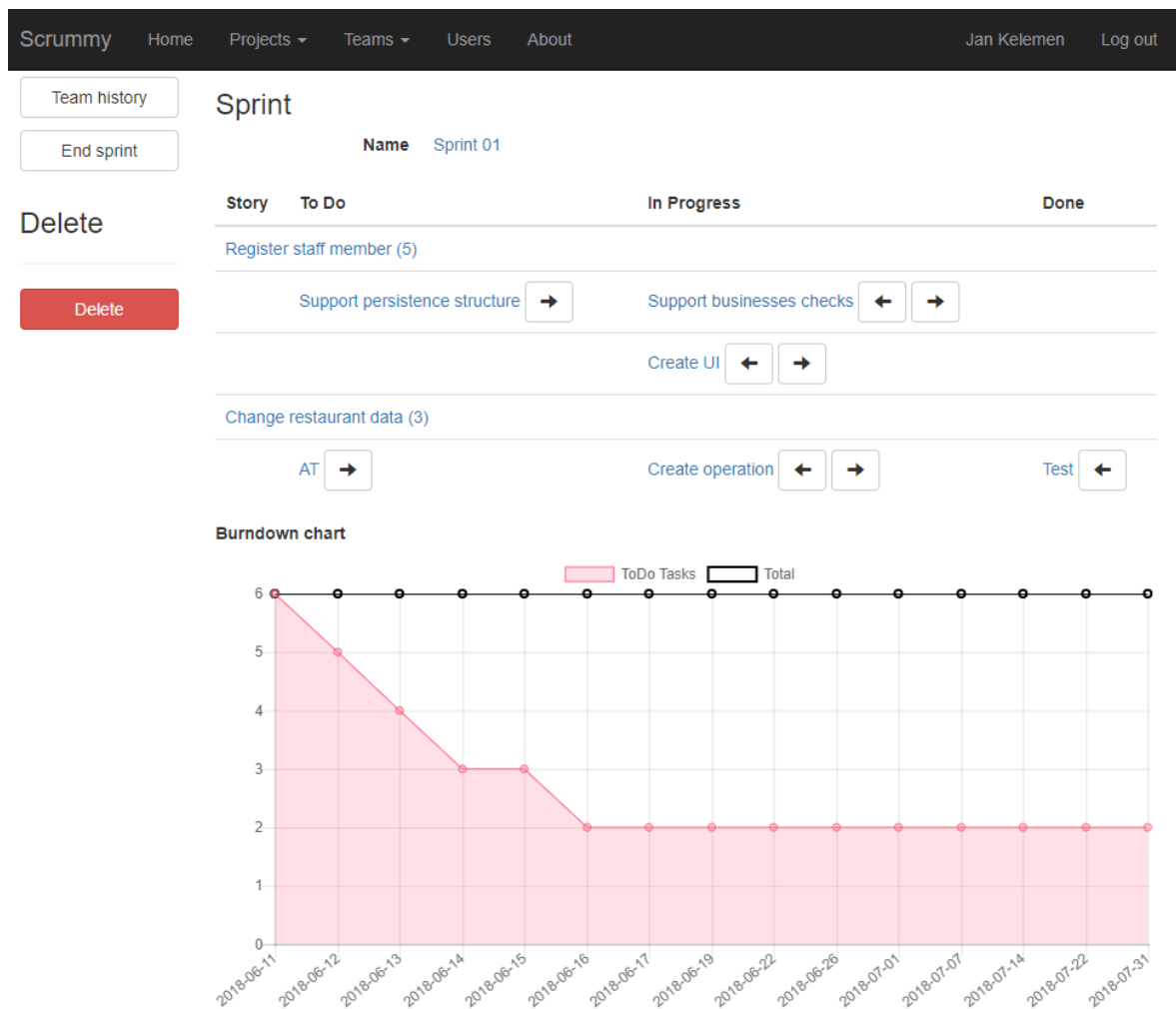
Actions

Sprint report - Sprint 01



Slika 6.26 Stranica s izvještajem sprinta

Kada na projektu postoji aktualan sprint, na dnu stranice za pregled detalja projekta nalaze se podaci o aktualnom sprintu. Prikazuje se ploča zadataka sprinta na kojoj korisnik može upravljati statusom pojedinih zadataka u sprintu te se prikazuje graf sagorijevanja. Korisnik mijenja status pojedinog zadatka klikom na ikone sa strelicama koje se nalaze pored imena tog radnog zadatka. Slika 6.27 prikazuje isječak sa stranice za pregled projekta s aktualnim sprintem. Na ploči zadataka nalaze se dvije korisničke priče od kojih svaka ima tri zadatka.



Slika 6.27 Stranica za prikaz projekta s aktualnim sprintem

Kada na projektu postoji aktualan sprint na stranici za pregled detalja projekta dostupna je opcija za završetak sprinta klikom na *End sprint*, na toj stranici korisnik odabire što će se dogoditi sa pričama koje su bile planirane u sprintu. Moguća su dva odabira, ako korisnik odabere *Done* priča će biti proglašena završenom, a ako odabere *Backlog*, priča će se vratiti u radnu listu proizvoda. Korisnik potvrđuje svoju odluku klikom na *Save* ili odustaje od završetka sprinta klikom na *Cancel*. Slika 6.28 prikazuje stranicu za završetak sprinta na

kojoj je korisnik odabrao da će se jedna priča proglasiti završenom, a druga će biti vraćena u radnu listu proizvoda.

The screenshot shows the 'End sprint - Sprint 01' interface. At the top, there is a navigation bar with 'Scrummy', 'Home', 'Projects', 'Teams', 'Users', and 'About'. On the right, it says 'Jan Kelemen' and 'Log out'. Below the navigation bar, the title 'End sprint - Sprint 01' is displayed. The main area contains a table with two columns: 'User Story' and 'Status'. There are two rows of user stories: 'Register staff member' with status 'Done' and 'Change restaurant data' with status 'Backlog'. A 'Cancel' button is on the left, and a 'Save' button is at the bottom.

User Story	Status
Register staff member	Done
Change restaurant data	Backlog

Slika 6.28 Stranica za završetak sprinta

Osim izvještaja o sprintu moguće je generirati i izvještaj o projektu. Izvještaj o projektu sadrži općenite statistike o broju zadataka povezanih s projektom tokom vremena i završenim sprintevima. Graf *Planned vs Completed* prikazuje statistike za najviše zadnjih deset sprinteva, a na njemu se može vidjeti koliko je od procijenjene ukupne težine korisničkih priča u sprintu bilo završeno, ovaj graf može se koristiti kao procjena artefakta brzine tima. Graf *Value burnup chart* prikazuje kumulativnu ostvarenu vrijednost kroz zadnjih deset sprinteva. Slika 6.29 prikazuje izvještaj projekta gdje su tokom razvoja sustava završena tri sprinta, na početku izvještaja pokazuju se statistike o broju zadataka u radnoj listi proizvoda, nakon čega slijedi izvještaj o sprintevima. Iz navedenog izvještaja može se vidjeti da u drugome sprintu nije bila završena jedna od planiranih korisničkih priča.

Actions

Project report - Wild 8 Open 2

Project

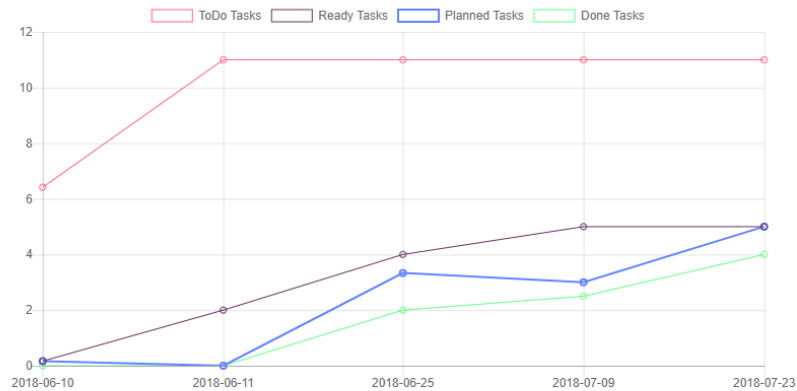
Project

Wild 8 Open 2

Completed sprints

3

Backlog statistics



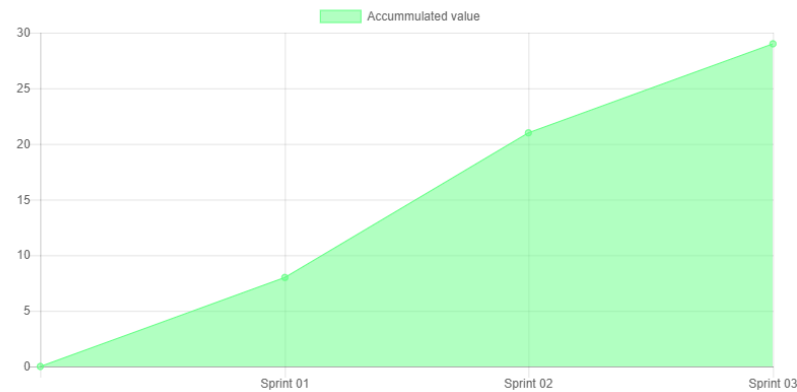
Sprints

Sprint	Planned stories	Planned value	Completed stories	Completed value	
Sprint 01	2	8	2	8	...
Sprint 02	2	18	1	13	...
Sprint 03	1	8	1	8	...

Planned vs Completed



Value burnup chart

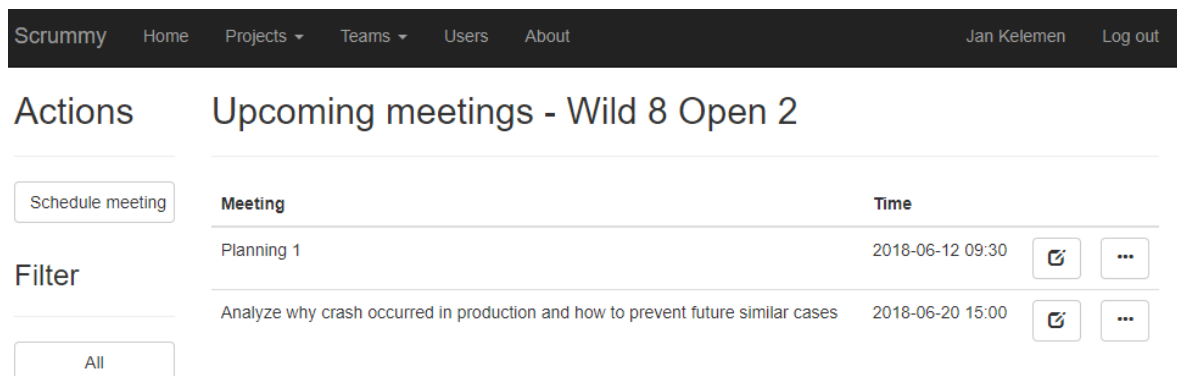


Slika 6.29 Stranica s izvještajem projekta



### 6.3.3. Upravljanje sastancima

Odabirom opcije *Meetings* sa stranice za pregled detalja projekta korisnik otvara stranicu na kojoj se prikazuju budući sastanci za taj projekt. S te stranice može odabrati opciju za kreiranje novih sastanaka odabirom *Schedule meeting* ili pregled svih sastanaka projekta odabirom filtera *All*. Stranica za pregled svih sastanaka projekta slična je stranici za pregled budućih sastanaka projekta. Slika 6.30 prikazuje stranicu za pregled budućih sastanaka projekta, a prikazana su dva zakazana sastanka.



Slika 6.30 Stranica za pregled budućih sastanaka za projekt

Odabirom opcije za kreiranje novog sastanka korisniku se otvara stranica za kreiranje sastanka na kojoj upisuje ime, datum održavanja, vrijeme trajanja, opis sastanka. Korisnik odabire osobe koje su pozvane na taj sastanak. Moguće je povezati i neki od postojećih dokumenta projekta uz sastanak koji će se kreirati. U polja za opis sastanka i polje za zapisnik sa sastanka korisnik može upisati tekst proizvoljne duljine. Stvaranjem sastanka kreirani sastanak automatski će se prikazivati osobama koje su pozvane na sastanak na stranici aktualnog posla kada se prijave na sustav. Slika 6.31 prikazuje stranicu za kreiranje sastanka na koji su pozvane četiri osobe, uz sastanak je povezan i jedan dokument. Stranice za pregled pojedinog sastanka i stranica za uređivanje sastanaka, slične su kao i odgovarajuće stranice za druge entitete koji su opisani u prethodnim poglavljima.

Scrummy
Home
Projects
Teams
Users
About
Jan Kelemen
Log out

## Create meeting - Wild 8 Open 2

**Name**

**Time**

**Duration**

**Description**

**Persons**

Person

Jan Kelemen

×

Luka Križan

×

Filip Gulan

×

Hrvoje Butina

×

**Documents**

Document

Meeting Log - 11.06.2018

×

**Log**

Create

Slika 6.31 Stranica za kreiranje sastanaka

### 6.3.4. Pregled dokumenata

Odabirom opcije *Documents* sa stranice za pregled detalja projekta korisnik dolazi do stranice s popisom dokumenata, na ovoj stranici nalazi se popis dokumenta te mogućnost kreiranja novih dokumenata. U izborniku pored popisa korisnik može primijeniti filter nad popisom te odabrati želi li da mu se prikazuju dokumenti zajednički za sve entitete, dokumenti vezani uz određenu vrstu entiteta. Slika 6.32 prikazuje stranicu s popisom svih dokumenata koji trenutno pripadaju projektu. Na popisu je vidljivo da trenutno ima pet

dokumenata.

Scrummy

Home

Projects ▾

Teams ▾

Users

About

Jan Kelemen

Log out

Actions

Create

Filter

Common

Project

Work task

Meeting

Sprint

All documents - Wild 8 Open 2

Document	Type		
Crash Dump	Common		
Meeting Log - 11.06.2018	Meeting		
Iteration calculation - 10.06.2018.	Sprint		
SRS	Work task		
Project motivation	Project		

Slika 6.32 Stranica za pregled popisa dokumenata

Korisnik stvara novi dokument odabirom opcije *Create* s prethodne stranice na kojoj korisnik unosi naziv dokumenta, sadržaj dokumenta, vanjske poveznice te odabire tip dokumenta, odnosno je li taj dokument vezan uz određenu vrstu entiteta ili je zajednički za sve. U polje sa sadržajem dokumenta korisnik može unijeti tekst proizvoljne dužine. Slika 6.33 prikazuje stranicu za kreiranje dokumenta gdje korisnik stvara dokument naziva „Project motivation“, koji pripada samo projektu i ima jednu vanjsku poveznicu.

Scrummy
Home
Projects ▾
Teams ▾
Users
About
Jan Kelemen
Log out

## Create document - Wild 8 Open 2

**Name**

**Document type**

Project ▾

**Content**

Project name - Wild 8 Open 2  
Ordered by - Faculty of Electrical Engineering and Computing

The purpose of the Wild 8 Open 2 project is to provide support for restaurant operations in the form of a web application that includes inventory management, meal order and restaurant management.

The project's goal is to develop a general solution for restaurant business support that can later be adapted or upgraded for the needs of a particular restaurant business. This project will make it easier for the restaurant business so far to automate certain administrative tasks and speed up the ordering and delivery process of restaurants.

Possible risk is completing tasks within a given time, this risk will be reduced by defining project control points that are in line with project deadlines. There is only one co-located team on the project, so there is no risk of difficult communication between team members, but there is a risk of computer equipment failure used for project development, there is no way to reduce this risk, but its impact will be mitigated by using source code management systems for the accompanying material of the project.

The final product will be a database system and a web portal, with documented installation and use of the system.

The success of the project is proportional to how many restaurants will accept the use of a developed information system and how it will speed up or facilitate the restaurant business.

The project will be successful if it is completed within the set deadlines, used in at least one restaurant.

**Links**

Link

+

×

Create

Slika 6.33 Stranica za kreiranje dokumenta

Po uspješnom kreiranju dokumenta korisnik će biti preusmjeren na stranicu za pregled dokumenta, na kojoj je dostupan popis svih entiteta uz koje je taj određeni dokument vezan. Na ovoj stranici također je dostupna opcija uređivanja dokumenta klikom na *Edit* i opcija brisanja dokumenta klikom na *Delete*. Stranica za uređivanje dokumenta slična je stranici za kreiranje dokumenta dok je postupak brisanja dokumenta, jednak kao i postupak brisanja ostalih entiteta, a brisanje dokumenta je uvijek dostupno. Slika 6.34 prikazuje stranicu za prikaz dokumenta naziva „Crash Dump“ koji ima jednu vanjsku poveznicu, te je povezan s jednim zadatkom, jednim sastankom i jednim sprintem.

## Actions Document - Crash Dump

Edit

Delete

Name

Crash Dump

Document type

Common

Project

Wild 8 Open 2

Content

Attached to the document is a crash dump which occured in production server during last week.

Links

Link

[https://drive.google.com/open?id=0B8FE1\\_Rr6atlOGIfTHlIYjFZdWM](https://drive.google.com/open?id=0B8FE1_Rr6atlOGIfTHlIYjFZdWM)

Linked tasks

Task

Fix crash dump

Linked meetings

Meeting

Analyze why crash occurred in production and how to prevent future similar cases

Linked sprints

Sprint

Iteration 01

Slika 6.34 Stranica za prikaz detalja dokumenta

## 6.4. Kritički osvrt na razvijeni sustav

Funkcionalnosti razvijenog sustava prikazane su na primjeru projekta razvoja web stranice za restoran, kao što su bile prikazane i funkcionalnosti postojećih rješenja analiziranih u poglavlju 3. Razvijeni sustav prikazao se prikladnim za upravljanje Scrum projektima, ali svakako postoje poboljšanja koja bi razvijeni sustav učinila boljim. Moguće poboljšanje razvijenog sustava bilo bi uvođenje više vrsta radnih zadataka ili njihovo proizvoljno definiranje, odnosno njihove hijerarhije, kao i uvođenje proizvoljnih statusa zadataka. Time bi se omogućio razvoj sustava koji ne slijede u potpunosti Scrum metodologiju već neku varijantu te metodologije.

## 7. Korištene tehnologije i alati

Tokom razvoja alata u sklopu ovog rada korišteno je razvojno okruženje Microsoft Visual Studio 2017 [13]. Navedeno razvojno okruženje omogućava razvoj svih vrsti aplikacija koje je moguće razvijati nad .NET platformom. Svi projekti osim projekta „Application.Web.MVC“ postavljeni su da koriste .NET Standard 2.0, inačicu .NET platforme, dok je projekt „Application.Web.MVC“ postavljen da koristi .NET Core 2.0. Ukoliko projekt koristi .NET Standard verziju platforme, moguće ga je koristiti u projektima koji koriste .NET Core ili .NET Framework verziju .NET platforme, odnosno .NET Standard projekti izgrađeni su nad zajedničkim presjekom te platforme [14]. „Application.Web.MVC“ projekt koristi .NET Core verziju platforme jer korištene tehnologije u tom projektu nisu dostupne u .NET Standard platformi. Za praćenje promjena nad izvornim kodom programa korišten je alat Git [15], dok je za pomoć pri razvoju korišten alat GitHub [16]. Na poslužiteljskoj strani korišten je isključivo programski jezik C# i .NET platforma, te biblioteka MongoDB.Driver verzije 2.5.0 koja se koristi za pristup MongoDB bazi podataka. Verzija MongoDB [12] baze podataka korištene tokom razvoja je 3.4.10. Na klijentskoj strani korištena je biblioteka Bootstrap verzije 3.3.7, biblioteke jQuery verzije 2.2.0, te biblioteka Chart.js. Navedene biblioteke koriste se za realizaciju izgleda i funkcionalnosti korisničkog sučelja web aplikacije.

## 8. Zaključak

Upravljanje razvojem informacijskih sustava važno je kako bi se održala kvaliteta razvoja i postojao pisani trag o tome što je razvijeno i što je potrebno razviti, kao i mogućnost evidencije promijenjenih zahtjeva ili nadogradnje postojećih funkcionalnosti. Razvijena programska podrška omogućuje pohranu podataka o razvoju informacijskog sustava Scrum metodologijom. Scrum metodologija je agilna metodologija razvoja u kojoj se informacijski sustav razvija u Scrum tim koji se sastoji od nekoliko osoba i njihovih uloga. Podržano je praćenje više Scrum projekata unutar iste organizacije. Praćenje razvoja sustava podržano je definiranjem radnih zadataka na projektu, njihovim raspoređivanjem prema prioritetu i procjena težine, raspoređivanje tih zadataka u sprinteve. Ugrađena je i podrška za organiziranje sastanaka za projekte koji se razvijaju. Uz navedene mogućnosti razvijenog sustava moguće je kreirati i izvještaje o povijesti razvoja sustava ili o pojedinim sprintevima, kao i pregledati koji od timova je radio u nekom vremenskom intervalu na projektu i koje su u tom timu bile osobe. Razvijeno programsko rješenje te dva već postojeća rješenja testirana su na projektu razvoja web aplikacije za restoran. Sva tri programska rješenja pokazala su se prikladnima za razvoj ovakvog sustava. Iako se razvijeno rješenje pokazalo prikladnim za praćenje razvoja informacijskih sustava razvijanih Scrum metodologijom, uvijek postoji mogućnost za poboljšanje, a jedna od spomenutih mogućnosti je definiranje proizvoljnih tipova radnih zadataka i njihove hijerarhije, kako bi se podržalo praćenje razvoja sustava koji ne slijede nužno Scrum metodologiju, već neku svoju, sličnu vlastitu varijantu.

---

Jan Kelemen

# Literatura

- [1] K. Schwaber i J. Sutherland, »The Scrum Guide,« Studeni 2017. [Mrežno]. Dostupno na: <http://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf>. [Pokušaj pristupa 26 Prosinac 2018].
- [2] K. Fertalj, *Nastavni materijali iz "Razvoj informacijskih sustava"*, 2017.
- [3] Nepoznat, »Systems development life cycle,« 24 Ožujak 2018. [Mrežno]. Dostupno na: [https://en.wikipedia.org/wiki/Systems\\_development\\_life\\_cycle](https://en.wikipedia.org/wiki/Systems_development_life_cycle). [Pokušaj pristupa 26 Ožujak 2018].
- [4] T. Hussung, »What Is the Software Development Life Cycle?,« 10 Ožujak 2016. [Mrežno]. Dostupno na: <https://online.husson.edu/software-development-cycle/>. [Pokušaj pristupa 26 Ožujak 2018].
- [5] S. Ambler, »The Agile System Development Life Cycle,« Ambyssoft, 2012. [Mrežno]. Dostupno na: <http://www.ambyssoft.com/essays/agileLifecycle.html>. [Pokušaj pristupa 26 Ožujak 2018].
- [6] ScrumDesk, »ScrumDesk,« ScrumDesk, [Mrežno]. Dostupno na: <https://www.scrumdesk.com/>. [Pokušaj pristupa 17 Ožujak 2018].
- [7] Atlassian Inc, »Jira,« Atlassian Inc, [Mrežno]. Dostupno na: <https://www.atlassian.com/software/jira>. [Pokušaj pristupa 1 Travanj 2018].
- [8] S. McConnell, *Code Complete*, Redmond: Microsoft Press, 2004.
- [9] I. Sommerville, *Software Engineering*, Pearson Education Limited, 2015.
- [10] R. Martin C, *Clean Architecture: A Craftsman's Guide to Software Structure and Design*, Prentice Hall, 2017.
- [11] M. Fowler, *Patterns of Enterprise Application Architecture*, Addison-Wesley Professional, 2002.



- [12] MongoDB Inc, »MongoDB,« MongoDB Inc, [Mrežno]. Dostupno na: <https://www.mongodb.com/>. [Pokušaj pristupa 4 Siječanj 2018].
- [13] Microsoft, »Visual Studio IDE, Code Editor, VSTS, & App Center,« Microsoft, [Mrežno]. Dostupno na: <https://www.visualstudio.com/>. [Pokušaj pristupa 11 Lipanj 2018].
- [14] I. Landwerth, » .NET Standard - Demystifying .NET Core and .NET Standard,« Rujan 2017. [Mrežno]. Dostupno na: <https://msdn.microsoft.com/en-us/magazine/mt842506.aspx>. [Pokušaj pristupa 11 Lipanj 2018].
- [15] Software Freedom Conservancy, »Git,« Software Freedom Conservancy, [Mrežno]. Dostupno na: <https://git-scm.com/>. [Pokušaj pristupa 11 Lipanj 2018].
- [16] GitHub Inc, »The world's leading software development platform · GitHub,« GitHub Inc, [Mrežno]. Dostupno na: <https://github.com/>. [Pokušaj pristupa 11 Lipanj 2018].

# Pomagalo za upravljanje Scrum projektima

## Sažetak

U radu je opisan proces razvoja programske podrške za upravljanje Scrum projektima. Dan je pregled Scrum metodologije i artefakata vezanih uz razvoj koji potječu od korištenja te metodologije. Napravljena je analiza postojećih rješenja za praćenje ovakvih projekata. Iz analize domene razvoja sustava Scrum metodologijom i analize postojećih rješenja kreiran je skup zahtjeva i model podataka kojeg bi takvo pomagalo trebalo podržati. Predstavljen je obrazac razvoja informacijskog sustava pomoću čiste arhitekture i njezine karakteristike. Funkcionalnosti razvijenog sustava prikazane su na primjeru projekta razvoja web-aplikacije za restoran. Napravljen je pregled korištene platforme za razvoj, te pomoćnih alata i biblioteka.

**Ključne riječi:** programska podrška, razvoj informacijskih sustava, Scrum, praćenje projekata, čista arhitektura

# Scrum Project Management Tool

## Abstract

This paper describes the development process for a tool which manages projects developed with a Scrum methodology. This paper also gives an overview of the Scrum methodology and artifacts related to development that come from using this methodology. Two existing solutions for monitoring such projects was made to create a set of requirements and develop a domain model that such tool would need to support. This paper gives an overview of development of information systems using clean architecture pattern. The functionality of the developed system is presented with the example of a project for development of a web application for a restaurant. At the end an overview of the development platform, auxiliary tools and libraries that were used during development is given.

**Key words:** software, development of information systems, Scrum, project tracking, clean architecture

## Dodatak: MongoDB baza podataka

Uz rad priložena je instanca MongoDB baze podataka korištena u demonstraciji funkcionalnosti razvijenog sustava u poglavlju 6. U priloženoj bazi podataka spremljeno je osam korisnika, tri tima i tri projekta. U nastavku je dan popis podataka s kojima se moguće prijaviti na sustav kao neki od prethodno definiranih korisnika.

Email	Lozinka
jan.kelemen@scrummy.org	scrummy
luka.krizan@scrummy.org	scrummy
filip.gulan@scrummy.org	scrummy
hrvoje.butina@scrummy.org	scrummy
glorija.markac@scrummy.org	scrummy
marko.matijascic@scrummy.org	scrummy
marina.brebric@scrummy.org	scrummy
borna.skukan@scrummy.org	scrummy

## Obnova baze podataka

Prije početka postupka obnove potrebno je otvoriti arhivu „mongodb.zip“ na proizvoljnom mjestu, te se pozicionirati u folder gdje se nalazi sadržaj te arhive. Potrebno je također pokrenuti instancu MongoDB servera upisivanjem sljedeće naredbe u naredbenoj liniji:

```
mongod.exe
```

Ovom naredbom bit će pokrenut MongoDB server na pretpostavljenoj adresi i portu. Kako bi se obnovila baza podataka, potrebno je iz direktorija u kojem se nalazi sadržaj datoteke mongodb.zip pokrenuti naredbu:

```
mongorestore.exe dump/
```

## Dodatak: Popis slika

Slika 2.1 Životni ciklus razvoja sustava [4] .....	2
Slika 2.2 Životni ciklus sustava razvijanog Scrum metodologijom .....	4
Slika 2.3 Jedna iteracija u Scrum metodologiji [5] .....	5
Slika 3.1 Prozor za stvaranje novog projekta u alatu ScrumDesk.....	7
Slika 3.2 Kartica Story Map u alatu ScrumDesk.....	8
Slika 3.3 Kartica Roadmap u alatu ScrumDesk .....	9
Slika 3.4 Kartica Backlog u alatu ScrumDesk .....	9
Slika 3.5 Prikaz ploče zadataka u alatu ScrumDesk.....	10
Slika 3.6 Izvještaj jednog sprinta kreiranog alatom ScrumDesk [6] .....	11
Slika 3.7 Kartica Backlog u alatu Jira .....	12
Slika 3.8 Prikaz ploče zadataka sprinta u alatu Jira.....	13
Slika 4.1 Konceptualni model podataka.....	16
Slika 4.2 Konceptualni model radnih zadataka .....	18
Slika 4.3 Konceptualni model dokumenata.....	18
Slika 5.1 Komponente sustava u čistoj arhitekturi [10] .....	20
Slika 5.2 Komponente razvijenog rješenja.....	21
Slika 5.3 Model entiteta timova i osoba .....	22
Slika 5.4 Model entiteta projekta i radne liste proizvoda.....	22
Slika 5.5 Model entiteta sprinta i liste zadataka sprinta .....	23
Slika 5.6 Model entiteta radnih zadataka, dokumenata i sastanaka.....	24
Slika 5.7 Dijagram klasa slučaja korištenja.....	25
Slika 5.8 Model slučaja korištenja za upravljanje radnom listom proizvoda.....	25
Slika 5.9 Sučelje tvornice za kreiranje slučajeva korištenja za upravljanje sprintevima ....	26
Slika 5.10 Sučelja repozitorija za pristup perzistenciji podataka .....	27

Slika 5.11 Model dokumenata osoba i timova .....	28
Slika 5.12 Model dokumenta projekta.....	28
Slika 5.13 Model dokumenata sprinta .....	29
Slika 5.14 Model dokumenata radnih zadataka, sastanaka i dokumenata.....	29
Slika 5.15 Model repozitorija za entitete sprinteva .....	31
Slika 5.16 Model upravljača.....	32
Slika 5.17 Model prezentera i modela pogleda za jedan slučaj korištenja .....	33
Slika 5.18 Dijagram slijeda izvršavanja jednog slučaja korištenja .....	34
Slika 6.1 Stranica za registraciju korisnika .....	35
Slika 6.2 Pokušaj registracije korisnika s već postojećom adresom elektroničke pošte .....	36
Slika 6.3 Stranica za prijavu na sustav .....	37
Slika 6.4 Stranica s aktualnim poslom nekog korisnika.....	38
Slika 6.5 Stranica za pregled profila korisnika.....	38
Slika 6.6 Stranica za uređivanje podataka korisnika .....	39
Slika 6.7 Stranica za promjenu lozinke korisnika .....	39
Slika 6.8 Stranica s izvještajem o povijesti timova korisnika .....	40
Slika 6.9 Stranica s popisom korisnika.....	40
Slika 6.10 Stranica za kreiranje tima .....	41
Slika 6.11 Stranica za pregled tima .....	42
Slika 6.12 Stranica sa izvještajem o promjeni članova tima .....	43
Slika 6.13 Stranica sa izvještajem o povijesti projekata tima.....	43
Slika 6.14 Stranica s popisom timova .....	44
Slika 6.15 Stranica za kreiranje projekta .....	45
Slika 6.16 Stranica za pregled detalja projekta.....	46
Slika 6.17 Stranica za pregled povijesti timova projekta .....	46
Slika 6.18 Stranica s popisom svih projekata.....	47

Slika 6.19 Stranica za pregled radne liste proizvoda.....	48
Slika 6.20 Stranica za upravljanje radnom listom proizvoda .....	49
Slika 6.21 Stranica za kreiranje epa.....	50
Slika 6.22 Stranica za prikaz korisničke priče.....	51
Slika 6.23 Stranica za pregled planiranih sprinteva .....	52
Slika 6.24 Stranica za kreiranje sprinta .....	53
Slika 6.25 Stranica za pregled sprinta .....	54
Slika 6.26 Stranica s izvještajem sprinta .....	55
Slika 6.27 Stranica za prikaz projekta s aktualnim sprintem.....	56
Slika 6.28 Stranica za završetak sprinta .....	57
Slika 6.29 Stranica s izvještajem projekta .....	58
Slika 6.30 Stranica za pregled budućih sastanaka za projekt .....	59
Slika 6.31 Stranica za kreiranje sastanaka.....	60
Slika 6.32 Stranica za pregled popisa dokumenata .....	61
Slika 6.33 Stranica za kreiranje dokumenta .....	62
Slika 6.34 Stranica za prikaz detalja dokumenta.....	63

## **Dodatak: Popis tablica**

Tablica 2.1 Primjer definicije završenog za sustav razvijan Scrum metodologijom.....	6
--	---



## **Dodatak: Popis isječka koda**

Isječak koda 5.1 Dokument entiteta projekta .....	30
---	----