

NORGES TEKNISK-NATURVITENSKAPELIGE
UNIVERSITET

INFORMATIKK PROSJEKTARBEID 1

IT1901

Rapport - Gruppe 9

Forfattere:

ASLAK HOLLUND

SONDRE WOLD

EBBA FINGARSEN

ØYVIND AUKNER

HÅKON S. KJÆRNLI

HANNA B. GIMSE

Studentnummer:

476716

476707

476762

257384

751261

757674

1. FEBRUAR 2018

ANTALL ORD: 5085



Sammendrag

Denne rapporten presenterer arbeidsprosess, produkt og læringsutbytte i faget IT1901, Informatikk Prosjektarbeid 1. Rammeverket Scrum ble benyttet for smidig produktutvikling, etter innføring fra foreleser samt selvstudie av prosjektmetodikken. Gjennom arbeidet ble det, i tråd med de oppførte læringsmålene i faget, fokusert på optimalisering av arbeidsprosess, gruppedynamikk og kompetansebygging i team. Produktet etter fullført prosjekt, var en fungerende nettapplikasjon, som oppfylte majoriteten av kundens ønskede funksjonalitet.

Innhold

1	Introduksjon	4
1.1	Faget	4
1.2	Prosjektet	4
1.3	Produktet	4
1.4	Gruppen	5
2	Arbeidsprosess	6
2.1	Scrum	6
2.2	Modifisering av scrum	6
2.3	Daily-scrum	7
2.4	Sprint-goals og brenndiagrammer	7
2.5	Kundeforholdet	8
2.6	Tid og sted	9
2.7	Organisatoriske verktøy	9
2.8	Rammeverk og teknologier	9
3	Utvikling og testing	10
3.1	Testplan	10
3.2	Testbasert utvikling	11
3.3	Akseptansetesting	11
3.4	Feilhåndtering	11
3.5	Demonstrasjon av produkt	12
4	Sprinter	12
4.1	Sprintlengde	12
4.2	Utgivelsesplan	13
4.3	Sprintplanlegging	13
4.4	Tidsestimering	14
5	Sprint-retrospektiv og sprint-review	14
5.1	Sprint 1	15
5.2	Sprint 2	16
5.3	Sprint 3	17
5.4	Sprint 4	17
5.5	Sprint 5	18
5.6	Retrospektiv-konklusjon	19

6 Konklusjon	20
Vedlegg	
A Sidekart	22
B Gruppekontrakt	23
C Referat fra daily-scrum	30
D Brenndiagrammer	31
E Product-backlog	36
F Utgivelsesplan	37
G Test-caser	38
H Referater fra sprint-retrospektiv	40
I Brukermanual	42
I.1 Inn- og utlogging	42
I.2 Opprettelse av grupper og brukere	42
I.3 Opprettelse av sjanger, artist, konsert og scene	43
I.4 Gi organizer/arrangør tilgang på adminpanelet	44
I.5 Opprette timeslots	45
I.6 Få en totaloversikt over alle konserter og hvilke scener de spilles på	45
I.7 Sende bookingtilbud til godkjenning av bookingsjef	46
I.8 Godkjenne bookingtilbud som bookingsjef før de går ut til artistmanager	46
I.9 Se oversikt over kontaktinfo, omtale og salgstall for artister som er booket	46
J Teknisk brukermanual	47

1 Introduksjon

1.1 Faget

Informatikk Prosjektarbeid 1 har som hovedformål å gi studentene praktisk erfaring med prosjektarbeid, samt det å forholde seg til kunde. Faget skal gi en dypere forståelse av hvordan samspillet mellom prosess og produkt kan by på utfordringer, samt gi studentene mulighet til å forbedre praktiske ferdigheter i gruppe-organisering og samarbeid. Læringsmålene i faget oppnås gjennom et gruppe-programmeringsprosjekt, der et team på fem til seks studenter skal bygge opp og sette sammen et større software-produkt, spesifisert av en kunde.

1.2 Prosjektet

Prosjektoppgaven i faget IT1901 inkluderte blant annet planlegging, tilrettelegging og utføring av arbeidsprosess, samt kommunikasjon og oppfølging av kunde. Dette ga studentene mulighet til å reflektere over de ikke-tekniske sidene ved et utviklingsprosjekt. Kunden ble oppfattet som reell, og all kommunikasjon med han/henne ble utført deretter. Dette ga studenten erfaring med å pleie kundeforhold, forholde seg til ikke-tekniskemedarbeidere, samt det å jobbe med en arbeidsgiver. En annen viktig komponent i prosjektarbeidet var gruppesamarbeid, og studentene fikk i løpet av prosjektet kjennskap til forskjellige aspekter ved problemløsning i team. Prosjektet ga også studentene en god innføring i prosjektmetodikken *Scrum*.

1.3 Produktet

Oppdraget i år ble gitt av en musikkfestivalarrangør. Han ønsket et internt system for festivalen, som skulle effektivisere organisering, booking og planlegging av konserter. Systemet skulle være slik at frivillige med ulike roller kunne logge inn og få tilgang til informasjon relevant for deres arbeidsområder. De aktuelle arbeidsrollene var i utgangspunktet arrangør, bookingsjef, bookingansvarlig, og tekniker. Festival-systemet skulle blant annet ha en oversikt over alle konserter som var planlagt, med tilhørende scene,

tidspunkt, tekniske behov og teknikere. I tillegg skulle det finnes en oversikt over alle bookingtilbud som har blitt sendt, samt relevant informasjon fra tidligere bookinger, både fra denne festivalen og andre tidligere festivaler. Nødvendig nøkkelinformasjon om artister og tidligere konserter skulle også være tilgjengelig i systemet, i tillegg til annen funksjonalitet. Se vedlegg (A) for et sidekart som viser den overordnede strukturen til produktet.

Prosjektet ble i Sprint 4 utvidet med nye brukerhistorier. Disse omhandlet funksjonalitet til nye roller i systemet som skulle ha ansvar for blant annet servering og promotering.

1.4 Gruppen

Gruppen besto av seks informatikkstudenter med forskjellig kompetanse og erfaring. Alle gruppens medlemmer hadde relativt ulik utdanningsbakgrunn, som bidro til å skape en produktiv, godt fungerende gruppedynamikk. Tidligere ervervet individuell kompetanse innenfor flere av fagets læringsmål viste seg å være særdeles viktig for gruppens progresjon. Den skjeve fordelingen av erfaring med utviklingsrammeverket Django ba likevel på utfordringer. Disse utfordringene ble tidlig adressert, og gruppen brukte mye tid i oppstartsfasen på opplæring og kompetansedeling. Dette viste seg i senere tid å være en produktiv beslutning, som førte til en høy gjennomføringsevne og et stort læringsutbytte.

Gruppen besluttet tidlig i oppstartsfasen at kompetansedeling skulle være et fokusområde under arbeidet. En av grunnpilarene i agil prosjektmetodikk er konseptet om *full-stack-utvikling*, der alle gruppens medlemmer skal kunne effektivt bidra i alle prosjektets deler. Gruppens fokus på kompetansedeling var sentral for å oppnå nettopp dette.

Før prosjektstart ble det skrevet en gruppekontrakt (B) som alle gruppens medlemmer signerte. Denne kontrakten la grunnlaget for de ikke tekniske aspektene av prosjektet

2 Arbeidsprosess

2.1 Scrum

Scrum er et smidig rammeverk for utførelse av komplekse prosjekter. Scrum ble opprinnelig formalisert for programvareutviklingsprosjekter, men det fungerer godt i de fleste komplekse og innovative arbeidsområder. En av nøkkelfaktorene til Scrums massive suksess i industrien er metodens strenge rammer, men interne fleksibilitet. Gruppen har valgt å implementere hele Scrum, inkludert alle eventer og artefakter. Dette har gitt gruppen klare og presise retningslinjer for hvordan utviklingen av produktet skal foregå, og har i praksis fungert svært godt. Gruppen har også hatt en felles *definition-of-done*, for å sikre at *product-backlog-items*, og *tasks* som ligger under disse, møter en bestemt standard og kvalitet før de kan klassifiseres som ferdige. Denne definisjonen har bestått av følgende kriterer [1]:

1. Koden må være leselig
2. Arbeidet må være dokumentert
3. Stavekontroll må gjennomføres
4. *Code-review* med et annet gruppemedlem må utføres
5. Kode må være *merget* til master

2.2 Modifisering av scrum

En av de første utfordringene for gruppen var forskjellene mellom arbeidskapasiteten og møtefrekvensen for en studentgruppe og en vanlig ”åtte til fire” arbeidssituasjon. Av dette fulgte et behov for en skreddersydd implementasjon av Scrum. Selv om Scrum i utgangspunktet har mange strenge regler og retningslinjer, som ikke skal modifiseres, så gruppen seg nødt til å gjøre flere mindre endringer. Et eksempel på dette er konseptet *daily-scrum*. Denne modifiseringen utdypes videre i seksjon 2.3.

Scrum rammeverket innebærer også utpekelsen av en *scrum-master*, som skal følge opp at gruppen arbeider i tråd de med verdier og retningslinjer som er bestemt av Scrum. For å maksimere gruppemedlemmenes læringsutbytte har

rollen som scrum-master gått på rundgang, slik at samtlige gruppemedlemmer har fått prøve seg i denne rollen [1].

2.3 Daily-scrum

Daily-scrum er ment som en daglig oppdatering på hva som foregår i *scrum-teamet*. Denne seansen er primært ment for utviklingsteamet, da dette er deres mulighet til å få et overblikk over progresjonen i en *sprint*, uten at ikke-teknisk personale fører samtalen. Scrum-master trenger i utgangspunktet ikke å være tilstede under daily-scrum, men må sørge for at utviklingsteamet har møtet og holder det innenfor tidsrammen på femten minutter. De dagene gruppen har hatt avtalte arbeidsøkter har daily-scrum blitt avholdt slik det spesifiseres av Schwaber & Sutherland (2017) i The Scrum Guide. Resten av uken har gruppen benyttet en kanal på kommunikasjonsverktøyet Slack for å foreta dette møtet, samt brukt en *slack-bot* for påminnelser. Et eksempel på en slik daily-scrum finnes i vedlegg (C) De tre spørsmålene som har blitt besvart under daily-scrum er som følger:

1. Hva har jeg gjort siden sist for å oppnå denne sprintens sprint-goal?
2. Hva skal jeg gjøre i dag for å oppnå denne sprintens sprint-goal?
3. Hva hindrer meg fra å oppnå denne sprintens sprint-goal?

2.4 Sprint-goals og brenndiagrammer

På starten av hver sprint-planlegging ble det bestemt et *sprint-goal* for perioden. Dette målet fungerte som en overordnet visjon for sprinten, og var utgangspunktet i opprettelsen av *product-backlog-items*. Under ligger en oversikt over alle målene i prosjektet:

- Sprint 1: Få på plass grunnleggende infrastruktur i prosjektet. Sette av tid til å drive internopplæring av Django i samspill med Postgres, vår database.
- Sprint 2: Sette opp teststruktur, etablere grunnleggende design, lage de første modellene i databasen samt finne en måte å håndtere brukerroller, adgangskontroll og redigeringsbegrensning.

- Sprint 3: Lage første rapportutkast og utvide det praktiske prosjektet med ønsket og mulig funksjonalitet etter oppsatte Trello-kort.
- Sprint 4: Fortsette arbeidet med både prosjekt og rapport samt ha spesielt fokus på å flytte redigeringsmuligheter for lavere brukerrang fremover i systemet.
- Sprint 5: Etterkomme enkelte av kundes ønsker i forbindelse med oppdaterte brukerhistorier og utgivelsesplan. Finne og luke ut funksjonsfeil og pirk i forkant av kundeovertakelse og kodefrys.
- Sprint 6: Ferdigstille rapporten, holde demo og avslutte prosjektet.

Et annet viktig verktøy som ble benyttet i forbindelse med sprintene, var bruken av brenndiagrammer. Disse diagrammene (D) viste progresjonen i en periode, samt den ideelle trenden. I Sprint 1 ble det laget et brenndiagram som viste den ideelle trenden for hele prosjektet basert på utgivelsesplanen. Dette viste seg å være et nyttig verktøy da gruppen enkelt kunne se om de lå på etterskudd i forhold til planlagt progresjon.

Det ble produsert brenndiagrammer for samtlige sprinter. Disse illustrerte hvor fort gruppen fullførte estimerte oppgaver, og var nyttige for at neste sprint kunne planlegges bedre.

2.5 Kundeforholdet

Tradisjonelt fungerer produkteier som et mellomledd mellom organisasjon og kunde. Produkteier tar del i mange av sprintens artefakter, og er et fullverdig medlem av teamet. I dette prosjektet var produkteier og kunde samme person, noe som skapte flere utfordringer. Vanligvis vil produkteier ha større personlig eierskap i produktet, samt ha ansvaret for å vedlikeholde prosjektets product-backlog (E). I gruppens tilfelle oppførte kunde/produkteier seg som en utenforstående, og representerte rent kommersielle ønsker.

2.6 Tid og sted

I en vanlig arbeidssituasjon jobber ofte de ansatte faste tider på et felles kontor. For gruppen ble denne regulariteten byttet ut med to faste møtetider i uken: tirsdager fra klokken 14:15 til 18:00, og torsdager fra klokken 08:15 til 11:00. I tillegg har gruppens medlemmer jobbet kontinuerlig på egenhånd de periodene dette har vært nødvendig.

Produkteier var også mindre tilgjengelig enn i en optimal arbeidssituasjon, og det ble kun arrangert kundemøte én gang i uken. I Scrum er det i utgangspunktet anbefalt at produkteier deltar på hvert møte (med unntak av daily-scrum), og spesielt ved sprintplanlegging. Gruppens løsning på dette var å kontakte produkteier over e-post de gangene det var nødvendig. På kundemøtene ble produkteier oppdatert på hvordan prosjektet lå an i forhold til den aktuelle sprinten og utgivelsesplanen (F). Kunde hadde også anledning til å komme med ønsker og innvendinger i løpet av møtet.

2.7 Organisatoriske verktøy

Oppgavebehandlingsapplikasjonen Trello ble brukt for å holde oversikt over oppgaver i de ulike sprintene, som erstatning for en fysisk tavle. Dette fungerte svært godt, og var et viktig verktøy for å holde alle oppdatert på hvor langt gruppemedlemmene var kommet med de ulike arbeidsoppgavene. Ettersom fysiske daglige møter ikke ble avholdt, ble det valgt å aktivt ta i bruk kommunikasjonsplattformen Slack, som en kompensasjon i organisering av alt fra utviklingsarbeid til møteplanlegging. Alt av fildeling utover prosjektets kode har blitt gjort på en felles Google Drive.

2.8 Rammeverk og teknologier

Programmeringsspråket Python, komplimentert av rammeverket Django er blitt brukt for utvikling av logikk og funksjonalitet i prosjektet. Front-end-delen av programmet består hovedsakelig av HTML, JavaScript og CSS-kode, med innslag fra Twitter Bootstrap-rammeverket. PostgreSQL ble benyttet for databasestruktur for det ferdige systemet.

Git og GitHub ble tatt i bruk som henholdsvis versjonskontroll og kodede-lingsplattform. Ved prosjektstart ble det gjort en grundig gjennomgang av skikk og bruk for et slikt verktøy, slik at gruppen startet arbeidet med like forventninger og relativt like forutsetninger. Som en del av gruppens felles definition-of-done, ble det satt som et krav at all kode skal godkjennes av et annet gruppemedlem ved merge til master, via GitHub's *pull-requests*. På denne måten har gruppen aktivt holdt hverandre til en høy standard, både når det kommer til generell kodekvalitet og til øvrige konvensjoner. I praksis har testing som planlagt skjedd i sluttet av hver sprint, mens brorparten av *code-review* har skjedd med hjelp av pull-request, før merge i det hele tatt har funnet sted. Dette resulterte i at code-review har foregått på kontinuerlig basis, mens koden fremdeles var friskt i minne, i stedet for i store skippertak på sluttet av hver sprint. Dette er i tråd med gruppens filosofi om at leselig kode og bred forståelse innad i gruppen har presidens over hurtig slipp av ny kode og funksjonalitet.

3 Utvikling og testing

3.1 Testplan

Gruppen bestemte seg tidlig for at en godt utarbeidet testplan ville være et verdifullt verktøy under prosjektarbeidet. Fokuset i utviklingen av testplanen var god kommunikasjon mellom medlemmene og kontinuerlig testing.

Planen baserte seg i hovedsak på et sett med test-caser (G), produsert med utgangspunkt i brukerhistoriene (E). Test-casene var utformet som steg-for-steg instruksjoner for å teste implementasjonen av brukerhistoriene. Sentral for testingen var også gruppens felles definition-of-done, som beskrevet i seksjon 2.1. Denne definisjonen eksisterte som en sjekklister i Trello, og ble kontinuerlig benyttet for å opprettholde kvalitetsnivået på prosjektet.

Regelmessig gjennomgang av test-caser, felles code-review av pull-requester og håndheving av gruppens definition-of-done var viktige verktøy for å holde medlemmene ansvarlige overfor deres egen testplan.

3.2 Testbasert utvikling

Basert på tidligere erfaring blant gruppemedlemmene og undersøkelser gjort før prosjektstart, besluttet gruppen å benytte seg av testdrevet utvikling. Testdrevet utvikling vil si at det først skrives automatiserte tester, og deretter skrives kode for å passere disse testene [2]. Dessverre var dette noe gruppen allikevel ikke hadde kompetanse og tid til å gjennomføre, og testingen av produktet ble derfor gjort på en annen måte.

Dette betydde i praksis at alle krav ble omformulert til konkrete *test-caser* (G), som produktet kontinuerlig ble testet og vurdert opp mot. Når alle testene for en brukerhistorie var passert og godkjent, ble nye test-caser opprettet for videre utvikling fant sted. Test-casene var kontinuerlig tilgjengelig i et dokument på gruppens felles Google Drive.

3.3 Akseptansetesting

I løpet av prosjektet ble det benyttet akseptansetesting for å undersøke hvorvidt produktet tilfredstilte brukerbehov og funksjonelle krav. Denne testprosessen ble utført både internt og ekstrement.

Interne akseptansetester skal i utgangspunktet utføres av personer som er interne for organisasjonen, men som ikke sitter på utviklingsteamet [3]. Grunnet fagets struktur har gruppen ikke hatt slike testpersoner tilgjengelig. Det ble derfor bestemt at alle utviklere regelmessig skulle teste funksjoner de ikke selv hadde vært med på å utvikle. Dette var i tillegg til code-review, som skjedde før enhver pull-request ble godkjent og merget til master.

Det ble utført ekstern akseptansetesting sammen med kunde både etter Sprint 2 og etter kodefrys. Under denne testingen fikk kunden håndtere programmet, prøve diverse test-caser og gi tilbakemelding på brukeropplevelsen.

3.4 Feilhåndtering

Gruppen ble etter første demonstrasjon gjort oppmerksom på viktigheten av å ha gode prosedyrer for feilhåndtering internt i systemet. Systemet hadde på dette tidspunktet flere åpenbare logiske feil som kunne anses som hindringer

for daglig drift. Gruppen hadde ikke laget gode nok valideringsmekanismer, noe som blant annet gjorde det mulig å selge flere billetter enn det var kapasitet til på en konsertarena. Dette ble håndtert ved å opprette validering med Django på de aktuelle modulene. Videre ble det bestemt at alle fremtidige brukertester skulle sjekke for tilfeller der det kunne forventes å finne feil.

3.5 Demonstrasjon av produkt

I løpet av prosjektets gang ble det gjennomført én presentasjon og to demonstrasjoner av produktet. Ved første presentasjon var det redegjørelse for gruppekontrakt (B) og utgivelsesplan (F) som stod på agendaen. De to påfølgende demonstrasjonene bestod av fremvisning av det henholdsvis foreløpige og ferdige produktsystemet, samt presentasjon av framdrift i henhold til utgivelsesplan. Det ble også ved hver demonstrasjon vist hvordan gruppens testplan har fungert i praksis, samt demonstrert utførelse av brukertester (G).

4 Sprinter

4.1 Sprintlengde

Scrum anbefaler at lengden på en sprint ikke overskredder 4 uker. Begrenset sprintlengde er med på å motvirke problemer som teknisk gjeld, endringer i ønsker fra kunde midt i en sprint, og uoversiktighet i prosjektet. Korte sprinter gir også økt pålitelighet og flyt i utvikling ved å lage hyppige, naturlige refleksjonsperioder [1]. For dette prosjektet ble sprintlengden satt til to uker. Den avgjørelsen ble tatt på bakgrunn av gruppens lave erfaringsnivå, sammenlignet med et profesjonelt scrum-team, og at møter med kunden bare var en gang i uken. Kort sprintlengde gjorde det også lettere for gruppen å sørge for at deres mål om å fokusere på egen læring ble ivaretatt, ettersom det ble mange punkter der det var naturlig å vise frem sitt arbeid og eventuell ny lærdom. Gjennom prosjektet viste det seg å være veldig positivt med korte sprintlengder, særlig i forbindelse med omprioriteringer fra kunde.

4.2 Utgivelsesplan

For å ha et overordnet bilde av progresjonen i prosjektet ble det tidlig i Sprint 1 utarbeidet en utgivelsesplan (F). Denne spesifiserte hvilke brukerhistorier gruppen forventet å fullføre i hver sprint. Utgivelsesplanen inneholdt nummererte brukerhistorier, samt et estimat for tidsbrukt og viktighet av disse. Estimaten ble utarbeidet i felleskap i gruppen, ved hjelp av estimeringsverktøyet *planning-poker*. Det ble opprinnelig bestemt at det skulle utvikles tre brukerhistorier per sprint, men etter Sprint 3 ble det klart at gruppen hadde opparbeidet seg nok teknisk kompetanse til å øke antallet til 4 brukerhistorier per sprint. Skalaene som ble benyttet for vurdering av viktighet og tid var relative til prosjektet.

Som utgivelsesplanen i vedlegg (F) illustrerer, ble det i prosjektets tidlige faser prioritert brukerhistorier som omhandlet funksjonalitet tilknyttet booking og tekniske behov. I Sprint 4 ble det fra kundens side uttrykt et ønske om å fokusere på nye brukerhistorier, deriblant brukerhistorier relatert til promotering. Gruppen valgte i samsvar med kunden å først fokusere på det som allerede stod oppført, og heller omrokkere prioriteringen for Sprint 5. Brenndiagrammet for Sprint 4 (D) viser at gruppen dro med seg seks timer med teknisk gjeld inn i Sprint 5, noe som videre forsinket arbeidet med de nye brukerhistoriene.

4.3 Sprintplanlegging

Hver sprint startet med sprintplanlegging. Ettersom tiden med kunde var svært begrenset, anså gruppen det som en god løsning å gjennomføre sprintplanleggingen uten kunde tilstede. Som erstatning for deltakelse på sprintplanleggingen ble kunde holdt oppdatert på planen og utviklingen via e-post og regelmessige møter.

Under sprintplanleggingen ble det bestemt hvilke, og hvor mange, brukerhistorier som skulle implementeres, og eventuelt hvilket annet arbeid som måtte gjøres i forbindelse med prosjektet. Brukerhistoriene ble valgt etter utgivelsesplanen, og deretter delt opp i mindre oppgaver som ble lagt inn i kort på Trello i en *sprint-backlog*. Gruppen tidsestimerte disse deloppgavene ved hjelp av *planning-poker*. Det arbeidet som ikke var direkte relatert til

produktet, men som hadde å gjøre med gruppearbeid og rapport, ble også delt opp i mindre oppgaver, tidsestimert og lagt inn sprint-backlog.

Når sprint-backloggen var klar, kunne gruppens medlemmer velge oppgaver. Fordelingen skjedde i hovedsak basert på gruppemedlemmenes ønsker, men det ble også lagt vekt på ferdighetsnivå og kapasitet. Målet var at alle i gruppen skulle ha oppgaver som både ga gruppemedlemmene anledning til å utfordre seg selv og å lære mer, samtidig som de skulle kunne gi individuell mestringsfølelse.

Overlapp i funksjonalitet mellom flere av brukerhistoriene ga rom for at like oppgaver kunne fordeles blant flere gruppemedlemmer. Dette førte til høyt læringsutbytte, og tilrettela for kompetansedeling innad i gruppen. Til tross for at dette valget førte til effektivitetstap i utviklingen totalt, var det i tråd med gruppens visjon om kompetanseheving over produksjon.

4.4 Tidsestimering

Tidsestimering har vært et nyttig verktøy i programvareutviklingen og sprint-planleggingen. I dette prosjektet har alle brukerhistorier, product-backlog-items og tasks blitt tidsestimert ved hjelp av konseptet planning-poker. Dette fungerte på følgende måte: Alle gruppens medlemmer uttrykte på likt hvor mange timer de mente en oppgave ville ta. Deretter fortalte de med høyest og lavest tidsestimat hvorfor de mente oppgaven ville ta nevnte tid. Som regel ble det som et resultat av denne diskusjonen klart hva som ville være et realistisk tidsestimat. Tidsestimatene ble deretter ført opp med sine respektive arbeidsoppgaver i Trello. Etter hver oppgave var ferdigstilt, i samsvar med gruppens felles definition-of-done, ble reelt tidsbruk for oppgaven også ført opp i Trello.

5 Sprint-retrospektiv og sprint-review

Hver sprint ble avsluttet med en *sprint-review* sammen med produkteier. Under disse møtene ble arbeidet som var gjort den aktuelle sprinten gjennomgått, og veien videre ble diskutert. Deretter ble det holdt et sprint-retrospektiv hvor gruppen selv vurderte den gjennomførte sprinten. En opp-

summering av referatene fra sprint-retrospektivene er presentert i vedlegg (H).

Selv om innholdet i hver retrospektiv varierte fra sprint til sprint var strukturen relativt stabil. Retrospektivene startet med sprintens siste daily-scrum, for så å gå over til en åpen diskusjon rundt sprinten og hvordan den hadde gått. Fokuset lå på å avdekke forbedringspotensiale, forutse fremtidige problemstillinger og å identifisere gode trender gruppen hadde opplevd. Det ble også rettet søkelys mot spesielt bemerkelsesverdig innsats av enkeltpersoner i gruppen. Etter diskusjonen gikk gruppen videre til generering av brenndiagram (D) for perioden, og utfylling av et *employee-satisfaction-sheet*.

Employee-satisfaction-sheet er et verktøy, utviklet av gruppen, som ble brukt både for å vurdere, men også tidlig oppdage endringer i gruppens arbeidsmiljø. Gruppen genererte en *average-employee-satisfaction* ved at hvert gruppemedlem individuelt satte en tallkarakter på hvor fornøyde de hadde vært med sprinten, på en skala fra 1 til 10. Gjennomsnittet av disse verdiene ble notert som average-employee-satisfaction i et eget dokument på Google Drive.

Utover retrospektivene i slutten av hver sprint forpliktet også gruppen seg til å møtes etter faget og prosjektet var avsluttet, for å gjennomføre en komplett *post-mortem*[4]. Denne post-mortemen skulle handle om gruppens innsats i hele faget, da inkludert arbeidet med rapporten. Dette ville foregå etter at alt fagarbeid var levert, og øvelsen ville altså kun bli for gruppens egen del. Målet med en slik post-mortem er å sikre egen læring og å gjøre gruppemedlemmene best mulig foreberedt på senere prosjekter i lik stil.

Nedenfor følger en oppsummering av hver sprint og deres tilhørende retrospektiver.

5.1 Sprint 1

I Sprint 1 tok gruppen i felleskap avgjørelsen om å ikke jobbe med produktet som skulle produseres, men heller fokusere på å legge et godt grunnlag for det fremtidige arbeidet. Medlemmene besluttet å generere en rekke brukerhistorier rundt gruppens prosjektrelaterte behov. Det var tre områder som var spesielt sentrale for denne sprinten; samkjøring av forventninger til prosjektet gjennom en god gruppekontrakt, internkursing i gruppen for å sikre

at alle medlemmene følte de kunne bidra til prosjektet, og grunnleggende oppsett av prosjektet og utviklingsmiljø hos alle medlemmene.

Under denne sprintens retrospektiv ble det tydelig at samtlige i gruppen var svært begeistret for avgjørelsen om å ha stort fokus på opplæring. Det ble tidlig klart at enkelte medlemmer av gruppen hadde mer forhåndskunnskap i teknologiene som skulle benyttes enn andre. Gruppen tok derfor nødvendige grep for at alle skulle kunne oppnå mestringsfølelse og læring relatert til prosjektet. Videre var hele gruppen veldig positivt overasket over hvor godt alle gikk overens sosialt. Gruppens gode arbeidsmoral og pliktoppfyllelse kunne i stor grad antas å være grunnet det gode samholdet blant medlemmene. Spesiell ros gikk til medlemmene i gruppen som hadde lagt ned mye arbeid for å lage kursplan for resten av teamet, og det ble uttrykt stor interesse for å fortsette med slike kurs videre i prosessen.

I Sprint 1 ble det veldig viktig å rette søkelyset mot de tingene som fungerte bra, for å legge grunnlaget for gode trender videre i prosjektet.

5.2 Sprint 2

Etter Sprint 2 var gruppen spesielt fornøyd med måten alle gruppemedlemmene tok ansvar for egen læring, og var villige til å lære bort egen kunnskap. Gruppen følte generelt at forståelsen av Git hadde økt, men de gruppemedlemmene som var litt mer uerfarene med dette verktøyet følte det fortsatt var rom for forbedring. Videre viste det seg at alle følte de hadde fått bidratt på en positiv måte, og gruppen som helhet følte at arbeidet med prosjektet hadde gått raskere fremover enn forventet. Under denne sprinten sluttet gruppen og booke rom til gruppearbeidet på tirsdager og det ble uttrykt ønske om at dette var noe som gruppen skulle begynne med igjen. Under veiledningstimen hadde gruppen få spørsmål til veileder. Gruppen mente dette ikke nødvendigvis var fordi det ikke var noe å lure på, men heller fordi gruppen ikke var godt nok forberedt. Gruppen ønsket til neste veiledningstime å stille mer forberedt, å ha planlagt spørsmål og problemstillinger som skulle tas opp.

5.3 Sprint 3

Før Sprint 3 ble brukerhistoriene omprioritert av kunde. Brukerhistorie 14 ble prioritert høyere, slik at den ble inkludert i Sprint 3. Dette ble i utgangspunktet ikke sett på som et problem i gruppen, og punktet ble tatt med i vurderingen på det påfølgende sprintplanleggingsmøte.

Den aktuelle brukerhistorien var ikke spesielt omfattende i funksjonalitet, men den var avhengig av flere brukerhistorier som ennå ikke var påbegynt eller planlagt. Gruppen tok avgjørelsen om først implementere de underliggende funksjonene, til tross for at dette påførte dem noe teknisk gjeld. På denne måten kunne gruppen fortsette å tenke langsiktig og helhetlig, i stedet for å lage en midlertidig løsning som senere ville måtte endres på. Hele gruppen var enige om at det var en god vurdering, selv om det førte til at enkelte brukerhistorier som egentlig lå i sprint-backlog ble betydelig berørt. I tillegg ble noen av de bakenforliggende tekniske aspektene flyttet over i Sprint 4.

I sprint-retrospektivet var et gjennomgående tema at kommunikasjonen innad i gruppen hadde vært svært god hele sprinten. Dette hadde ført til at endringen i utgivelsesplanen gikk uten større problemer. Fordelingen av arbeidsoppgaver og individuelt arbeid var også vellykket, og til tross for større arbeidsmengde ble alle oppgavene fullført på en tilfredstillende måte. Det ble uttrykt misnøye med hvor mye fokus det hadde vært på front-end, og at det flere ganger hadde vært behov for kodefrys, uten at det hadde blitt overholdt.

5.4 Sprint 4

I Sprint 4 var det for første gang i løpet av prosjektet to gruppemedlemmer som ikke var til stede under sprint-review og sprintretrospektiv. I tillegg stilte enkelte, om ikke samtlige, gruppemedlemmer delvis redusert og lite forberedt denne sprint-reviewen. Dette ga tydelig utslag på kvaliteten for både sprint-retrospektivet og sprintplanleggingen av Sprint 5. På tross av at gruppen ikke var fulltallige denne dagen, ble det nevnt at alle gruppemedlemmene hadde vært til stede på samtlige møter så langt. Selv om prosjektet ble gjennomført samme semester som blant annet studentfestivalen UKA, hadde umeldt fravær ikke vært et problem tidligere.

Gruppemedlemmene var også fornøyde med å ha kommet godt i gang med

rapporten, og følte dette var en del av prosjektet som gruppen hadde god kontroll på. Som det siste positive punktet i dette retrospektivet ble det tatt opp at gruppen denne sprinten hadde vært bedre forberedt til møtene med kunde og veileder.

Under denne sprinten ble det lagt til flere nye brukerhistorier i product-backloggen. Gruppen syntes dette var utfordrende, da flere av disse brukerhistoriene ble oppfattet som irrelevante for produktets spesifikasjoner. Det var vanskelig for gruppen å se hvordan funksjonaliteten etterspurt i disse brukerhistoriene skulle passe sammen med resten av produktet.

Samtidig glemte gruppen å informere kunde om at mange av de nye brukerhistoriene som ble lagt til i oppgaveteksten i løpet av Sprint 4, ikke ville la seg ferdigstille til kodefrys. Her tok gruppen selvkritikk, da dette selvfølgelig burde blitt informert om ved første anledning. Etter kundemøtet midt i sprinten var kunden inneforstått med at ikke alle brukerhistoriene ville bli implementert.

5.5 Sprint 5

Etter Sprint 5 var gruppen svært fornøyd med at alle gruppemedlemmene nå hadde opparbeidet seg stor nok kompetanse i Django og resten av teknologistakken til å kunne jobbe helt selvstendig. Dette førte til at alle gruppemedlemmene i større grad enn i tidligere sprinter valgte oppgavene som var listet som uferdige på Trello og implementerte de på egenhånd. Gruppen mente dette var et resultat av at gruppemedlemmene nå var trygge på egen kompetanse i verktøyene som ble brukt i prosjektet. Avgjørelsen om å prioritere gruppemedlemmenes egen læring, og målsetningen om at alle skulle lære alt som ble gjort i prosjektet ble dermed sett på som oppnådd.

Under retrospektiven ble uttrykt misnøye med kvaliteten på planlegningen i forhold til i tidligere sprinter. Særlig på Trello var dette godt merkbart. Kortene var dårligere organisert enn de hadde vært tidligere, og ikke alle arbeidsoppgavene i forbindelse med et kort var skrevet ned. Oversikten over hva som skulle gjøres var dermed ikke like god som den hadde vært i tidligere sprinter. Sett i ettertid hadde det vært gunstig for sprinten om gruppen hadde planlagt mer omfattende på starten av første møte etter sprintplanleggingen, ettersom deler av gruppen ikke hadde vært tilstede på selve

sprint-planleggingen. På denne måten kunne problemene og forvirringen som oppstod vært forhindret.

5.6 Retrospektiv-konklusjon

Sprint-review og sprint-retrospektivet på slutten av hver sprint viste seg å være svært givende for gruppens videre arbeid med prosjektet. Alt som hadde fungert bra i den foregående sprinten ble bemerket, og på denne måten kunne gruppen gå inn i neste sprint bevisst på hvilke deler av arbeidprosessen som burde stå i fokus. På dette punktet ble også de av gruppens medlemmer som hadde lagt inn ekstra arbeid eller vist forbedring både i utvikling og arbeidsprosess applaudert. Denne anerkjennelsen, samt kontinuerlige tilbakemeldinger på godt arbeid, fungerte som motivasjon for gruppemedlemmene. Gruppen gjennom prosjektet fokusert på at innsats settes pris på, og at det skulle gis ros der det var fortjent.

Gruppen vurderte i hver av sprint-retrospektivene medlemmenes employee-satisfaction. Gjennomsnittet av disse verdiene er presentert i tabell 1 nedenfor.

Sprint	Average-employee-satisfaction
1	8.7
2	8.7
3	9
4	9.1
5	9.3

Tabell 1: Oversikt over average-employee-satisfaction i gruppa etter gjennomført sprint. Skala fra 1 til 10.

Gruppen tok også for seg mulig forbedringspotensiale i sprint-reviewene. Det var konstruktive tilbakemeldinger på hvilke områder av gruppens arbeidsprosess som kunne fungert bedre. Tilbakemeldingene kom ofte med konkrete forslag til hvordan prosessen kunne justeres for å fungere bedre i påfølgende sprinter. Dette var svært nyttig for videre arbeid, og førte til kontinuerlig forbedring i gruppedynamikk og samhandling, samt økt effektivitet i utviklingsarbeidet.

Et oppsummert referat fra alle sprint-retrospektivene kan finnes som vedlegg (H).

6 Konklusjon

Det er enighet blant samtlige i gruppen om at dette gruppeprosjektet var vellykket. Alle involverte jobbet jevnt, samt ga beskjed når fravær var uunngåelig. Til tross for studentfestivalen UKA, sykdom og annet engasjement var oppmøte særdeles bra.

Gruppens beslutning om å fokusere på kompetansedeling og læringsutbytte viste seg å være et pedagogisk og produktivt valg, som førte til at alle gruppens medlemmer tilegnet seg nok teknisk kompetanse til å jobbe på egenhånd med prosjektet. Dette hadde som konsekvens at gruppen hadde rask fremdrift i utviklingsarbeidet og stor mestringsfølelse.

Arbeidsprosessen til gruppen dreide seg i stor grad om bruken av rammeverket Scrum. Gruppen hadde to faste møtetidspunkt i uken, samt begrenset mulighet til å møte produkteier, gjennom hele prosjektet. På bakgrunn av disse begrensningene så gruppen det som nødvendig å gjøre enkelte endringer på Scrum. Blant annet fikk alle gruppemedlemmene i løpet av prosjektet muligheten til å ha rollen scrum-master. Gruppen mente dette var et godt tiltak for å øke medlemmenes forståelse av rollen, og rammeverket generelt.

Gruppen møtte en utfordring knyttet til dialog med produkteier. Under Sprint 4 ble det lagt til flere brukerhistorier, men gruppen informerte ikke produkteier umiddelbart om at flere av disse brukerhistoriene ikke kom til å bli ferdigstilt. Dette burde opplagt vært gjort ved første mulighet da dette kunne resultert i omprioriteringer fra kunde.

En annen utfordring var at gruppen mente at mange av brukerhistoriene som ble lagt til under Sprint 4 var irrelevante og gikk utenfor det opprinnelige omfanget av produktet. Dette gjorde det vanskelig for gruppen å opprettholde et helhetlig inntrykk av produktets spesifikasjoner. Derfor så gruppen det som nødvendig å diskutere om spesifikasjonene til produktet hadde endret seg, eller om mye av funksjonaliteten etterspurt i de nye brukerhistoriene var mindre viktig.

Avslutningsvis mente gruppen at sluttproduktet var velfungerende, og hadde de viktigste funksjonalitetene implementert. Dette sa produkteier seg enig i under den siste akseptansetesten.

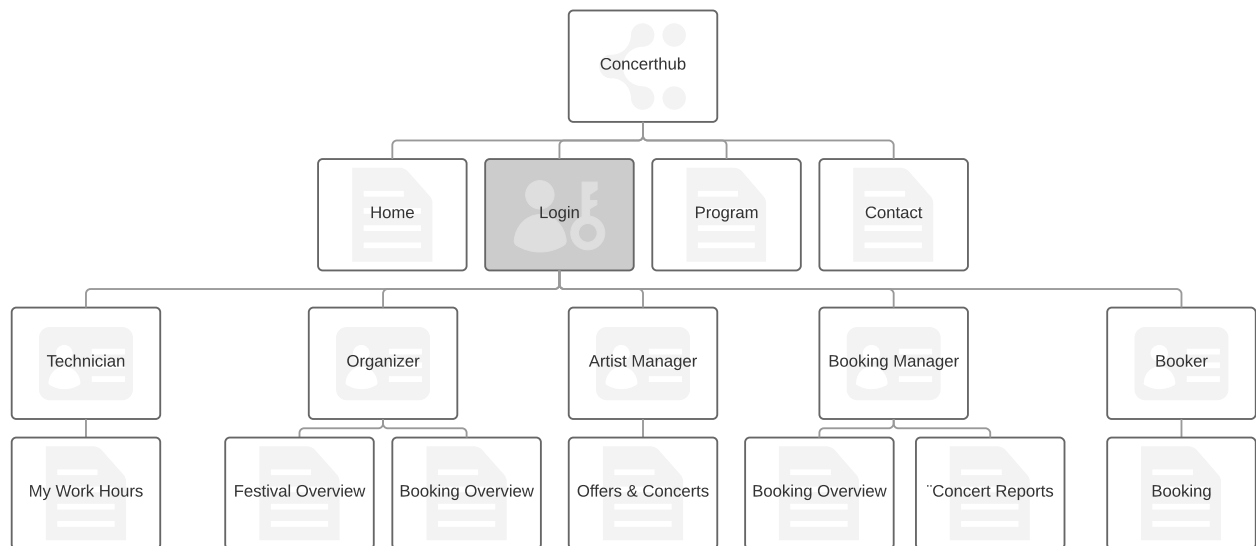
Referanser

- [1] Schwaber, K & Sutherland, J. (2017) *The Scrum Guide*. United States: scrumguides.org
- [2] Kniberg, H. (2015). *Scrum and XP from the Trenches* (2. utgave). United States: C4Media.
- [3] Acceptance Testing Fundamentals. (2017). Hentet fra: <http://softwaretestingfundamentals.com/acceptance-testing/>
- [4] Dingsøy, T. (2004). Postmortem reviews: purpose and approaches in software engineering. *Information and Software Technology*, 47, 293–303

A Sidekart

CONCERTHUB SITE MAP

Sondre Wold | November 13, 2017



B Gruppekontrakt

Gruppekontrakt IT1901

Gruppe 9

September 2017

Gruppemedlem	Studentnummer	Epost-adresse
Aslak Hollund	476716	aslakho@stud.ntnu.no
Sondre Wold	476707	sondrewold1@gmail.com
Øyvind Aukner	257384	oyvind.auk@gmail.com
Ebba Fingarsen	476762	ebbafin@gmail.com
Håkon Kjærnli	751261	hakon.s.kjarnli@gmail.com
Hanna Bergljot Gimse	757674	hanna.bergljot@gmail.com

Studentassistent og kunde: Håkon Solbjørg, hlsolbjorg@gmail.com

Innhold

1	Prosedyrer	2
1.1	Fast møtetidspunkt	2
1.2	Kommunikasjonskanaler	2
1.3	Avgjørelser i gruppen	2
1.4	Agendaer og møter	2
1.5	Arbeid og effektivitet	2
2	Forventninger	3
2.1	Ønsket standard	3
2.2	Strategi for å oppnå ønsket standard	3
2.3	Foretrukken form for ledelse	3
3	Individuelle forventninger	3
3.1	Individuell deltakelse	3
3.2	Individuelt ansvar	4
4	Konsekvenser for brudd av gruppekontrakt	4
A	Disiplinærsystem	6

1 Prosedyrer

1.1 Fast møtetidspunkt

Gruppen skal ha minst én arbeidsøkt på 4 timer i uken som grunnlag. Utover dette tilstreber gruppen også minst én tilleggsøkt i uken, der innholdet bestemmes etter behov. Denne økten kan brukes enten som en ekstra arbeidsøkt, et statusmøte, en sosial sammenkomst osv. Gruppen ønsker også å ta i bruk daglige scrum-checkin's over Slack, for å sikre kontinuerlig kommunikasjon mellom gruppemedlemmene.

1.2 Kommunikasjonskanaler

Gruppen har blitt enig om å hovedsakelig bruke følgende kommunikasjonskanaler, i tillegg til muntlig kommunikasjon i arbeidsøkter og møter:

- Slack-team
- Google Drive
- Trello

1.3 Avgjørelser i gruppen

Alle større avgjørelser skal diskuteres i plenum i gruppen. Ved uløselig konflikt bestemmer flertallet. Dersom gruppen ikke klarer å komme til enighet på denne måten, konsulteres kunden/studentassistenten, og hans/hennes ønsker prioriteres.

1.4 Agendaer og møter

Scrum-master setter agenda for møter, og påser at agendaen følges. Hele gruppen har sammen ansvar for å minimere digresjoner og forstyrrelser.

1.5 Arbeid og effektivitet

Gruppen har kommet til enighet om et disiplinærsystem for å påse at frister blir overholdt, og at arbeid blir utført som det skal. Se Appendix A. Utover disiplinærsystemet brukes daglige scrum-checkin's for å måle fremgang, og tilrettelegge for enkel kommunikasjon.

2 Forventninger

2.1 Ønsket standard

Gruppen streber etter best mulig produkt, men prioriterer tildels prosedyre og egen læring over kvaliteten på sluttproduktet. Samtlige medlemmer i gruppen er svært motivert for å jobbe mot en bra karakter på oppgaven.

2.2 Strategi for å oppnå ønsket standard

Gruppens grunnleggende filosofi baserer seg på å prioritere lesbarhet og gruppens totale læringsutbytte foran hurtig utviklet ny funksjonalitet. Gruppen ønsker også å bruke testdreven utvikling, god versjonskontroll og å unngå "feature creep".

2.3 Foretrukken form for ledelse

Gruppen bruker Scrum som prosjektplanleggingsverktøy, og ønsker derfor å ha en flat struktur. Gruppen vil også bytte på hvem som er "scrum-master" fra sprint til sprint. Dette gjøres for å oppnå økt læringsutbytte, og jevnere arbeidsfordeling. Dette er i tråd med vår filosofi om å prioritere gruppens individuelle kompetanseheving.

3 Individuelle forventninger

3.1 Individuell deltakelse

Det forventes at hvert gruppemedlem svarer på daglig Scrum, kommer på oppsatte møter og gjør sine oppgaver etter beste evne. Det er også forventet at medlemmene under de oppsatte møtene dedikerer sin fulle oppmerksomhet til gruppearbeidet. Det forventes at alle i gruppen vurderer egen kapasitet og evne, og ikke tar på seg mer arbeid enn han/hun klarer å utføre. Det er hvert enkelt gruppemedlem sitt ansvar å melde i fra om han/hun har behov for hjelp, sitter fast eller føler seg overveldet. Ettersom gruppen ønsker et høyest mulig læringutbytte totalt er det også forventet at hver enkelt i gruppen bidrar med sin kunnskap, slik at vi lærer av hverandre. Alle har også et ansvar for å inkludere samtlige gruppemedlemmer, samt å holde seg oppdatert på prosjektets framgang.

3.2 Individuelt ansvar

Medlemmene i gruppen er forventet å overholde frister og utføre tildelte oppgaver, så langt det lar seg gjøre. Dersom et medlem ikke klarer/har mulighet til å utføre en oppgave innen tidsfristen er det medlemmets eget ansvar å si i fra og be om hjelp fra resten av gruppa. Det forventes også at alle medlemmer i gruppen er tilgjengelige gjennom de avtalte kommunikasjonkanaler. Dette inkluderer å jevnlig oppdatere Trello, samt å sjekke Slack minst én gang om dagen (kun ukedager).

4 Konsekvenser for brudd av gruppekontrakt

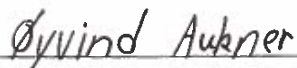
Retningslinjer for konsekvenser for mindre brudd på gruppekontrakt er oppført i Appendix A. Større forseelser eller situasjoner som oppstår vil diskuteres og vurderes individuelt av gruppen. Dersom neglisjering av forpliktelser eller liknende vedvarer vil studentassistent konsulteres.

1. Jeg deltok i formuleringen av samtlig innhold i denne kontrakten.
2. Jeg forstår at jeg er forpliktet til å følge disse vilkår og betingelser under hele prosjektets varighet.
3. Jeg forstår at dersom jeg ikke følger de vilkår og betingelser som er oppgitt, vil dette få konsekvenser som beskrevet.



Aslak Hollund

Trondheim, 12. september 2017



Øyvind Aukner

Trondheim, 12. september 2017



Ebba Fingarsen

Trondheim, 12. september 2017



Håkon Kjæslø

Trondheim, 12. september 2017



Sondre Wold

Trondheim, 12. september 2017



Hanna Bergljot Gimse

²⁸ Trondheim, 12. september 2017

A Disiplinærsystem

Gruppen har valgt å innføre det kjente disiplinærsystemet ofte kalt **ølstraff**. En ølstraff tilsvarer én enhet, eller 33 kroner. Tre ølstraffer kan kombineres til én vinstraff, tilsvarende 100 kroner. Enheten skal kjøpes og medbringes til gruppens førstkommende eller avtalte team-building. Brudd på dette punktet resulterer i en ny ølstraff.

Følgende betingelser gir grunnlag for å dele ut ølstraff til gruppemedlemmer:

1. Å komme for sent til et oppsatt møtetidspunkt (for sent defineres som at klokken har tikket over på neste siffer, altså slik at kl. 14.16 er for sent dersom møtetiden var satt til kl. 14.15), og man ikke melder ifra før senest 15 minutter før.
2. Å ikke fullføre en trello-oppgave (Unntaket her er dersom medlemmet har bedt om hjelp fra andre i gruppen, og fortsatt venter på å få det).
3. Dersom over 50% av gruppens resterende medlemmer mener det er riktig, kan ølstraff utdeles for andre overtredelser.


C Referat fra daily-scrum

#dailyscrum


☆ | 👤 6 | 📌 1 | ✎ Add a topic

📞 ⓘ ⚙️


Wednesday, September 27th

**slackbot** 12:00 PM


Reminder: Daily scrum time.

**Sondre Wold** 12:02 PM ☆


I går jobbet jeg med frontenden, og ble sånn ish ganske ferdig med det. Er en del iffy saker der som sikkert kunne blitt gjort bedre, men iom at dette ikke er en prioritering så vurderer jeg å sette opp en pull request allerede nå. I dag har jeg fikset litt bugs, ingenting hindrer meg utover manglende css-kunnskaper.

**Aslak Hollund** 🇳🇴 12:02 PM


Siden sist har jeg merget alle branches til master og generelt ryddet i gitprosjektet vårt. Jeg har også mekket modell for artist sammen med Håkon. Til neste gang tenkte jeg å se over det som ble gjort etter jeg dro i går, og kanskje leke mer med git. Hindres ikke av noe spesielt.

**Håkon Kjærnli** 12:12 PM


I går fiksa jeg artistmodellen. I dag har jeg ikke planlagt å gjøre noe og det er ikke noe som hindrer meg fra det

**Ebba Fingarsen** 12:19 PM

I går fiksa jeg og Hanna concert modellen og oppdaterte en del variabler. Jeg starta også på bookingoffer fordi jeg følte for det :p i dag hadde det vært fint om noen fikk til å fikse/se på master greine og om alt funker ellers skal jeg ingenting og ikkeno hindrer meg fra det

**Hanna Gimse** 1:06 PM

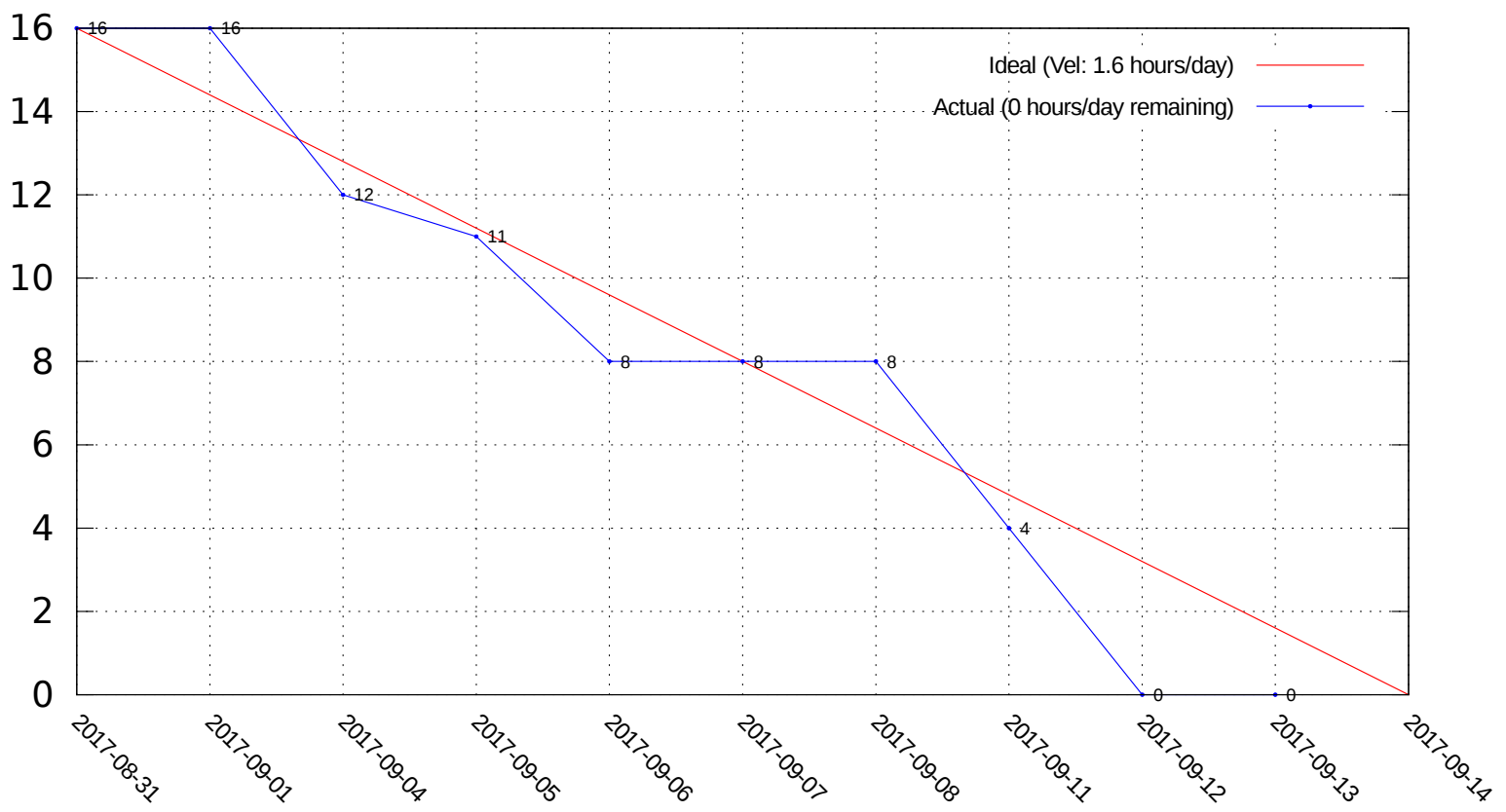
I går fikset jeg på concert-modellen sammen Ebba og hadde generelt en sjukt bra dag og masse mestringsfølelse(😁😁), i dag skal jeg ikke jobbe noe med prosjektet, og det går bra:)

**Øyvind Aukner** 1:20 PM

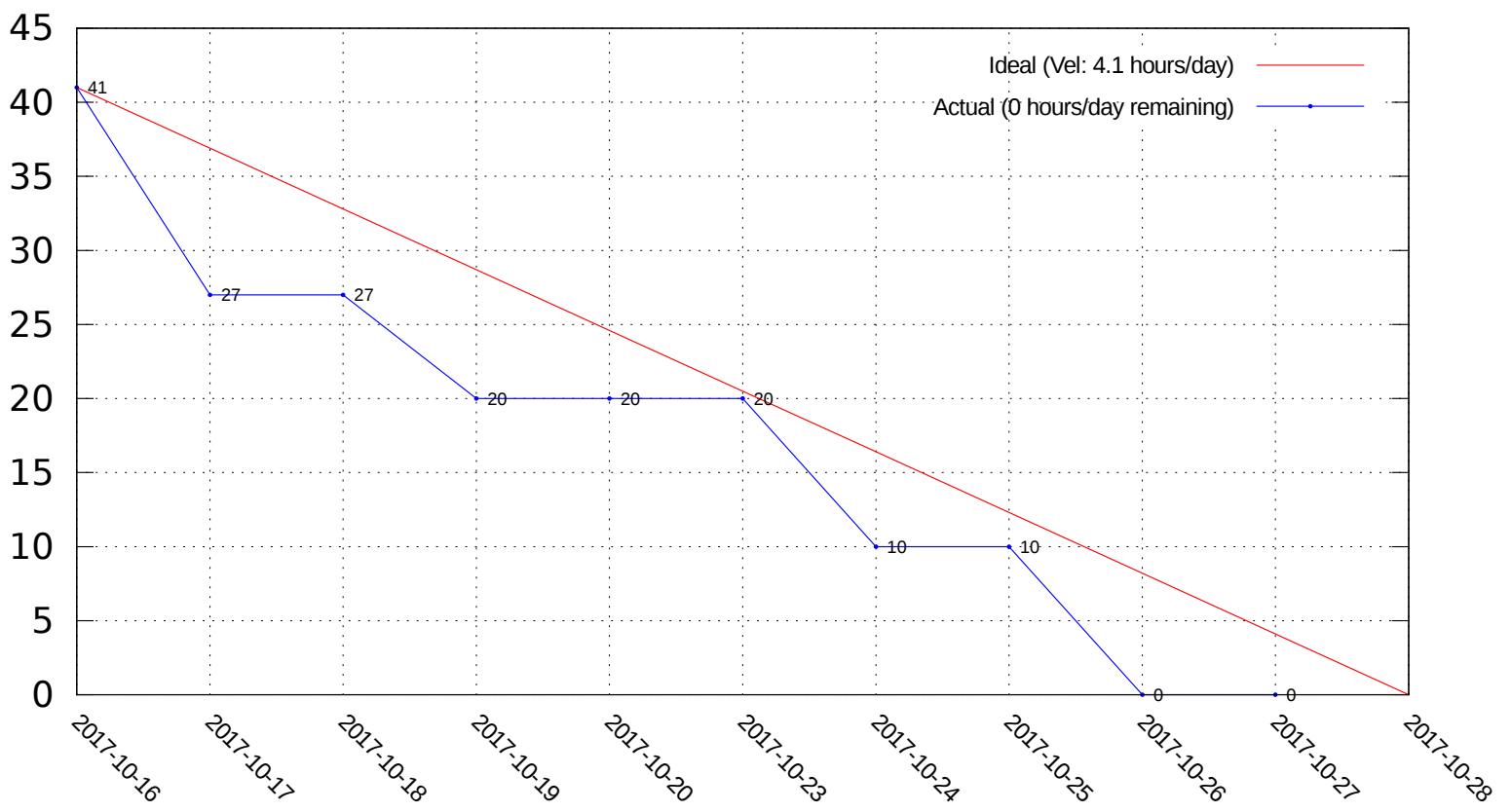
I går fordelte jeg litt tasks og hjalp jeg til der jeg trengtes i gruppetimen i tillegg til at jeg kom i mål på django-filtreringa som lar oss vise spesifikke konserter basert på hvilken teknikker som er innlogget. Whoop! Laga også ny PR med hanna_ebba som base. Håper den kan bli review'a i dag og migrert. Det var også en del problemer med databasen og jeg endte opp på å nuke'e min. Dette ser ut til å ha spredd seg og vi kan komme til å måtte gjøre det alle mann. I dag bør vi sette oss sammen en gang og merge til master og få ting klart til review/demo(?) i morgen. Hindres av at jeg er mye opptatt i dag og evt. ekstra tid det tar den enkelte å opprette sin egen database på nytt

D Brenndiagrammer

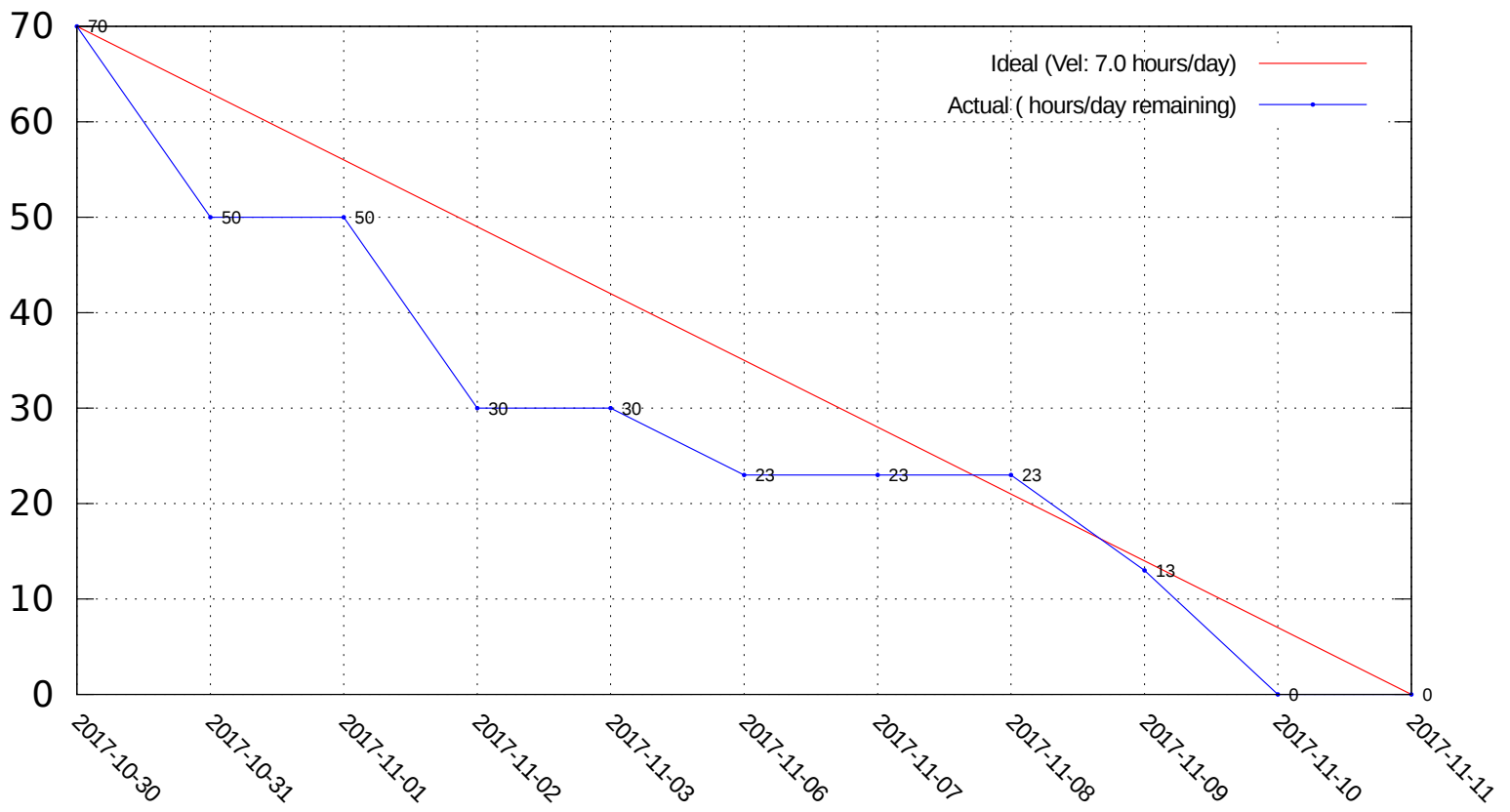
Sprint 1



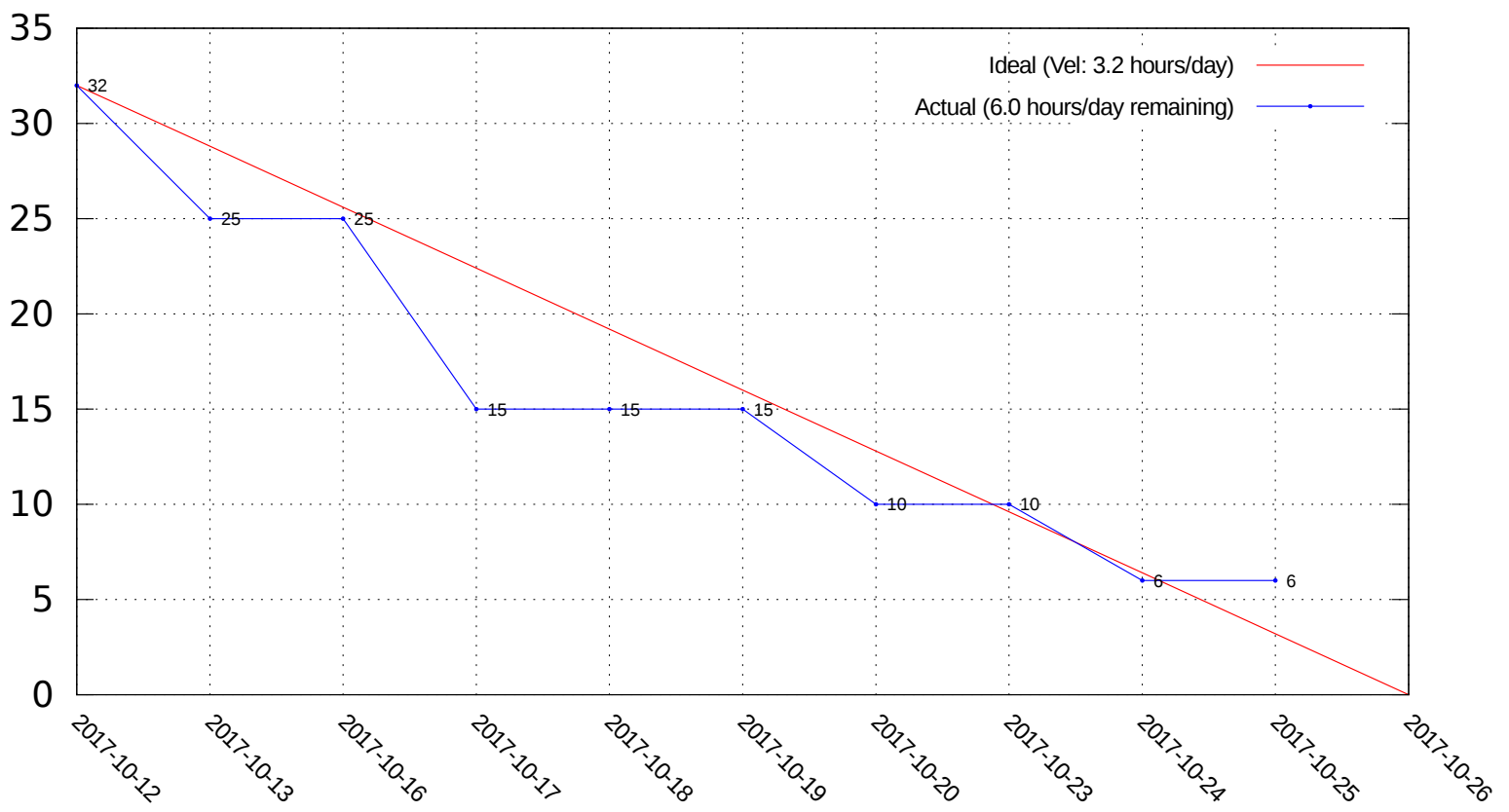
Sprint 2



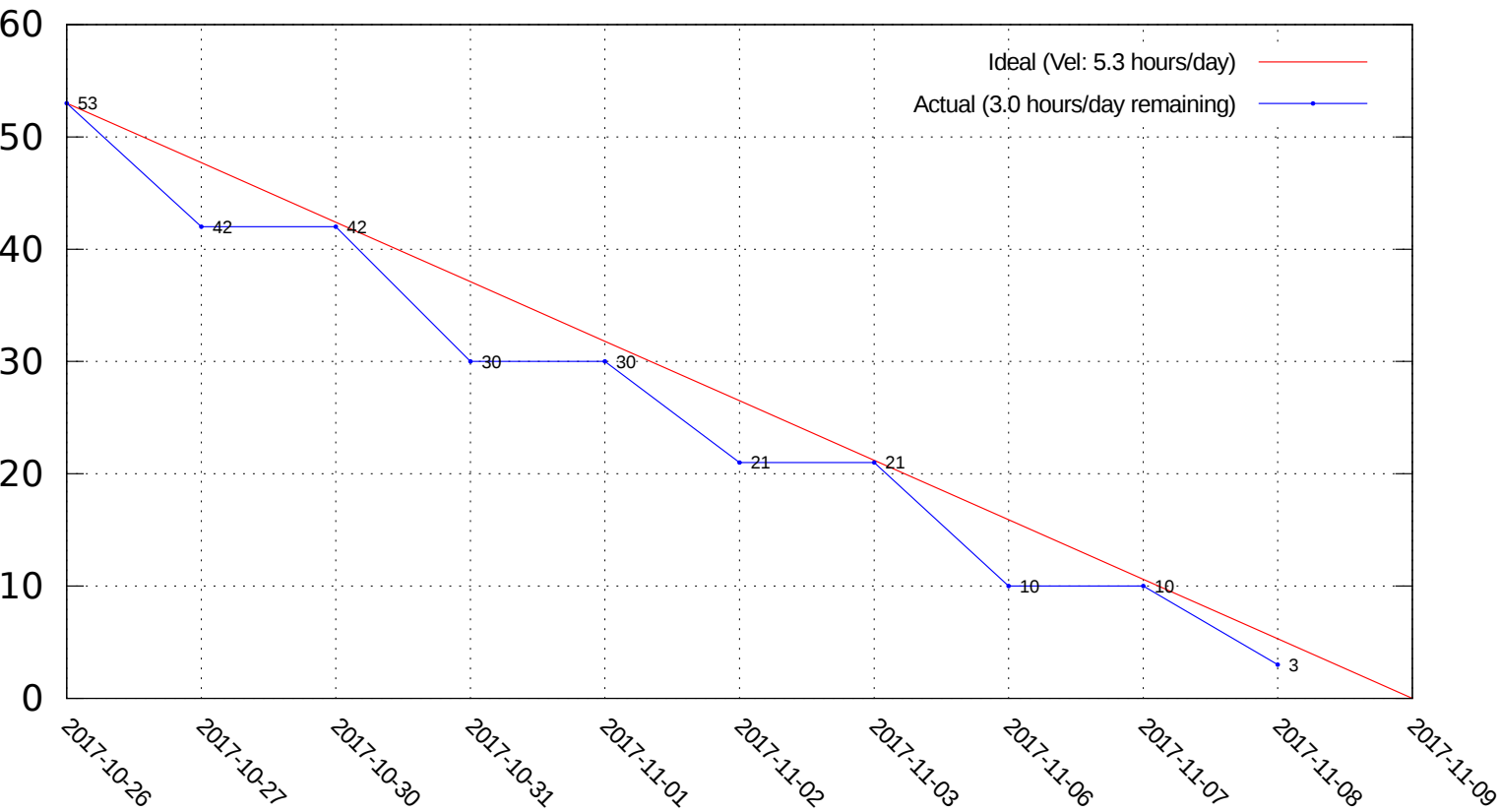
Sprint 3



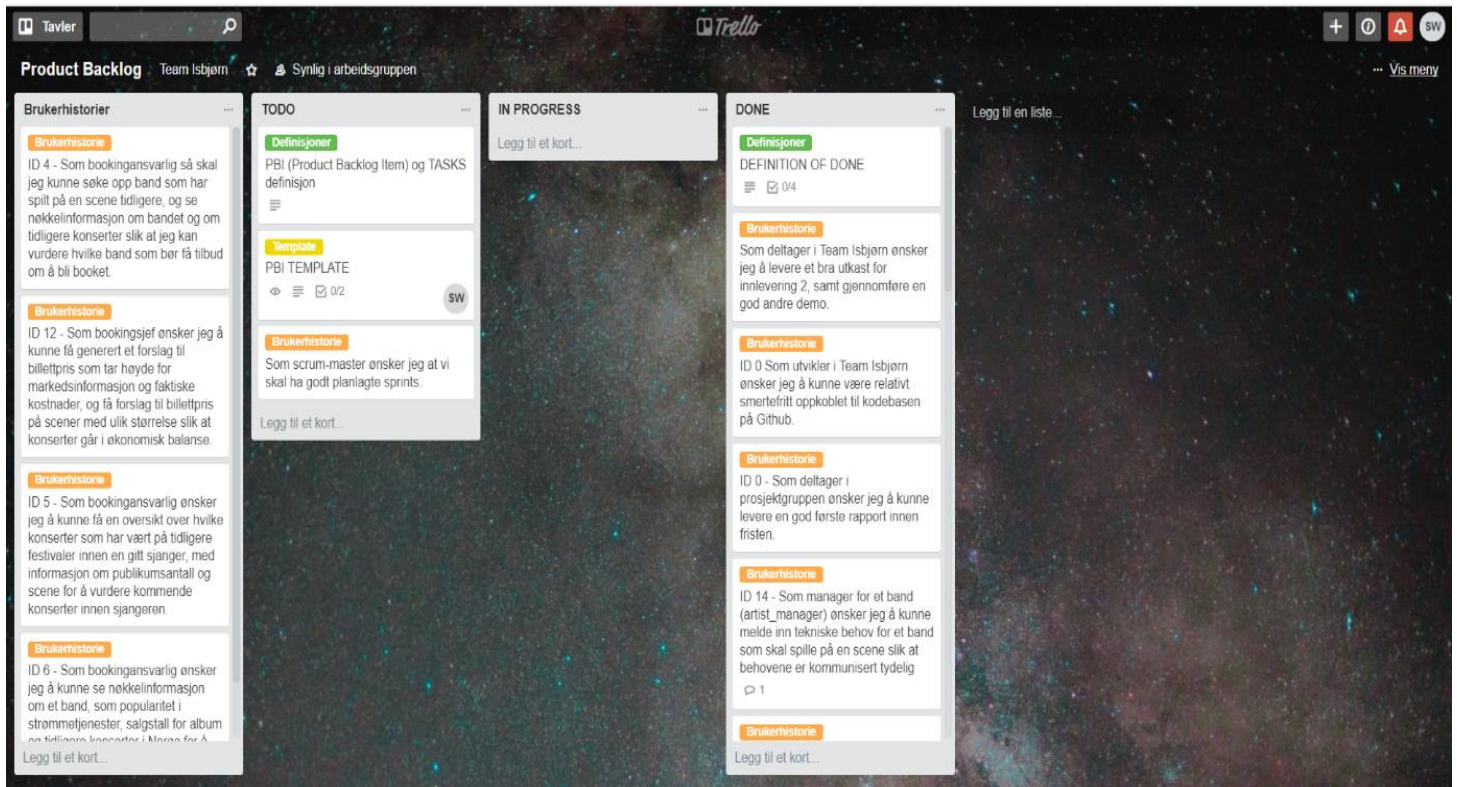
Sprint 4



Sprint 5



E Product-backlog



F Utgivelsesplan

	ID	Beskrivelse	Viktighet	Estimat
Sprint 1		Som deltager i prosjektgruppen ønsker jeg å kunne levere en god første rapport innen fristen.	101	10
		Som scrum-master ønsker jeg at vi skal ha godt planlagte sprints	100	4
		Som utvikler i Team Isbjørn ønsker jeg å kunne være relativt smertefritt oppkoblet til kodebasen på prosjektrepoet.	70	2
Sprint 2	2	Som arrangør av konserter ønsker jeg å få opp en totaloversikt over alle konserter på alle scener under en festival.	100	10
	1	Som arrangør av en konsert skal jeg kunne se en liste med hvem som hjelper til med rigging og tekniske oppgaver under konserten.	90	8
	13	Som lyd eller lystekniker skal jeg kunne få opp en oversikt over konserter jeg skal jobbe med.	80	8
Sprint 3	8	Som bookingsjef skal jeg kunne få en oversikt over antall konsertdatoer booket, antall tilbud sendt og antall ledige datoer.	70	12
	7	Som bookingsansvarlig ønsker jeg å kunne vise en liste med tekniske behov for et band for å forberede de som har teknisk ansvar.	60	7
	14	Som manager for et band ønsker jeg å kunne melde inn tekniske behov for et band som skal spille på en scene slik at behovene er kommunisert tydelig.	90	6
Sprint 4	10	Som bookingsjef skal jeg kunne se en oversikt over bookingtilbud som er sendt til band.	55	8
	3	Som bookingsansvarlig skal jeg kunne sende tilbud til manager for et band på booking på en spesifikk dato og med en spesifikk pris. Tilbudet skal være godkjent av bookingsjef.	50	6
	11	Som bookingsjef så skal jeg kunne se en rapport om konserter som viser publikumstall, kostnader og økonomisk resultat for alle konserter på en scene slik at jeg kan vurdere må	40	8
Sprint 5	9	Som bookingsjef skal jeg kunne godkjenne eller avslå tilbud til band som er forberedt av bookingsansvarlige.	40	5
	26	Som tekniker så vil jeg kunne melde fra om oppmøte på rigging til konserter.	40	5
	21	Som bookingsansvarlig skal jeg kunne legge inn en omtale av hvert band som er booket.	35	4
	24	Som PR-ansvarlig for en festival ønsker jeg å kunne se hvilke band som er booket med kontaktinformasjon, salgstall og lenker til presseomtale.	30	6
	25	Som PR-ansvarlig så ønsker jeg å få generert en oversikt over alle konserter på en festival med en kort omtale av hvert band.	30	5
Uferdig	23	Som manager for et band skal jeg kunne se hvem som er kontaktpersoner for rigging til konserten, for å avklare tekniske detaljer.	30	6
	17	Som arrangør av en festival så ønsker jeg å kunne se status på billettsalg og status på tekniske oppgaver på alle scener.		
	4	Som bookingsansvarlig så skal jeg kunne søke opp band som har spilt på en scene tidligere, og se nøkkelinformasjon om bandet og om tidligere konserter slik at jeg kan vurdere		
	12	Som bookingsjef ønsker jeg å kunne få generert et forslag til billettpris som tar høyde for markedsinformasjon og faktiske kostnader, og få forslag til billettpris på scener med ulik		
	5	Som bookingsansvarlig ønsker jeg å kunne få en oversikt over hvilke konserter som har vært på tidligere festivaler innen en gitt sjanger, med informasjon om publikumsantall og s		
	6	Som bookingsansvarlig ønsker jeg å kunne se nøkkelinformasjon om et band, som popularitet i strømmetjenester, salgstall for album og tidligere konserter i Norge for å vurdere o		
	16	Som ansvarlig for servering ønsker jeg å kunne beregne innkjøp av varer til barer på en scene for en gitt konsert.		
	15	Som ansvarlig for servering på konserter ønsker jeg å kunne se forventet publikumstall, start og slutt på en konsert, samt musikk sjanger.		
	22	Som manager for et band skal jeg kunne registrere hvilke låter som spilles under en konsert, og hvilke som øves på.		
	19	Som bandmedlem skal jeg kunne se en kalender hvor datoer for konserter og mulige konserter er merket slik at jeg har oversikt.		
	20	Som bandmedlem skal jeg kunne se hvilke scener som har gitt tilbud til manager om booking for å ha oversikt over mulige jobber fremover.		
	18	Som bandmedlem skal jeg kunne få opp en oversikt over øvingstider for bandet.		

G Test-caser

Title	Description	Test step	Expected result	User histories
Login - Arrangør	Verify that user registered as Arrangør can login successfully. Precondition: User must already be registered as Arrangør.	1. Navigate to login-page. 2. Click on 'Username'-field, and enter valid username. 3. Click on 'Password'-field, and enter valid password. 4. Click on the 'Sign in'-button.	Arrangør-home page will open, with 'Logged in' module showing in top right corner.	Som arrangør av konserter ønsker jeg å få opp en totaloversikt over alle konserter på alle scener under en festival.
Login - Tekniker	Verify that user registered as Tekniker can login successfully. Precondition: User must already be registered as Tekniker.	1. Navigate to login-page. 2. Click on 'Username'-field, and enter valid username. 3. Click on 'Password'-field, and enter valid password. 4. Click on the 'Sign in'-button.	Tekniker-home page will open, with 'Logged in' module showing in top right corner.	Som lyd eller lystekniker skal jeg kunne få opp en oversikt over konserter jeg skal jobbe med.
Login - Bookingsjef	Verify that user registered as Bookingsjef can login successfully. Precondition: User must already be registered as Bookingsjef.	1. Navigate to login-page. 2. Click on 'Username'-field, and enter valid username. 3. Click on 'Password'-field, and enter valid password. 4. Click on the 'Sign in'-button.	Bookingsjef-home page will open, with 'Logged in' module showing in top right corner.	Som bookingsjef skal jeg kunne få en oversikt over antall konserterdatoer booket, antall tilbud sendt og antall ledige datoer.
Login - PR-Ansvarlig	Verify that user registered as PR-Ansvarlig can login successfully. Precondition: User must already be registered as PR-Ansvarlig.	1. Navigate to login-page. 2. Click on 'Username'-field, and enter valid username. 3. Click on 'Password'-field, and enter valid password. 4. Click on the 'Sign in'-button.	PR-Ansvarlig-home page will open, with 'Logged in' module showing in top right corner.	Som PR-ansvarlig for en festival ønsker jeg å kunne se hvilke band som er booket med kontaktinformasjon, salgstall og lenker til presseomtale.
Login - Bookingansvarlig	Verify that user registered as Bookingansvarlig can login successfully. Precondition: User must already be registered as Bookingansvarlig	1. Navigate to login-page. 2. Click on 'Username'-field, and enter valid username. 3. Click on 'Password'-field, and enter valid password. 4. Click on the 'Sign in'-button.	Bookingansvarlig-home page will open, with 'Logged in' module showing in top right corner.	Som bookingansvarlig ønsker jeg å kunne vise en liste med tekniske behov for et band for å forberede de som har teknisk ansvar.
Login - Unsuccessful	Verify that user not registered cannot log in. Precondition: User is not registered.	1. Navigate to login-page. 2. Click on 'Username'-field, and enter invalid username. 3. Click on 'Password'-field, and enter invalid password. 4. Click on the 'Sign in'-button.	Login-page will reload, with error message asking to try again displayed.	
Arrangør - View concert	Verify that Arrangør is able to view all concerts and corresponding venues for a festival. Precondition: Successfully logged in as Arrangør.	1. From Arrangør-home page, click on the 'Program'-button.	New page with list of all concerts and corresponding venues and dates will open in same tab.	Som arrangør av konserter ønsker jeg å få opp en totaloversikt over alle konserter på alle scener under en festival.
Arrangør - View tekniker	Verify that Arrangør is able to view all technicians who are scheduled to work on a specific concert. Precondition: Successfully logged in as Arrangør.	1. From Arrangør-home page, click on the 'Program'-button.	New page with list of all concerts and a corresponding list of all technicians scheduled to work on the existing concerts will open in the same tab.	Som arrangør av en konsert skal jeg kunne se en liste med hvem som hjelper til med rigging og tekniske oppgaver under konserten.
Tekniker - View concerts	Verify that Tekniker is able to view all concerts he or she is scheduled to work on. Precondition: Successfully logged in as Tekniker.	1. From Tekniker-home page, click on the 'My Work Hours'-button.	New page with list of all concerts and corresponding venues he or she is scheduled to work on will open in same tab.	Som lyd eller lystekniker skal jeg kunne få opp en oversikt over konserter jeg skal jobbe med.
Bookingsjef - Concerts booked, offers sent and available dates	Verify that Bookingsjef is able to view the number of concerts booked, offers sent and available dates remaining. Precondition: Successfully logged in as Bookingsjef.	1. From the Bookingsjef-home page, click on the 'Booking overview'-button	New page with an overview over the booking offers he or she has sent and the concerts that have been booked. Further down on the same page is an overview over the available timeslots.	Som bookingsjef skal jeg kunne få en oversikt over antall konserterdatoer booket, antall tilbud sendt og antall ledige datoer.
Bookingansvarlig - View tech needs	Verify that Bookingansvarlig is able to view a list of all technical needs for a specific concert. Precondition: Successfully logged in as Bookingansvarlig.	1. From the Bookingansvarlig-home page, click on the 'Booking'-button.	New page with list of all concerts and corresponding venues and dates will first open in same tab. When clicking on one of the listed concerts, a new page will open, with a list of all the technical requirements for this specific concert.	Som bookingansvarlig ønsker jeg å kunne vise en liste med tekniske behov for et band for å forberede de som har teknisk ansvar.
Admin - Add user	Verify that Admin is able to successfully create a user and add it to a specific group. Precondition: Successfully logged in as Admin.	1. From the Admin-home page, click on the "Legg til"-option next to the "Brukere"-button. 2. Enter a username into the "Brukernavn"-field, and a password, fulfilling all the displayed requirements, in both the "Passord"-field and the "Passordbekreftelse"-field 3. Scroll down, and click on "Lagre". 4. On the next page, scroll down, and click on one of the available group options in the box to the left, and then click on the arrow to the right, between the two boxes. 5. Scroll down, and click on "Lagre".		
Admin - Add artist	Verify that Admin is able to successfully create an artist, and add all necessary information. Precondition: Successfully logged in as Admin.	1. From the Admin-home page, click on the "Legg til"-option next to the "Artist"-button. 2. Enter an artist name into the "Name"-field, a description into the "Description"-field, and a genre into the "Genre"-field. 3. Scroll down, and click on "Lagre". 4. Verify that the artist you just created appears in the list on the new page that loads.		
Admin - Add Booking offer	Verify that Admin is able to successfully add a booking offer. Precondition: Successfully logged in as Admin.	1. From the Admin-home page, click on the "Legg til"-option next to the "Artist"-button. 2. Enter an artist name into the "Name"-field, a description into the "Description"-field, and a genre into the "Genre"-field. 3. Scroll down, and click on "Lagre". 4. Verify that the artist you just created appears in the list on the new page that loads.		

Manager - Register tech needs	Verify that Manager is able to register and view technical needs for his or her bands. Precondition: Successfully logged in as Manager.	1. From the Manager-home page, click on the 'Offers & Concerts'-button. 2. To change/register the technical needs click the technical need and a new page with a technical needs field will show. 3. Fill this field with text and press the 'update' button to update the technical need and return to the 'Offers & Concerts'-page	New page with an overview over the bookingoffers including the technical needs and the ability to change these needs.	Som manager for et band ønsker jeg å kunne melde inn tekniske behov for et band som skal spille på en scene slik at behovene er kommunisert tydelig.
Bookingsansvarlig - Bookingoffers	Verify that Bookingsansvarlig is able to send a booking offer including specifications like date and price. Precondition: Successfully logged in as Bookingsansvarlig.	1. From the Bookingsansvarlig-home page, click on the 'Booking'-button 2. To send a new booking offer press the 'New'-button on the right hand side of the page. 3. Fill in the necessary fields, click the 'create'-button. This will return you to the 'Booking'-page.	New page with the ability to create new booking offers.	Som bookingsansvarlig skal jeg kunne sende tilbud til manager for et band på booking på en spesifikk dato og med en spesifikk pris. Tilbudet skal være godkjent av bookingsjef.
Bookingsjef - Concert reports	Verify that Bookingsjef is able to see an overview over the audience-numbers, costs and economical results for every concert. Precondition: Successfully logged in as Bookingsjef.	1. From the Bookingsjef-home page, click on the 'Concert Reports'-button	New page with an overview over the concerts and their economical details.	Som bookingsjef så skal jeg kunne se en rapport konserter som viser publikumstall, kostnader og økonomisk resultat for alle konserter på en scene slik at jeg kan vurdere marked for band og sjangre.
Bookingsjef - Accept offers	Verify that Bookingsjef is able to accept booking offers prepared by Bookingsansvarlig. Precondition: Successfully logged in as Bookingsjef.	1. From the Bookingsjef-home page, click on the 'Booking Overview'-button. 2. To approve or decline a booking offer click the 'Approve'-button or the 'Decline'-button on the specific booking offer.	New page with an overview over booking offers, including buttons to approve or decline a booking offer.	Som bookingsjef skal jeg kunne godkjenne eller avslå tilbud til band som er forberedt av bookingsansvarlige.
PR-Ansvarlig - Booking info	Verify that PR-Ansvarlig is able to view an overview over booked bands, their sales numbers, contact information and pressreview. Precondition: Successfully logged in as PR-Ansvarlig.	1. From the PR-Ansvarlig-home page, click on 'Festival Info'-button	New page with an overview over the contact information, sales numbers and press review of each booked band.	Som PR-ansvarlig for en festival ønsker jeg å kunne se hvilke band som er booket med kontaktinformasjon, salgstall og lenker til presseomtale.
PR-Ansvarlig - Concert reviews	Verify that PR-Ansvarlig is able to view an overview over concerts and the corresponding reviews. Precondition: Successfully logged in as PR-Ansvarlig.	1. From the PR-Ansvarlig-home page, click on the 'Program'-button. 2. Scroll down to view the second table for an overview over concerts and their corresponding reviews.	New page with an overview over the concerts and their corresponding reviews.	Som PR-ansvarlig så ønsker jeg å få generert en oversikt over alle konserter på en festival med en kort omtale av hvert band.
Manager - Technical contact person	Verify that Manager is able to view the technical contact person for each concert. Precondition: Successfully logged in as Manager.	1. From the Manager-home page, click on the 'Offers & Concerts'-button. 2. Scroll down to view the second table for an overview over concerts and their technical contact person.	New page with an overview over the concerts and their corresponding technical contact person.	Som manager for et band skal jeg kunne se hvem som er kontaktpersoner for rigging til konserten, for å avklare tekniske detaljer.
Tekniker - Meetup	Verify that Tekniker is able to report that he showed up and did his job at his concerts. Precondition: Successfully logged in as Tekniker.	1. From the Tekniker-home page, click on the 'My Work Hours'-button. 2. To edit the techs met press the 'Edit'-button, highlight the techs met and press the 'Update'-button to update and return to the 'My Work Hours'-page.	New page with an overview over view over future concerts including techs met.	Som tekniker så vil jeg kunne melde fra om oppmøte på rigging til konserter.
Bookingsansvarlig - Reviews	Verify that Bookingsansvarlig is able to add and edit a review on every band they have booked. Precondition: Successfully logged in as Bookingsansvarlig.	1. From the Bookingsansvarlig-home page, click on the 'Booking'-button. 2. Scroll down to view the second table for an overview over booked bands and the corresponding review. 3. To edit the review press the 'Edit this Review'-button. 4. Fill in the new review and press the 'Update'-button. This will update the review.	New page with an overview over the booked bands, their reviews and the ability to edit this review.	Som bookingsansvarlig skal jeg kunne legge inn en omtale av hvert band som er booket.

H Referater fra sprint-retrospektiv

Hva fungerte godt?	Hva kunne blitt gjort bedre?
<ul style="list-style-type: none">- Bra med intern kursing- Bra at alle bidrar- Bra at noen i gruppen kunne Django fra før- Bra oppmøte	<ul style="list-style-type: none">- Litt treg oppstart med selve prosjektet- Kunne lest oss litt mer opp på oppgaven

Tabell 2: Sprint-retrospektiv 1

Hva fungerte godt?	Hva kunne blitt gjort bedre?
<ul style="list-style-type: none">- Høy velocity- Bra og mye kommunikasjon- God dokumentasjon- Lav terskel for å spørre om hjelp	<ul style="list-style-type: none">- Kan bli bedre på git og pull requests- Vi bør booke rom til møtene våre- Forberede oss bedre til veiledertimene

Tabell 3: Sprint-retrospektiv 2

Hva fungerte godt?	Hva kunne blitt gjort bedre?
<ul style="list-style-type: none">- Fortsatt høy velocity- God kommunikasjon- Bra oppmøte- God arbeidsfordeling	<ul style="list-style-type: none">- Kan fortsatt bli bedre på git og pull requests- Lite fokus på rapport- Mer definitive kodestopp- Litt for mye fokus på front end

Tabell 4: Sprint-retrospektiv 3

Hva fungerte godt?	Hva kunne blitt gjort bedre?
<ul style="list-style-type: none"> - Høy velocity - Bra oppmøte, til tross for UKA - Skrevet masse på rapporten - Bedre forberedt til kundemøtet 	<ul style="list-style-type: none"> - Glemte å si til kunde at vi ikke ville bli ferdig med alle brukerhistoriene

Tabell 5: Sprint-retrospektiv 4

Hva fungerte godt?	Hva kunne blitt gjort bedre?
<ul style="list-style-type: none"> - Bedre oversikt over flyten i programmet - Gruppen har jobbet jevnt og bra - Bedre forberedt til kundemøtet - Gøy å kode 	<ul style="list-style-type: none"> - Mangel på avsatt tid til testing, burde vært gjort tidligere - Test-casene burde vært ferdigstilt tidligere - Sprint-planleggingen var av litt dårlig kvalitet

Tabell 6: Sprint-retrospektiv 5

I Brukermanual

For å kunne bruke nettsiden må du logge inn som en bruker med de ønskede rettighetene. Hvis du bruker testdatasettet kan du finne login informasjon under `dummy_data` i dokumentasjonen.

I.1 Inn- og utlogging

For å logg inn og ut av systemet gjennomføres følgende trinn:

1. Gå til nettsiden, dersom den kjøres lokalt på din maskin finnes den på `http : //localhost : 8000/`
2. Trykk på "Login" øverst i det høyre hjørnet av vinduet.
3. Fyll inn gyldig brukernavn og passord i feltene, og trykk "Log in".
4. Du er nå logget inn, og kan se gruppen du tilhører øverst til høyre.
5. For å logge ut igjen trykker du på din gruppe for å åpne en nedtrekksmeny.
6. I denne menyen kan du trykke "Logout", og du vil logges ut av systemet.

I.2 Opprettelse av grupper og brukere

For at brukerne skal kunne benytte systemet, må en administrator først opprette brukergrupper med tilhørende brukerkontoer. I vårt sett med testdata er dette allerede gjort. Nedenfor følger en trinnvis gjennomgang av hvordan dette gjøres, med tekniker-gruppen som eksempel.

1. Logg inn som administrator.
2. Gå til `http : //localhost : 8000/admin`
3. Trykk på "Grupper".
4. Trykk på "Legg til ny gruppe+".
5. Skriv inn gruppenavn i navnefeltet.

6. I dette eksempelet skal det skrives "technician".
7. Andre gyldige gruppenavn kan finnes i `dummy_data` i dokumentasjonen.
8. Gå tilbake til adminpanelet, ved å trykke på "Hjem" oppe i høyre hjørne.
9. Trykk på "Brukere".
10. Trykk på "Legg til ny bruker+".
11. Fyll inn brukernavn.
12. Fyll inn og bekreft passord.
13. Trykk på "Lagre og fortsett å redigere" nederst på siden.
14. Velg technician-gruppen i grupper-feltet.
15. Trykk "Lagre".

I.3 Opprettelse av sjanger, artist, konsert og scene

Sjangere, artister, konserter og scener må også opprettes i adminpanelet. Fremgangsmåten for opprettelse av disse er veldig lik, så under følger kun instruksjoner for å opprette artist. Dette bør gi tilstrekkelig innføring i hvordan prosessen gjennomføres.

1. Logg inn som administrator.
2. Gå til `http://localhost:8000/admin`
3. Trykk på "Artists".
4. Trykk på "Legg til ny artist+".
5. Skriv inn navn på artisten i navnefeltet.
6. Velg sjanger fra nedtrekkslisten (du må ha opprettet sjanger før artist).
7. Velg artistsmanager fra nedtrekkslisten (du må ha opprettet artistmanager før artist).
8. Skriv inn en anmeldelse hvis det er ønskelig.

9. Fyll inn kontaktinformasjon.

10. Trykk "Lagre".

Opprettelse av sjanger, konsert eller scene gjøres på samme måte. Opprett nye objekter ved å fylle ut de nødvendige feltene. Det eneste som er ulikt for de forskjellige objektene er hvilke felter som må fylles inn.

I.4 Gi organizer/arrangør tilgang på adminpanelet

Vi har gjort det slik at kun administrator har tilgang på opprettingsfunksjonene som er beskrevet ovenfor, men dersom det er ønskelig er det ikke vanskelig og gi en arrangør tilgang på de samme funksjonene. Nedenfor følger en beskrivelse av hvordan dette gjøres.

1. Sørg for at organizer-gruppen er opprettet, og at en arrangør har gruppen oppført på sin bruker.
2. Velg arrangørens bruker fra brukeroversikten på `http://localhost:8000/admin/auth/user/`
3. Velg avmerkningsboksene som heter "administrasjonsstatus" og "superbruker".
4. Trykk "Lagre".

Resten av funksjonaliteten til programmet er tilgjengelig uten bruk av adminpanelet. Majoriteten av funksjonaliteten skal være selvforklarende og intuitivt, men vi har allikevel valgt å presentere et utvalg av brukerhistoriene i avsnittene nedenfor.

I.5 Opprette timeslots

Timeslots er måten vi har valgt å påse at to konserter ikke skal kunne eksistere på samme scene til samme tid. Bookingansvarlig, bookingsjef og arrangør har mulighet til å opprette timeslots. Dette gjøres på følgende måte:

1. Logg inn som en av brukerne nevnt over, med tilgang til å opprette timeslots.
2. Trykk på elementet i navigasjonsbaren som heter "Booking Overview".
3. Trykk på "New Timeslot" mot bunnen av siden.
4. Fyll inn ønsket dato og tid.
5. Velg scene fra nedtrekksmenyen.
6. Trykk "Create".
 - (a) Dersom det allerede eksisterer en timeslot som overlapper med den du har laget vil du få en feilmelding.

Endringer i timeslots gjøres nesten likt, men da ved å trykke på "Edit"-knappen i oversikten over timeslots.

I.6 Få en totaloversikt over alle konserter og hvilke scener de spilles på

Nedenfor følger en trinnvis beskrivelse av hvordan du som arrangør kan se en totaloversikt over samtlige konserter.

1. Logg inn som arrangør.
2. Trykk på "Program" i navigasjonsbaren.

I.7 Sende bookingtilbud til godkjenning av bookingsjef

Nedenfor følger en trinnvis beskrivelse av hvordan bookingansvarlig kan opprette bookingtilbud, og tilgjengeliggjøre disse for godkjenning av bookingsjef.

1. Logg inn som bookingansvarlig.
2. Trykk på "Booking" i navigasjonsbaren.
3. Trykk på "New" øverst i høyre hjørne.
4. Fyll inn alle påkrevde felter.
5. Trykk "Create".

I.8 Godkjenne bookingtilbud som bookingsjef før de går ut til artistmanager

Nedenfor følger en trinnvis beskrivelse av hvordan bookingsjef kan godkjenne bookingtilbud opprettet av bookingansvarlig, før de sendes ut til artistmanager.

1. Logg inn som bookingsjef.
2. Trykk på "Booking Overview" i navigasjonsbaren.
3. Finn tilbudet du ønsker å godkjenne, og trykk på "Approve".
4. Tilbudet vil nå dukke opp for artistens manager.

I.9 Se oversikt over kontaktinfo, omtale og salgstall for artister som er booket

Nedenfor følger en trinnvis beskrivelse av hvordan PR-ansvarlig kan få oversikt over kontaktinfo, salgstall og omtale for ulike artister.

1. Logg inn som PR-ansvarlig
2. Trykk på "Festival Overview" i navigasjonsbaren.

J Teknisk brukermanual

Python 3, pip, virtualenv og PostgreSQL er tekniske krav. Python 2.7 er ikke støttet. Kildekoden kan hentes fra:

[https : //github.com/aukcode/concerthub/](https://github.com/aukcode/concerthub/)

For oppsett og konfigurering av PostgreSQL følges instruksene i PostgreSQL i vedlagt dokumentasjon.

For oppsett og konfigurering av prosjektet følges README.md i GitHub-repoet. Instruksene for Linux-oppsett er også gjengitt nedenfor:

```
$ virtualenv -p python3 it1901
$ cd it1901
$ source bin/activate
$ mkdir src
$ cd src

$ git clone '[repo link]' .

$ pip install -r requirements.txt

$ python manage.py runserver

$ python manage.py makemigrations
$ python manage.py migrate
```

Når dette er gjort skal prosjektet fungere. Før bruk må man enten lage en såkalt *superuser*, for å opprette andre brukere, eller laste inn et sett med testdata. Testdata kan enkelt lastes inn med følgende to kommandoer:

```
$ python manage.py loaddata concerthub/fixtures/01_data.json

$ python manage.py loaddata booking/fixtures/01_data.json
```


Mer info om Django data-fixtures og hvordan vi bruker dette kan finnes i dokumentasjonen i `django_fixtures` i dokumentasjonen.

For å lage en superuser, altså ikke bruke våre prekonfigurerte testdatasett, kan man kjøre følgende kommando:

```
$ python manage.py createsuperuser
```