Visjonsdokument ­– Harmoni

System for konsertarrangement

Anvendt Informasjonsteknologi, IDI, NTNU

# Revisjonshistorikk

|  |  |  |  |
| --- | --- | --- | --- |
| **Dato** | **Versjon** | **Beskrivelse** | **Forfatter** |
| 05.12.2019 | 1.0 | Første utkast | Jørgen Aune, Gaute Furenes, Stian Liland, André Henriksen |
| 11.12.2019 | 2.0 | Orientert produktet mot arrangører | Stian Liland |
| 20.12.2019 | 2.1 | Korrigeringer opp mot tilbakemelding fra faglærer | Jørgen Aune |
| 28.12.2019 | 2.2 | Lagt til en del funksjonelle krav | Jørgen Aune |
| 28.12.2019 | 2.3 | Små endringer samt korrekturlesing | Gaute Furenes |
| 03.01.2020 | 2.4 | Små justering av ikke-funksjonelle krav | Grethe Sandstrak  Alexander Holt |

# Innholdsfortegnelse

[Revisjonshistorikk 1](#_Toc28469020)

[Innholdsfortegnelse 2](#_Toc28469021)

[1 Innledning 4](#_Toc28469022)

[2 Sammendrag – problem og produkt 5](#_Toc28469023)

[2.1 Problemsammendrag 5](#_Toc28469024)

[2.2 Produktsammendrag 5](#_Toc28469025)

[3 Overordnet beskrivelse av interessenter og brukere 7](#_Toc28469026)

[3.1 Oppsummering interessenter 7](#_Toc28469027)

[3.2 Oppsummering brukere 7](#_Toc28469028)

[Hva er forskjellen på en bruker og en eventplanlegger? Hvorfor skal en admin opprette kategorier for feil? Feil! Bokmerke er ikke definert.](#_Toc28469029)

[3.3 Brukermiljøet 7](#_Toc28469030)

[3.4 Sammendrag av brukernes behov 7](#_Toc28469031)

[3.5 Alternativer til vårt produkt 10](#_Toc28469032)

[4 Produktoversikt 10](#_Toc28469033)

[4.1 Produktets rolle i brukermiljøet 10](#_Toc28469034)

[4.2 Forutsetninger og avhengigheter 10](#_Toc28469035)

[5 Produktets funksjonelle egenskaper 11](#_Toc28469036)

[5.1 Alternativer til utvidelser 13](#_Toc28469037)

[5.2 Ikke funksjonelle egenskaper og krav 13](#_Toc28469038)

[5.3 WIKI-struktur 14](#_Toc28469039)

[5.4 Domenemodell 14](#_Toc28469040)

[5.5 User stories 15](#_Toc28469041)

[5.6 Brukertester og konseptuelle modeller 15](#_Toc28469042)

[5.7 Arkitektur 15](#_Toc28469043)

[5.8 Prosjektstruktur 16](#_Toc28469044)

[5.9 Klassediagram 16](#_Toc28469045)

[5.10 Server-tjenester 16](#_Toc28469046)

[5.11 Databasemodell 16](#_Toc28469047)

[5.12 Dokumentasjon av kildekode 16](#_Toc28469048)

[5.13 Installasjon og kjøring 16](#_Toc28469049)

[5.14 Kontinuerlig integrasjon og testing 16](#_Toc28469050)

[5.15 Sikkerhet 16](#_Toc28469051)

# Innledning

Dette dokumentet beskriver overordnede krav til prosjektoppgaven i emnet TDAT 2003 Systemutvikling 2 med webapplikasjoner. Oppgaven består i å utvikle webapplikasjonen Harmoni. Prosjektet skal utføres av 2. år dataingeniør i starten av vårsemesteret 2020.

# Sammendrag – problem og produkt

## Problemsammendrag

Vi i Harmoni arrangerer flere konserter i året, hvor vi hver gang må gå gjennom den samme prosessen for å finne artister, booke artister, fikse hotell og transport, promotere, arrangere selve konserten, oppvartning til artister, betaling, osv. Denne prosessen går gjennom flere ledd og kan raskt ende med dårlig informasjonsflyt og mangel på kommunikasjon. Ved hjelp av Harmoni-systemet vil arrangører kunne opprette konserter, markere hvor man er i bookingprosessen, osv., og alle parter har ett felles sted for å få informasjon. Per dags dato blir det sendt mail med dokumenter og kontrakter mellom de ulike partene.

|  |  |
| --- | --- |
| **Problem** **med** | Mangel av system for strukturert planlegging og booking av musikkarrangementer |
| **berører** | Alle som er med på å arrangere konsert. |
| **som resultat av dette** | Kan det bli dårlig intern kommunikasjon angående konsertene som skal arrangeres. Informasjonen blir delt på epost og kan føre til dårligere konsertopplevelser og misfornøyde artister. |
| **en vellykket løsning vil** | Gjøre planlegging og gjennomføring av konserter lettere ved å samle alle info om konserter i én og samme løsning, hvor alle parter har oversikt. |

## Produktsammendrag

Harmoni er ment for bruk av konsertarrangører. Dette skjer via et innloggingssystem. Arrangørene vil ha muligheter for å opprette arrangement, velge hva som trengs av personell og riders. Hvert arrangement skal ha en kontrakt, og det skal opprettes ulike billettkategorier for arrangementet. Kort fortalt skal hvert arrangement opprettes med alt av nødvendig informasjon. Systemet skal kunne brukes uavhengig av hvor man er i Norge.

|  |  |
| --- | --- |
| **For** | Arrangører i musikkbransjen. |
| **Som** | Lar brukeren opprette arrangement for konserter og lar alle parter legge inn vesentlig og mindre viktig informasjon for gjennomføring av konserten. |
| **Harmoni** | Er et system som er tilgjengelig på tvers av plattformer og nettlesere, minimum for nettleserne Google Chrome og Mozilla Firefox. |
| **I motsetning til** | Tidligere manuelt system som er tungvint og gir dårlig oversikt. |
| **Er det nye produktet** | Webbasert, kryssfunksjonelt med tanke på plattformer, som er mer effektiv og enklere for arrangører å bruke. Samler alle konserter i en og samme løsning. |

# Overordnet beskrivelse av interessenter og brukere

## Oppsummering interessenter

Her beskriver vi de ulike interessentene og hvilke roller de har i Harmoni, samt hvilken rolle de skal ha under utviklingen av systemet.

|  |  |  |
| --- | --- | --- |
| **Navn** | **Utdypende beskrivelse** | **Rolle under utviklingen** |
| Sluttbruker | Sluttbruker av systemet | Bistår med innspill og er sentrale under testing. |
| Veiledere | Faglærere | Bistår prosjektgruppa hvor det er nødvendig. |
| Prosjektgruppe | Utviklere av systemet | Står for selve utviklingen av systemet. |
| Oppdragsgivere | Produkteiergruppe | Har rollen som «kunde» og kommer med krav og ønsker som utføres av prosjektgruppa. |

## Oppsummering brukere

|  |  |  |  |
| --- | --- | --- | --- |
| **Navn** | **Utdypende beskrivelse** | **Rolle under utviklingen** | **Representert av** |
| Bruker | Bruker som registrerer seg i systemet. | Testing | Utviklerne, eventuelt eksterne personer. |
| Administrator | Opprette kategorier for feil. | Testing | Utviklerne |

## Brukermiljøet

Alle brukere bruker løsningen med valgfri nettleser både på PC og mobil, derfor må plattformen fungere i de vanligste nettleserne og ha støtte for mobilvisning og navigering.

## Sammendrag av brukernes behov

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Behov** | **Prioritet** | **Vedrører** | **Dagens Løsning** | **Foreslått løsning** |
| Registrere bruker | Høy | Innlogging | Ingen | Selvregistrering hvor bruker velger brukernavn og passord |
| Endre passord | Lav | Innlogging | Ingen | Mulighet for å endre for innlogget bruker. Må bekrefte med dagens passord |
| Innlogging | Høy | Innlogging | Ingen | Innlogging med brukernavn og passord. |
| Registrere arrangement | Høy | Arrangement | Etter kontrakt er signert, sendes det mail til ledelsen om at det blir arrangement | Egen side i systemet med mulighet for å registrere arrangement med nødvendig informasjon. |
| Slette/arkivere arrangement | Høy | Arrangement | Ingen | Ved feil, eller etter et arrangement skal det være mulig å arkivere dette arrangementet. |
| Redigere arrangement | Høy | Arrangement | Ingen | For hvert arrangement skal det være mulighet for å gå inn og redigere registrert informasjon om arrangementet. |
| Kartvisning på arrangement | Middels | Arrangement | Ingen | Integrert google-kart som viser lokasjonen til arrangementet. Vises for hvert arrangement. |
| Legge til billettkategori | Høy | Arrangement | Skrives på mail til de som setter opp billetter | Ulike kategorier på billettene. Eksempel: E*arly Bird, VIP, standard.* |
| Laste opp kontrakt | Middels | Booking | Lagres i google disk | Kontrakt skal kunne lastes opp i systemet. |
| Registrere behov for riders\* | Middels | Booking | Lagres i google disk | Man velger nødvendige riders fra en liste over alle tilgjengelige riders. |
| Personell (liste på hva man trenger av folk, eks. Lysman, lydman, artistkontakt) | Middels | Arrangement | Scenesjef setter opp det man trenger av personell | Man velger nødvendig personell til arrangementet. |
| Søke opp arrangement | Lav | Arrangement | Ingen | Et søkefelt gjør at man kan søke opp arrangement basert på arrangementsnavn eller lokasjon for arrangementet. |
| Kalender | Lav | Arrangement | Excel dokument | En egen side med kalender som viser kommende arrangement (i nærmiljøet) |

\*Rider beskriver en musikers liste over tekniske og praktiske nødvendigheter og ønsker i forbindelse med en fremføring. En rider er et tillegg i kontrakten som inngås med arrangøren.

## Alternativer til vårt produkt

Det eksisterer noen alternativer til slike sider for å opprette og administrere arrangementer på nett. Derimot vil vi ha en rettet mot konsertarrangører. Eksempler:

[Eventbrite](http://www.eventbrite.com/)

[Facebook](http://www.facebook.com/events/)

[Ticketmaster](https://www.ticketmaster.no/)

# Produktoversikt

## Produktets rolle i brukermiljøet

Skisse av produktet med hensyn til omgivelsene, ta utgangspunkt i brukermiljøet (4.3)

Et bilde som inneholder skjermbilde

Automatisk generert beskrivelse

## Forutsetninger og avhengigheter

Registrering av bruker forutsetter at bruker har en e-postadresse. Alle som vil kan registrere seg selv som bruker i systemet.

# Produktets funksjonelle egenskaper

En liste over funksjonelle egenskaper som produktet må ha for å løse problemene beskrevet i 3.4. NB! Det vil ikke nødvendigvis være et en-til-en-forhold. De funksjonelle egenskapene må være beskrevet slik at de danner et godt utgangspunkt for et use case-diagram (og dermed også tekstlig use case).

|  |  |
| --- | --- |
| **Funksjonelle egenskaper** | **Beskrivelse** |
| Funksjon for å opprette ny bruker | Det skal være mulig å opprette en ny bruker i systemet, hvor typisk e-postadresse og et valgt passord vil brukes for å skille brukerne. |
| Funksjon for å logge inn | Når bruker er opprettet skal det være mulig å logge seg inn med valgt brukernavn(e-post)/passord. |
| Endre passord | Det skal være mulig å endre passord til sin egen bruker. |
| Redigere egen bruker | Man skal ha mulighet til å oppdatere sin egen profil, og det viktigste er kontaktinformasjon som mobilnummer, e-post osv. |
| Laste opp profilbilde | Det kan være nyttig å ha mulighet for å ha et profilbilde for å gjøre det lettere for andre å knytte navn opp mot et ansikt. |
| Opprette arrangement | For å opprette arrangement må man kunne legge til sted, tid, type, billettkategorier, antall billetter, personell, riders osv. |
| Endre arrangement | Det skal være mulig å endre tidspunkt, sted, pris osv. |
| Avlyse arrangement | Man skal kunne avlyse arrangementer, og automatisk gi beskjed til aktuelle interessenter via e-post. |
| Slette arrangement | Gjennomførte arrangementer må kunne slettes fra systemet etter en tid. |
| Sortere arrangementer | Man må kunne sortere arrangementer på eksempelvis pris, størrelse, sted osv. |
| Søke etter spesifikt arrangement | Man må kunne søke etter arrangementer på sted, navn og andre relevante attributter. |
| Knytte artist(er) til arrangement | Man må kunne legge til artister til hvert arrangement. |
| Opprette/legge til riders | Alle artister skal kunne få mulighet til å komme med krav og ønsker (riders). Det kan være små ting som en brus, kaffe osv. |
| Redigere rider | Artister og arrangører må kunne endre tilgang på riders frem til arrangementstart, og bli enig om de premissene som gjelder. |
| Funksjonalitet for å knytte alle arrangementer opp mot kartfunksjon (google maps) | Alle arrangementer skal være knyttet til en adresse, og dette burde visualiseres med en kartfunksjon slik at man kan se visuelt hvor konserten finner sted. |
| Mulighet for å gi tilbakemelding om feil/bugs som skulle oppstå | Alle brukere av systemet skal ha mulighet til å gi direkte tilbakemelding på feil, og andre utfordringer de måtte støte på. |
| Mulighet for å lagre dokumenter | Det burde være en mulighet for brukeren å se relevante dokumenter for seg selv. Eks: Arrangøren burde ha mulighet til å finne og lese kontrakten og listen over riders mellom seg selv og artistene. |
| Mulighet for å laste opp og ned dokumenter fra systemet | Brukerne skal ha mulighet til å laste opp/ned relevante dokumenter de måtte trenger. Eks, rider |

## Alternativer til utvidelser

Legge til en brukergruppe for publikum/kunde, som kan kjøpe billetter og få alt av informasjon om konserter fra Harmoni. I fremtiden vil vi også kunne bruke Harmoni til å markedsføre kommende arrangementer via Harmoni. Legge til et sosialt aspekt for publikum, la brukeren stille spørsmål om arrangementet.

I senere tid hadde det også vært spennende å se på mulighetene for å knytte frivillige opp mot systemet slik at de kan få alt de trenger av informasjon om arrangementet fra systemet.

## Ikke funksjonelle egenskaper og krav

Det kreves at enhetstesting er implementert. På serversiden skal det være minst 50% dekningsgrad. På klienten er det ikke krav til dekningsgrad, men det er ønskelig med tester her også.

Alle data som benyttes av applikasjonen Harmoni skal lagres i skolen MYSQL-database. Databasefunksjonaliteten skal tilgjengeliggjøres for klienten via REST-tjenester.

En plattformuavhengig nettleser med støtte for en nyere HTML-standard må kunne brukes som klient mot applikasjonen. Minimumskrav er støtte for både Chromium (Chrome) og Firefox, så begge deler må sjekkes.

Responstiden bør være under et sekund selv med 100 samtidige brukere.

Løsningen skal ha god brukskvalitet, den skal være tilpasset målgruppen, og lett og intuitiv å bruke. Dette skal dokumenteres.

Løsningen skal ha god sikkerhet. Minstekrav er at løsningen sjekkes opp mot OWASP (2013) A1 og A3.

All kode skal lagres på skolens Gitlab (https://gitlab.stud.iie.ntnu.no/)

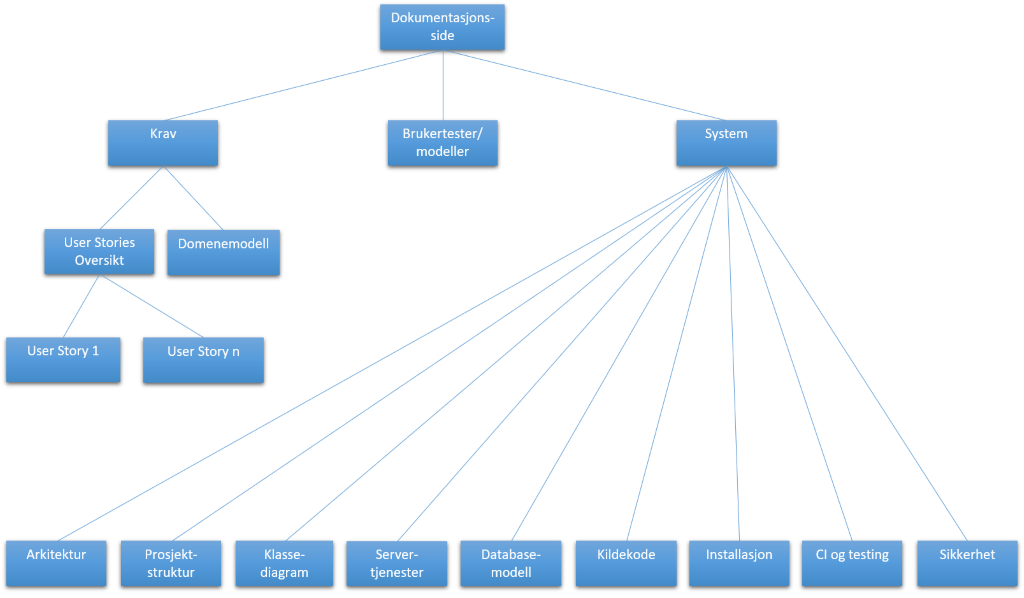
Sluttrapporten leveres som en PDF og skal inneholde følgende:

* Backlog/Sprintbacklog for sprint 1 og 2
* Sprintmål for sprint 1 og 2
* Retrospektiv for sprint 1 og 2
* Oppsummering av sprintene
* Individuelt refleksjonsnotat for hver deltaker i prosjektgruppen der en beskriver egen innsats i prosjektet og hvordan en opplevde samarbeidet i prosjektgruppen

Følgende dokumentasjon skal leveres som WIKI i GitLab

* Krav
* Systemdokumentasjon
* Brukertester / konseptuelle modeller for brukergrensesnitt og lignende

## WIKI-struktur



## Domenemodell

Lag en enkel domenemodell som beskriver problemdomenet.

## User stories

Lag user stories som dekker all brukerfunksjonalitet for systemet. En user story kan skrives på denne formen:

Som <hvem?>

Ønsker jeg <hva?>

Slik at <hvorfor?>

Et eksempel kan være:

Som administrator

Ønsker jeg å opprette brukerkontoer

Slik at jeg kan gi nye brukere tilgang til systemet

Ta også med en kort liste med akseptansekriterier (krav) til hver user story. For eksempel:

* Gyldig epost, navn og telefonnummer må skrives inn
* Hvis epost er registrert fra før skal det vises en feilmelding.
* Systemet skal sende epost til den nye brukeren med et system-generert passord.
* Det systemgenererte passordet skal være på 12 tegn og bestå av helt tilfeldige tall eller små/store bokstaver.
* Passordene skal krypteres med «hashing med salt».

## Brukertester og konseptuelle modeller

Beskriv brukertester som er foretatt og eventuelle konseptuelle modeller for brukergrensesnitt som er brukt.

## Arkitektur

Lag en arkitekturskisse som beskriver de viktigste komponentene i systemet: klienter, server-tjenester, datakilder og kommunikasjonen mellom disse. Vis også eventuell lagdelt arkitektur for større komponenter.

En kort tekstlig beskrivelse bør være med for hver figur.

## Prosjektstruktur

Beskriv hvilke biblioteker og rammeverk dere bruker og vis fil- og katalogstrukturen for prosjektet, kildekode, pakker, og hvordan disse er organisert.

## Klassediagram

Lag et overordnet klassediagram for serverdelen av løsningen. Hvis klasser ikke er i bruk kan dere levere en oversikt over moduler som er brukt.

Klassediagrammet skal også ha en tekstlig forklaring.

## Server-tjenester

Beskriv de ulike REST-ressursene. Dette gjelder også for eksempel WebSocket-endepunkter.

## Databasemodell

Vis en databasemodell her.  
Gi en kort forklaring av databasemodellen.

## Dokumentasjon av kildekode

Her kan dere beskrive og vise til dokumentasjon av kildekode. Ta med hvordan en eventuelt genererer og viser dokumentasjonen.

## Installasjon og kjøring

Lag installasjonsveiledning. Her må alt faglærer eller andre trenger å vite for å kunne installere og kjøre løsningen tas med.

## Kontinuerlig integrasjon og testing

Beskriv kontinuerlig integrasjon-oppsettet prosjektet benytter.

Beskriv også generelt testene som er laget; hvilke typer tester som er med, hva som er, eller ikke er, testet, og testdekning. Legg ved instruksjoner på hvordan en kjører testene.

## Sikkerhet

Beskriv kort hvordan sikkerheten i løsningen er ivaretatt.