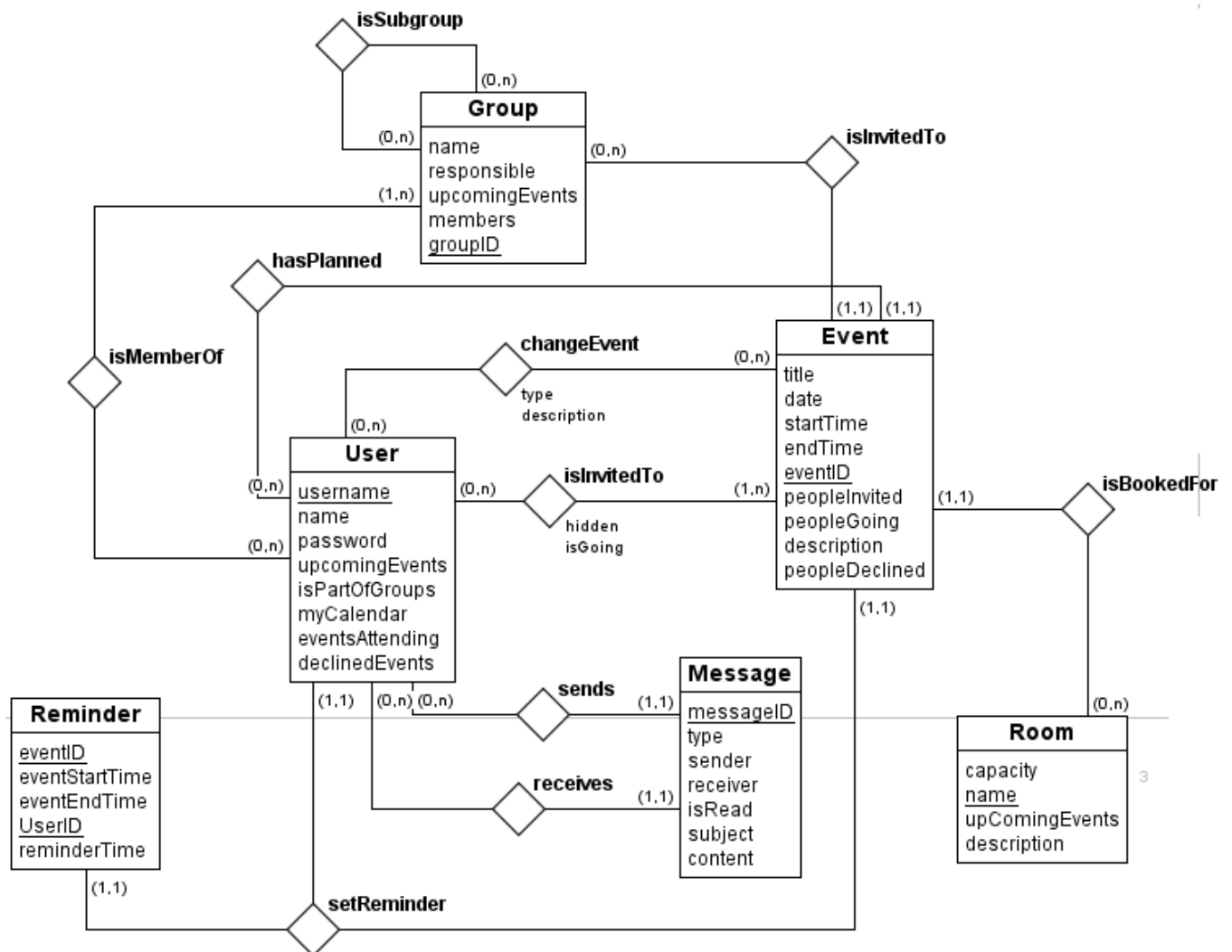


ER-modell av systemet



Krav til kalendersystemet m/forklaring

Hver ansatt i Firma X skal ha en personlig kalender. Hver gruppe skal ha en gruppekalender. En gruppe kan bestå av flere personer og også av subgrupper. En person kan være medlem av flere grupper.

1. Logge på. Ansatte får tilgang til kalendersystemet ved å logge seg på kalenderklienten med brukernavn og passord

Har login-metode i Calendar-klassen som ber brukeren om å skrive inn brukernavn og passord i konsollen. Metoden itererer gjennom brukerne av systemet og sjekker om brukernavn+passord-kombinasjonen stemmer med noen av brukerne i systemet. Dersom brukernavn + passord skrives inn riktig, blir denne brukeren til den nåværende brukeren (current_user).

2. Legge inn avtale. Ansatte skal kunne legge inn avtaler i kalendrene sine. En avtale legges inn på avtaledato med et start- og sluttidspunkt, samt en kort beskrivelse av avtalen ("Bil på verksted") og eventuelt sted for avtalen ("Strandveien Auto"). Det skal være lov å legge inn flere avtaler for overlappende tidspunkt.

Employee-klassen har en `addEvent`-metode som legger til nytt event. Event-objekter instansieres med bl.a. `startTime`-, `endTime`- og `description`-felt i konstruktøren. Programmet vil ta inn input fra brukeren som blir parametre i Event-konstruktøren. Deretter kalles `addEvent()`.

3. Håndtere møtedeltakere. Den som har lagt til en avtale skal også kunne legge til (eller fjerne) (potensielle) deltakere, ved å angi enkeltpersoner og/eller grupper. Grupper administreres utenfor systemet og kan ses på som en samling personer uten noen spesiell struktur (roller/hierarkier). Inviterte deltakerne skal kunne bekrefte evt. avkreffe at de deltar. Dersom en deltaker avkrefter deltakelse, så kan han/hun også velge om den skal skjules i kalenderen sin.

Har metoder i Employee-klassen for å invitere både hele grupper og personer samt trekke invitasjonene. Dette skjer i form av å legge til- og slette deltakere fra mengden inviterte deltakere. Employee-objekter har felt for mengden av eventer de er invitert til og metoder `acceptInvitation(Event)` og `declineInvitation(Event)` for å bekrefte/avkreffe om de vil delta eller ikke. Employee-objekter vil også ha en mengde av Events de er invitert til, men hvor de ikke har svart på invitasjonen (som dukker opp i kalenderen), og en mengde Events som de har takket ja/nei til.

4. Slette avtale. Ansatte skal kunne slette avtaler de har opprettet, og disse forsvinner da fra kalenderen til alle deltakere.

(se `employee.cancelEvent(Event)`)

Event-objekter har `creator`-felt som angir hvem som har opprettet event'et. Da kan Employee-objekter iterere over `eventsAttending` og trekke ut de event'ene der man selv er "creator". Dersom event'et man forsøker å slette er i denne mengden, fjernes det fra `upcomingEvents`-mengden. Samtidig itererer man over alle deltakere på event'et og fjerner event'et fra hver enkelt sin `upcomingEvents`- eller `eventsAttending`-mengde. Man sender ikke varsel til deltakere som ikke har sagt at de kommer på møte, kun de som kommer.

5. Endre avtale. Ansatte skal kunne endre på avtaler de har opprettet. Alle feltene kan endres. Alle deltakere blir varslet om endringen (på en eller annen måte), så de kan forholde seg til det og evt. endre på om de deltar eller ikke.

Har set'ere for hvert felt i Event. Bruker observatør-observert-teknikk for å gi beskjed til alle deltakere om endringen.

6. Reservere møterom. I stedet for å skrive inn sted for en avtale eller et møte, skal kunne velge møterom blant de som er ledige i det angitte tidsrommet. Ved av tidsrommet, skal kalendersystemet automatisk reservere møterommet for nye tidsrommet, men ikke holde på det om andre har reservert det da. Det være mulig å angi antall deltakere som rommet skal kunne ta, uavhengig av antallet deltakere som er invitert og evt. har bekreftet at de kommer.

Hvert Event har et room-felt som spesifiserer møterommet. Hvert Room-objekt har en liste av events 'upcomingEvents' (sortert på starttidspunkt) som angir indirkete når det er opptatt. Når man setter møterommet for en event må man derfor være påpasselig med å ikke velge et møterom som er opptatt til tidspunktet man ønsker å holde møtet på.

7. Visning. Kalenderklienten skal vise en ukeoversikt der alle avtaler og møter i den personlige kalender vises. Det skal være et tydelig skille mellom avtaler som en har opprettet selv og avtaler som en er invitert til. For avtaler en er invitert til så skal det være tydelig angitt status for egen deltakelse (om en har bekreftet/avkreftet deltakelsen). Det skal være tydelig angitt om en avtale er endret, så brukeren kan forholde seg til endringen. (Merk at det ikke er definert hva som gjør at systemet tror brukeren har forholdt seg til endringen.)

Hvert Employee-objekt har lister over upcomingEvents, attendingEvents og declinedEvents. Disse skal være sortert på starttidspunkt. Da er det enkelt å se hvilke events man har godtatt/avslått invitasjon til. Hvert event har et felt 'creator' som angir hvem som har opprettet event'et. Ved å iterere over alle attendingEvents, kan man hente ut de med creator-felt lik seg selv. Vi genererer en ukeoversikt over ønsket uke i ønsket år, spesifisert av brukeren der events man har sagt at man kommer på blir markert med 'A' og events man ikke har takket ja/nei til ligger med 'U'.

8. Status for deltakelse. For møter med inviterte deltakere så skal det være tydelig angitt om 1) en alle har svart eller ikke og 2) om det er noen som har avslått invitasjonen.

Hvert Event inneholder felt peopleGoing, peopleInvited og peopleDeclined. Se evt punkt 7 for mer info.

9. Melde avbud for møte. En ansatt kan melde avbud på en møteinnkalling ved å slette avtalen i sin personlige kalender. Når en ansatt melder avbud, sendes melding til alle de andre møtedeltakerne. Møteleder kan da velge om møtet skal avlyses eller om han/hun skal endre tidspunkt på møtet.

Employee-klassen har metode declineInvitation(Event) som gir beskjed til et event om den ansatte deltar eller ikke. Da sendes det automatisk en Message til andre møtedeltakere.

10. Reservere møterom. I stedet for å skrive inn sted for en avtale eller et møte, skal brukeren kunne reservere møterom. Databasen skal lage en liste med tilgjengelige møterom (tilgjengelig betyr ikke reserverte) i tidsperioden for avtalen/møtet. Brukeren kan da velge møterom fra denne listen. Om en avtale med reservert møterom slettes, skal reservasjonen slettes på databasen. Det samme gjelder for møter som avlyses.

Se punkt 6 for hvordan lista over ledige møterom hentes. Må passe på at cancelEvent-metoden fjerner event'et fra lista til Room-objektet over upcomingEvents.

11. Visning. Kalenderklienten skal vise en ukekalender der alle avtaler og møter i den ansattes personlige kalender vises. Det skal være enkelt å bla mellom ukene.

Se punkt 7.

12. Spore møteinnkallinger. Kalenderklienten skal indikere i ukekalenderen om a) en møteinnkalling venter på svar fra en eller flere deltakere, b) en eller flere møtedeltakere har avslått møteinnkalling, eller c) om alle innkalte har godtatt møteinnkallingen.

Vi holder hele tiden oversikt over hvem som har godtatt, avslått og ikke besvart event-invitasjon med listene i Event: peopleInvited, peopleGoing, peopleDeclined. Antall stk som har meldt avbud vil stå i kalenderen som "People declined (ant. declined)". F.eks. dersom to stk har avslått invitasjon, vil det markeres med "People declined (2)" i kalendervisningen. Hvis alle har godtatt: "All going".

13. Vis flere kalendere. Det skal være mulig å se kalendere (avtaler og møter) for flere ansatte i samme ukesoversikt.

Vi ser for oss at man i konsollen kan velge hvilken kalender man vil se på. Alternativene kan dukke opp med ulike navn. Navnene vil typisk være "Min kalender" (den ansattes egen personlige kalender), [gruppenavn1], [gruppenavn2]. For å se flere personlige kalendere samtidig, kan man ha et siste alternativ: "Other". Etter å ha skrevet "Other" til konsollen, kan brukeren skrive inn brukernavnene og eller gruppenavnene (så sant man er medlem av gruppa) til alle brukere og grupper man vil se kalenderene til.

14. Alarm. Det skal være mulig for hver ansatt å konfigurere enhver avtale slik at avtalen genererer en alarm en gitt tid før møtet.

Vi ser på det som meget utfordrende å få til dette når kalenderen vår kun skal vises når vi kjører programmet i eclipse-konsollen. Ettersom kalenderen ikke vil være "online", vil dette kravet i praksis måtte gå ut: har man eclipse lukket, vil jo ikke påminnelsen bli gitt til tide uansett. Derfor har vi nedprioritert dette kravet kraftig. Vi har likevel ført opp en Reminder-klasse i ER-modellen vår slik at hvis vi får tid, vil vi implementere denne funksjonaliteten også.