

Piljetter.se

Inlämningsuppgiften går ut på att bygga ett system för världsomspännande biljettförsäljning i musikbranchen. Systemet kommer att ha användare av två olika typer (a.k.a roller); kunder och administratörer. och skall tillhandahålla funktionerna listade nedan för varje roll. Det innebär att det förmodligen behövs olika "program" för de olika rollerna. Dessa program kan implementeras som enskilda konsollapplikationer, formsapplikationer eller webbapplikationer, beroende på vad DU är mest bekväm med.

Piljetter.se's huvudsakliga funktion är att ge människor möjlighet att köpa biljetter till konserter med sina favoritartister i sitt närområde. Piljetter.se använder sin egen valuta, pesetas. En kund som registrera sig på piljetter.se får ett eget pesetas-konto, som hen kan sätta in pengar på, för att sedan använda till att köpa biljetter. För att verksamheten ska gå runt ekonomiskt har ägarna av piljetter.se också tillgång till att ta fram statistik över biljettförsäljningen, de har möjlighet att boka in nya konserter och möjlighet att avlysa konserter som inte visar sig bli lönsamma.

Nedan följer en översikt över de funktioner som systemet skall tillhandahålla för de olika rollerna. Läs igenom noggrant för att identifiera vilken data du behöver lagra i databasen. Normalisering, säkerhet (främst sql-injections), prestanda och korrekthet är aspekter som bör beaktas i alla funktioner.

Administrator

En **administrator** är en person som jobbar på piljetter.se. Personen jobbar delvis med att underhålla information om **artister**, **scener** och **konserter**, och delvis med att plocka fram **rapporter** för att undersöka verksamhetens lönsamhet.

Registerera kommande konsert

En administrator skall kunna registrera en kommande **konsert**. Administratören skall kunna ange vid vilken **tid** och vilken **scen** artisten kommer att uppträda. Att tänka på:

- En artist kan inte ha konsert på två platser samtidigt. En rimlig begränsning är att en artist max kan hålla en konsert per dag.
- Det kan inte hållas konsert med två artister på samma scen under samma dag.
- En konsert kan inte registreras med ett passerat datum.
- **Biljettpris** (i **pesetas**) skall registreras för varje konsert.
- Konsertens kostnad för arrangören som baseras på artistens **popularitet** och scenens **renommé** (välj själv hur).

Ställa in konsert

En administrator skall kunna ställa in en konsert. Alla köpta biljetter bör bli refunderade. Administratörerna skall om önskas också kunna utfärda en **biljettkupong** till drabbade **kunder**. En biljettkupong kan av kund användas för köp av biljett oavsett pris. En biljettkupong skall bara kunna **användas en gång** och bara kunna användas **av den kund den blev tilldelad**, och en biljettkupong har ett **utgångsdatum**.

Kunna se statistisk

En administrator bör kunna ta ut rapporter från systemet. Välj själv vilken data (minst 3) rapporterna skall innehålla. Var noga med att rapporten ska kunna motiveras utifrån piljetter.se's perspektiv. Här kommer några exempel.

- Antal sålda biljetter i en period, tillsammans med total försäljningsintäkt. (kan användas som underlag till bokföring)
- En lista på de tio bäst säljande artisterna i en period. (kan användas för att avgöra om en artist ska bokas igen)
- En läsbar översikt över utgivna kuponger och dess utgångsdatum. (kan användas för att estimera hur många pesetas piljetter.se ligger ute med.)
- En översikt över konserters lönsamhet. (kan användas som underlag för att avlysa en konsert)

Kund

En kund är en person som vill gå på konsert och vill kunna förhandsköpa biljett till konserten. En kund måste själv registrera sig i systemet. Vi kan bortse hantering av lösenord men kunden måste identifiera sig för att kunna utföra vissa funktioner.

Registrera sig/Logga in

När en kund startar programmet måste hen antingen registrera sig eller identifiera sig (logga in). Inloggningen förblir tills programmet avslutas. Startar man programmet på nytt måste man logga in igen. En kund får en viss mängd **start-pesetas** vid registrering.

Söka efter konsert

En kund måste ges möjlighet att kunna **söka** upp konserter hen är intresserad av. Kunden bör kunna söka efter konserter på **artist, land, stad, scen, datum** eller **datumintervall**, eller **kombinationer** av dessa kriterier. Tänk att konserter som redan skett kanske inte är av intresse för kunden.

Kunna köpa biljett till konsert

En kund skall kunna välja att köpa en biljett till en konsert. Att tänka på:

- Konserten kan inte redan ha skett eller påbörjats
- Det måste finnas **biljetter kvar** till konserten
- Hur hanteras flera **samtidigt pågående köp**, speciellt när biljetterna börjar ta slut?
- En kund kan endast köpa biljetten om kunden har tillräckligt många pesetas registrerade på sitt konto, eller om hen vill nyttja en biljettkupong.

Köpa pesetas

En kund kan köpa pesetas (piljetter.se's egna valuta). Pesetas registreras på kundens konto direkt efter genomfört köp.

Kunna se lista över köpta biljetter

En kund skall kunna se en lista över biljetter hen har köpt med information om konserten biljetten gäller, när köpet genomfördes och vad biljetten kostade (eller om kupong användes)

Kunna se lista över sina kuponger

En kund skall kunna se en lista över sina kuponger med relevant information.

Bonusuppgift

Implementera fritextsök för kunden. Detta kan, med tusentals artister, städer, scener och konserter, bli komplicerat i en relationsdatabas. Du kan använda exempelvis Elastic Search (en dokumentdatabas) för att lösa detta. Vilka problem stöter du på, och vilka kompromisser måste göras?