



Bokningssystem för Kårspexet

User Requirements Document (URD)

Arvidsson, Kalle – kallear@kth.se
Boström, Peter – pbos@kth.se
Eklund, Erik – eekl@kth.se
Gräsman, André – grasman@kth.se
Göransson, Rasmus – rasmusgo@kth.se
Hagsten, Per – hagsten@kth.se
Hallberg, Victor – victorha@kth.se
Modée, Anna Maria – ammodee@kth.se
Nyberg, Daniel – dnyb@kth.se
Stjernberg, Johan – stjer@kth.se
Tarandi, Andreas – taran@kth.se

Nyx

Version 1.1 – *Revised*
January 27, 2011

Abstract

This document aims to describe the user requirements for Kårspexets online booking system, developed by Nyx. It contains the system's capabilities, constraints, assumptions, and dependencies, as well as its user characteristics, and operational environment. The document also features an extensive list of detailed requirements of the system, deduced from the preceding descriptions.

Dokumentversioner

Dokumentet har genererats från följande deldokument.

URD/abstract version: 4.

URD/Ändringslogg version: 13.

Gruppmedlemmar version: 3.

URD/Introduktion version: 4.

URD/Introduktion/Syfte version: 10.

URD/Introduktion/Mjukvarans omfattning version: 6.

URD/Introduktion/Definitioner akronymer och förkortningar version: 69.

URD/Introduktion/Källor version: 17.

URD/Introduktion/Dokumentöversikt version: 8.

URD/Allmän beskrivning version: 5.

URD/Allmän beskrivning/Produktperspektiv version: 28.

URD/Allmän beskrivning/Allmän funktionalitet version: 15.

URD/Allmän beskrivning/Allmänna begränsningar version: 45.

URD/Allmän beskrivning/Användarbeskrivning version: 45.

URD/Allmän beskrivning/Antaganden och beroenden version: 31.

URD/Allmän beskrivning/Plattform version: 12.

URD/Specifika krav version: 4.

URD/Specifika krav/Funktionalitetskrav version: 1.

URD/Specifika krav/Kravbegränsning version: 50.

URD/appendix version: 2.

Ändringslogg

| Version | Ändringar |
|---------|--|
| 1.1 | Tillkommna krav från SRD:n. Numrering av krav. |
| 1.0 | Sista utkastet för inlämning. Sista genomläsningen för enhetlig formulering. Krav formatteras konsekvent. |
| 0.6 | Konsekvent formulering. Inklusion av databasmodellen i appendix. Fåtal nya krav samt gruppering av dessa. |
| 0.5 | Allmän korrigering, strukturering och omformulering. Lite teknisk format- tering samt inklusion av gruppmedlemmar i dokumentet. |
| 0.4 | Omformuleringar. Ny version för granskning. Ny punktformatering, alla krav innehåller samma punkter. |
| 0.3 | Texter ska vara färdigskrivna. Version som skickas för granskning av URD:n. Revisionsnummer flyttade till egen punkt. |
| 0.2 | Texter under bearbetning men i stort sätt färdigkomponerade. Texter i dokumentet numreras numera med sitt revisionsnummer i dokumenthante- ringssystemet för spårbarhet. Kravdatabas innehåller kundens funktionali- tetskrav, men beskrivande texter saknas till vissa av dem. |
| 0.1 | Första sammanställd version av dokumentet. |

Grupppmedlemmar

Projektgruppen **Nyx** består av följande medlemmar.

Kalle Arvidsson – 890601-2490, kallear@kth.se

Peter Boström – 890224-0814, pbos@kth.se

Erik Eklund – 880816-0454, eekl@kth.se

André Gräsman – 890430-3214, grasman@kth.se

Rasmus Göransson – 850908-8517, rasmusgo@kth.se

Per Hagsten – 870529-0115, hagsten@kth.se

Victor Hallberg – 890121-0057, victorha@kth.se

Anna Maria Modée – 871120-0363, ammodee@kth.se

Daniel Nyberg – 900104-4495, dnyb@kth.se

Johan Stjernberg – 890315-0533, stjer@kth.se

Andreas Tarandi – 890416-0317, taran@kth.se

Innehåll

| | | |
|----------|--|-----------|
| 1 | Introduktion | 5 |
| 1.1 | Syfte | 5 |
| 1.2 | Mjukvarans omfattning | 5 |
| 1.3 | Definitioner, akronymer och förkortningar | 5 |
| 1.4 | Källor | 8 |
| 1.4.1 | Allmän kunskap av Människa-dator interaktion och användarvänlighet . . | 8 |
| 1.4.2 | Apache HTTP Server | 8 |
| 1.4.3 | Apache Software Foundation | 8 |
| 1.4.4 | CentOS | 8 |
| 1.4.5 | Installation av Passenger på CentOS 5 | 8 |
| 1.4.6 | Minimikrav för att installera och köra CentOS på en dator | 9 |
| 1.4.7 | Undersökning av prestanda för Rails | 9 |
| 1.4.8 | Webbläsarstatistik | 9 |
| 1.5 | Dokumentöversikt | 9 |
| 2 | Allmän beskrivning | 10 |
| 2.1 | Produktperspektiv | 10 |
| 2.2 | Allmän funktionalitet | 10 |
| 2.2.1 | Boka biljetter från hemsidan | 10 |
| 2.2.2 | Registrera betalningar | 10 |
| 2.2.3 | Administrera biljetter | 10 |
| 2.2.4 | Lämna ut bokade biljetter | 11 |
| 2.2.5 | Sälja biljetter direkt | 11 |
| 2.2.6 | Administrera mailutskick | 11 |
| 2.2.7 | Administrera föreställningar och teatrar | 11 |
| 2.2.8 | Administrera konton | 11 |
| 2.2.9 | Statistik | 11 |
| 2.3 | Allmänna begränsningar | 11 |
| 2.3.1 | Datamodell | 11 |
| 2.3.2 | Resurser | 12 |
| 2.3.3 | Kundbehov | 12 |
| 2.3.4 | Tekniska begränsningar | 12 |
| 2.4 | Användarbeskrivning | 13 |
| 2.4.1 | Kund | 13 |
| 2.4.2 | Säljare | 13 |
| 2.4.3 | Administratör och ekonomiansvarig | 14 |
| 2.5 | Antaganden och beroenden | 15 |
| 2.5.1 | Avgörande faktorer | 15 |
| 2.5.2 | Antaganden | 15 |
| 2.5.3 | Beräkningar | 16 |
| 2.5.4 | Slutsats | 16 |
| 2.6 | Plattform | 16 |
| 3 | Specifika krav | 18 |
| 3.1 | Funktionalitetskrav | 18 |
| 3.1.1 | Administration. | 18 |
| 3.1.2 | Bokning. | 22 |
| 3.1.3 | Diverse. | 27 |

| | | |
|----------|----------------------------|-----------|
| 3.1.4 | Loginsystem. | 28 |
| 3.2 | Begränsande krav | 29 |
| 3.2.1 | Prestanda | 29 |
| 3.2.2 | Säkerhet | 29 |
| 3.2.3 | Miljö | 30 |
| 3.2.4 | Användbarhet | 31 |
| 3.2.5 | Externa system | 32 |
| A | Databasmodell | 34 |

1 Introduktion

1.1 Syfte

Dokumentets syfte är att specificera detaljer kring projektets produkt som omfattning och funktionalitet. Den är speciellt skriven för att underlätta arbetet inom projektgruppen, men skall också kunna läsas av vår kund Kårspexet. Dokumentet redogör för vilka funktioner som ingår i de olika kravversionerna Standard, Plus och Delux av produkten. Standard omfattar vårt löfte på funktionalitet i produkten vi levererar till Kårspexet.

1.2 Mjukvarans omfattning

Produkten består av ett webbaserat biljettbokningssystem med ett enkelt användargränssnitt för besökare och ett administrationsverktyg för Kårspexet. Administrationsverktygen består av tre gränssnitt; ett för säljare, ett för ekonomiansvariga och ett för administratörer.

1.3 Definitioner, akronymer och förkortningar

Algoritm *Inom matematik och datorvetenskap är detta en begränsad uppsättning tydliga instruktioner för att utföra en uppgift.*

Apache *Syftar i detta dokument på webbservern Apache HTTP Server.*

Apache HTTP Server *Världens mest använda webbserver. Är gratis att använda. [1.3.1]*

Apache Software Foundation *Organisation som stödjer ett antal open source-projekt, bland annat Apache HTTP Server. [1.3.2]*

Applikation *I datasammanhang även kallat tillämpningsprogram. Ett dataprogram som fyller ett direkt syfte för användaren.*

Bandbredd *I vardagligt tal en storhet för hur mycket information som kan överföras på en viss tid. Vanlig enhet är Mbit/sekund.*

Bit (Binary Digit) *Den grundläggande enhet som datorer arbetar med. En bit kan anta ett utav två möjliga värden (ofta angivna som 0 eller 1).*

Byte *En vanlig enhet för informationsmängd i datasammanhang. En byte är ett paket bestående av åtta bitar.*

CentOS *Ett operativsystem baserat på Red Hat Enterprise Linux som är gratis att använda. [1.3.3]*

Databas *En databas är en samling information ordnad på ett sådant sätt att informationen i den effektivt går att hitta.*

Mail *Elektroniskt brev (engelska: email).*

Mailklient *Datorprogram för att hantera/läsa/sicka mail.*

GHz *Enhet för antalet miljarder svängningar per sekund. "G" är binärt prefix för 10^9 . "Hz" är förkortning för Hertz.*

Gränssnitt *Utformningen av kommunikationen mellan en mjukvarumodul och användare eller annan mjuk-/hårdvara.*

HTML (Hyper Text Markup Language) *Ett språk och webbstandard som används för att beskriva strukturering av text, bilder och annan media på en webbsida.*

HTTP (HyperText Transfer Protocol) *Ett standardiserat protokoll som definierar hur kommunikation över webben sker.*

Hårdvara *Även kallat Maskinvara. Ett samlingsnamn för en dators fysiska komponenter.*

Interface *Se gränssnitt.*

kB (kilobyte) *Se kbyte.*

kbyte (kilobyte) *Enhet för datamängd. "k" är prefix för 10^3 . För "byte", se Byte.*

KiB (kibibyte) *Enhet för datamängd. "Ki" är ett binärt prefix för 2^{10} . "B" är förkortning för Byte.*

Klockfrekvens *Beteckning för den hastighet i vilken en processor arbetar i.*

KTH (Kungliga Tekniska Högskolan) *Sveriges största tekniska universitet.*

Latens *Även känt som svarstid, tidsfördröjning eller lagg. Tidsskillnaden mellan en begäran och respons på begäran.*

Latency *Engelskt ord för Latens.*

Linux *Unix-liknande operativsystem. Linux är fri mjukvara.*

Logik *Vetenskapen om att dra korrekta slutsatser från givna påståenden.*

Mb (Megabyte) *Se Mbyte.*

Mbyte (Megabyte) *Enhet för datamängd. "M" är prefix för 10^6 . För "byte", se Byte.*

MHz *Enhet för antalet miljoner svängningar per sekund. "M" är binärt prefix för 10^6 . "Hz" är förkortning för Hertz.*

MiB (mebibyte) *Enhet för datamängd. "Mi" är ett binärt prefix för 2^{20} . "B" är förkortning för Byte.*

MiBit/s (mebibit per sekund) *Enhet för datahastighet. "Mi" är ett binärt prefix för 2^{20} . "Bit" är den minsta enheten för informationsmängder i datasammanhang.*

Mjukvara *Även kallat programvara. En organiserad samling av data och maskininstruktioner.*

Mjukvarubibliotek *En samling av redan existerande program eller delar av program som används för att utveckla mjukvara.*

Modul *Term för komponenter eller funktioner som går att separera från resten av systemet och som inte nödvändigtvis krävs för att systemet ska fungera som helhet.*

MVC (Model-View-Controller) *Ett koncept som bygger på att separera data (modeller), logik (kontroller) och användarinterface (vyer).*

MVC ramverk *Mjukvarubibliotek designade efter MVC-konceptet.*

MySQL *En typ av relationsdatabas baserad på SQL-standarden. Ett relationsdatabas hanteringssystem där flera användare kan arbeta med flera databaser.*

Open Source *Engelskt låneord för öppen källkod.*

Operativsystem *Ett datorprogram vars syfte är att underlätta användandet av en dator genom att vara länken mellan programvara och hårdvara.*

Passenger *I Rails-sammanhang en modul som gör det möjligt att köra Ruby on Rails på webbservern Apache.*

PHP *Ett programmeringsspråk som ofta används för att skapa webbapplikationer.*

Processor *Den komponent i en dator som utför beräkningar efter instruktioner.*

Programmeringsspråk *Språk som människor använder för att skapa datorprogram.*

Rails *I datorsammanhang vanlig förkortning för Ruby on Rails.*

Red Hat Enterprise Linux *Variant av Linux.*

Rendering *I datasammanhang (även känt som Rendring) det program som framställer en bild/animering med hjälp av beräkningar från en beskrivning.*

Ruby *Ett objektorienterat programmeringsspråk.*

Ruby on Rails *Ett abstrakt mjukvarubibliotek med öppen källkod för utveckling av webbapplikationer.*

Systemminne *Även kallat RAM (Random Access Memory), arbetsminne eller primärminne. Används för att tillfälligt lagra data som datorn arbetar med.*

Spex (Spektakel) *Humoristisk studentamatörteaterföreställning.*

SQL (Structured Query Language) *Ett språk designat för att interagera med databaser.*

Unix *Ett operativsystem som ofta används i olika typer av servrar och arbetsstationer.*

URL (Uniform Resource Locator) *Den formella benämningen av en webbadress. En text som beskriver var en viss resurs på internet finns, samt hur den går att komma åt.*

Webb *Även känt som WWW (World Wide Web). Det system som används för att hämta, visa och manipulera delar på internet. WWW utgörs av standarderna URL, HTTP respektive HTML.*

Webbapplikation *Samlingsnamn för mjukvara som användare kommer åt via en webbläsare.*

Webbläsare *Ett program som hämtar, tolkar och återger webbsidor kodade exempelvis som HTML.*

Webbserver *Program som körs på en server och distribuerar webbsidor och/eller andra filer*

som en webbläsare begär via HTTP-protokollet.

Webbsida *En fil, innehållandes exempelvis HTML, avsedd att visas av en webbläsare.*

Öppen källkod *Innebär möjlighet att ändra i konstruktionen för ett system. I ett datorprogram som har öppen källkod kan den som vill göra ändringar i programmet och utveckla det vidare.*

1.4 Källor

Referenser till de källor som använts i dokumentet är listade här under. En och samma källa kan refereras vid flera ställen i texten. En referens är på formatet [Sektion.Rubrik.Löpnummer]. Exempelvis är [2.5.1] den första (1) referensen för rubriken “Antaganden och beroenden” (5) under sektion “Allmän beskrivning” (2).

1.4.1 Allmän kunskap av Människa-dator interaktion och användarvänlighet

Användarcentrerad systemdesign-en process med fokus på en användare och användbarhet Jan Gulliksen & Bengt Göransson, Studentlitteratur 2002, Studentlitteratur AB, Lund, tryckt 2010

Hänvisning till källan görs från referenserna: [2.1.1]

1.4.2 Apache HTTP Server

<http://httpd.apache.org/>

Hänvisning till källan görs från referenserna: [1.3.1].

1.4.3 Apache Software Foundation

<http://www.apache.org/>

Hänvisning till källan görs från referenserna: [1.3.2].

1.4.4 CentOS

<http://www.centos.org/>

Hänvisning till källan görs från referenserna: [1.3.3].

1.4.5 Installation av Passenger på CentOS 5

<http://hasham2.blogspot.com/2008/07/install-phusion-passenger-on-cent-os-5.html>

Hänvisning till källan görs från referenserna: [2.3.1].

1.4.6 Minimikrav för att installera och köra CentOS på en dator

http://www.centos.org/docs/5/html/CDS/install/8.0/Installation_Guide-Support-Platforms.html

Hänvisning till källan görs från referenserna: [2.3.3].

1.4.7 Undersökning av prestanda för Rails

<http://www.rubyenterpriseedition.com/comparisons.html>

Hänvisning till källan görs från referenserna: [2.3.2].

1.4.8 Webbläsarstatistik

http://en.wikipedia.org/wiki/Usage_share_of_web_browsers

Hänvisning till källan görs från referenserna: [3.2.1].

1.5 Dokumentöversikt

Systemet som Nyx utvecklar åt Kårspexet ersätter ett gammalt system, detta behandlas i sektion 2.1. Sektion 2.2 presenterar systemets användare och går igenom tänkta användarscenarion för dessa. Projektets allmänna begränsningar behandlas i sektion 2.3, medan systemets användare beskrivs i mer detalj i sektion 2.4. De antaganden som finns angående systemets drift hittas i sektion 2.5 och de mer operativa kraven beskrivs i 2.6. I sektion 3 specificeras all funktionalitet i tabellform.

2 Allmän beskrivning

2.1 Produktperspektiv

Kårspexet vill ha ett nytt bokningssystem till sina föreställningar eftersom de är missnöjda med sin nuvarande lösning. De vill ha ett väldokumenterat system med tillgång till källkoden för att vid behov kunna vidareutveckla systemet. Lösningen måste vara så enkel att Kårspexet slipper lägga mer tid än nödvändigt på administrationen, vilket ger dem mer tid till att fokusera på andra aktiviteter som marknadsföring och att anordna bra spex.

Vårt uppdrag är att skapa ett nytt bokningssystem efter Kårspexets önskemål. Vi skall fokusera på att skapa ett enkelt och visuellt tilltalande system för Kårspexet och deras kunder. Bokningssystemet som används idag ser något föråldrat och komplicerat ut och designen är ej anpassad till övriga delar av hemsidan. Systemet körs på en extern server som Kårspexet inte har tillgång till. Det system som vi kommer att konstruera ska ha olika gränssnitt för kunder, administratörer och säljare på Kårspexets hemsida. Varje gränssnitt kommer att anpassas till sin målgrupp och dokumenteras därefter. På så sätt kommer interaktionen med hela systemet bli lättare och angenämare för alla användare.

Ett nytt bokningssystem kan hjälpa Kårspexet att höja sina intäkter genom en ökad biljettförsäljning. Icke-användarvänliga system kan få osäkra besökare att avstå från ett köp, där ett enkelt system kan locka till sig fler kunder [2.1.1]. Ett bra bokningssystem kan ge ett bättre intryck på studenter och andra besökare, vilket kan ge möjligheten att producera fler spex som leder till ytterligare intäkter.

2.2 Allmän funktionalitet

Bokningssystemet ska användas av fyra typer av användare: kund, säljare, ekonomichef och administratör. Dessa har olika roller som interagerar med varandra.

2.2.1 Boka biljetter från hemsidan

Kunder ska kunna boka biljetter från kårspexets hemsida. Efter att kunden har genomfört en bokning ska kunden få ett mail med betalningsuppgifter och bokningsnummer.

2.2.2 Registrera betalningar

Ekonomichefen ska kunna registrera betalningar för bokningar som kunder gjort.

2.2.3 Administrera biljetter

Efter att kunden har bokat och ekonomichefen registrerat kundens betalning placerar administratören ut vilka stolar kunden ska få sitta på under föreställningen. När placeringen är klar får kunden ett mail med uppmaning att hämta ut sina biljetter.

2.2.4 Lämna ut bokade biljetter

Säljare verifierar att en kunds biljetter är redo att hämtas, lämnar ut biljetterna och registrerar i systemet att biljetterna har hämtats. Gränssnittet för säljare ska vara lätt att lära sig eftersom säljarna ofta byts ut.

2.2.5 Sälja biljetter direkt

Säljaren ska kunna sälja biljetter kontant. Det är då säljaren som väljer vilka stolar kunden får sitta på.

2.2.6 Administrera mailutskick

Administratören ska kunna ändra informationen i de automatiska utskicken som sker samt kunna göra nya utskick till valda bokningars kontaktpersoner.

2.2.7 Administrera föreställningar och teatrar

När det vankas nya föreställningar är det administratören som matar in dem i systemet. Priser ska kunna ändras och om föreställningen är på en ny teater ska teatern kunna läggas till. Detta innebär att nya salongsskisser med nya sektioner och stolar ska kunna matas in. Administratören ska kunna välja vilka föreställningar det går att boka/köpa biljetter till.

2.2.8 Administrera konton

Administratören ska kunna ändra både sitt eget och andras lösenord. Säljarens lösenord ska kunna genereras automatiskt och vara giltigt en begränsad tid.

2.2.9 Statistik

Administratören och ekonomichefen ska kunna se statistik om antalet bokade och sålda biljetter för att kunna få inblick i verksamheten.

2.3 Allmänna begränsningar

2.3.1 Datamodell

Datamodellen finns bifogad med beskrivning i appendix.

2.3.2 Resurser

Vi kommer vara begränsade i vilka och hur många funktioner vi kommer kunna implementera då vi totalt är fem programmerare. På kort tid ska vi hinna implementera fyra gränssnitt för bokningssystemets användare. Gränssnitten kommer behöva testas men eftersom vi har lika många testare som vi har utvecklare kommer inte detta utgöra ett hinder för tidsplanen. En stor del av arbetet kommer behöva läggas på utvecklingen av administratörsgränssnittet då det är där de flesta och mest avancerade funktionerna kommer finnas.

Vi har inte någon budget för projektet och vi kommer inte att tillföra egna pengar för att köpa in något, detta gör att vi begränsas till att använda programvara som är gratis. Detta skulle kunna innebära ett problem i vissa projekt, men just inom webbutveckling finns det starka open source-programvaror att använda för våra ändamål.

2.3.3 Kundbehov

Nyx mål är att leverera ett fullständigt bokningssystem med alla de funktioner som Kårspexet har specificerat. På grund av systemets förväntade komplexitet och projektets tidsram kommer kvaliteten i delar av slutprodukten vara begränsad.

Gränssnittet för besökare (slutkunder) respektive säljare ska designas på ett sätt som gör att det går att använda utan några speciella förkunskaper inom vårt system. Det ska alltså fungera på ett sätt som efterliknar liknande produkter. Detta begränsar oss i hur pass många funktioner och val vi kan låta användarna exponeras för på en och samma gång. Administratörsgränssnittet är inte begränsat på samma sätt då dess användare kommer utbildas i förväg.

2.3.4 Tekniska begränsningar

Kårspexet står för den server som kommer köra vår webbapplikation. Vi har ingen kontroll över deras hårdvara, men vi har verifierat att operativsystemet som körs på servern är kompatibelt med Apache, Ruby on Rails [2.3.1] och MySQL.

Applikationen kommer inte inkludera avancerade algoritmer utan till störst del involveras mycket trafik till och från databasen. I och med att webbapplikationen och databasen körs på en och samma dator undviks eventuella begränsningar i nätverksanslutningen.

Systemet kommer enligt våra uppskattningar exponeras för upp till åtta samtidiga användare. Rails under Apache kommer i detta fall att, under godtycklig tidpunkt, använda uppskattningsvis c:a 250 MB systemminne [2.3.2]. CentOS anger 256 mb minne samt en klockfrekvens på minst 500 MHz som minimikrav för datorer som kör operativsystemet [2.3.3]. Med MySQL och Apache körandes samtidigt utöver dessa bör servern ha minst en gigabyte systemminne och en processor med klockfrekvensen 1 GHz eller högre. Kårspexets server har en processor med klockfrekvensen 2,6 GHz och 1 Gb systemminne, vilket alltså bör vara tillräckligt.

2.4 Användarbeskrivning

Produkten kommer ha fyra olika typer av användare: kund, säljare, administratör och ekonomiansvarig.

2.4.1 Kund

Teknisk bakgrund för kund

Kunder går att dela upp i två distinkta grupper: de som studerar på en teknisk högskola eller ett universitet respektive släktingar eller bekanta till Kårspexets medlemmar som inte är associerade med en teknisk högskola eller ett universitet.

Studenter

Använder datorer dagligen, antingen som en del i sin utbildning och/eller för privat bruk. Användarna är vana med mailklienter och anpassar sig i behaglig takt till nya webbsidor och gränssnitt.

Släktingar och bekanta

Den tekniska kompetensen varierar stort inom denna grupp från datorvana tonåringar till pensionärer som inte är lika datorvana. Användarna är vana med mailklienter till viss mån, men kan ta lång tid på sig att anpassa sig till nya gränssnitt.

Typscenario för kunden

Kunden besöker Kårspexets hemsida och klickar på boka biljett. En snabb och utförlig överblick av vilka föreställningar som finns och antalet platser i respektive sektion visas. Kunden får snabb återkoppling på sina val och går igenom bokningens steg; val av föreställning, val av sektion, betalningsuppgifter och bokningsbekräftelse. När kunden slutfört bokningen skickas ett mail från Kårspexet som bekräftelse. Mailet innehåller information om hur kunden kan betala sin bokning. När betalningen har registrerats kommer ett mail från Kårspexet om att hans/hennes biljett finns att hämta hos ombud.

2.4.2 Säljare

Teknisk bakgrund för säljare

Säljare är medlemmar i Kårspexet och är därmed sannolikt kårmedlemmar vid en teknisk högskola. De är först och främst aktiva med Kårspexets arrangemang och är säljare i andra hand. Det är därför viktigt att Säljargränssnittet är enkelt, då säljarna inte skall behöva någon utbildning i systemet.

Typscenario för säljare

Säljaren står i kårhuset eller på utsatt plats och loggar in på Kårspexets hemsida. Antingen säljs biljetter direkt på plats eller så kommer en kund som bokat sin biljett via hemsidan och valt att betala kontant. I båda fallen placerar säljaren ut en plats i den sektion kunden har valt och tar emot betalning för bokningen. Säljaren ger även ut utplacerade biljetter som blivit betalda till kunder som fått mailbekräftelse från Kårspexet om att deras biljett finns att hämta ut.

2.4.3 Administratör och ekonomiansvarig

Allmän teknisk bakgrund för administratören och ekonomiansvarig

Både administratören och ekonomiansvarige är studenter på en teknisk högskola, i Kårspexets fall KTH. De är därmed vana att navigera i personliga inloggningssidor, t.ex. Mina sidor, eller studera.nu. De är även vana användare av mailklienter. Deras tekniska bakgrund är uppdelad i två jämna lager; de mindre datorvana och de med lite mer datorvana.

Mindre datorvana

Organisatören kan ha en bakgrund i matematik, biologi, kemi, eller liknande ämne, där datorer inte är en väsentlig del av utbildningen. Organisatören kan hantera textredigerare väl, då han/hon är van att skriva rapporter. Det tar lite längre tid för den mindre datorkunnige att använda nya program eller anpassa sig till nya gränssnitt.

Mer datorvana

Organisatören kan ha en bakgrund i datalogi, teknisk fysik, eller liknande ämne, där datorer har varit en större del av utbildningen. Organisatören har mer erfarenhet av gränssnitt och navigering på webben. En sida söks igenom systematiskt och organisatören lär sig snabbt arbeta i nya gränssnitt och program.

Typscenario för administratören

Administratören loggar in på Kårspexets hemsida. Han/hon har uppskattningsvis 30 minuter till förfogande att jobba med sina uppgifter.

Görs ofta: placerar ut betalda bokningar, planerar föreställningar och tar hand om specialbokningar, t.ex. handikappsbokningar eller stora företagsbokningar.

Görs mer sällan: lägger till en ny teater, lägger till en ny omgång, skickar massutskick till b.l.a. kundbokningar, kollar på statistik.

Typscenario för ekonomiansvarige

Ekonomiansvarige loggar in på Kårspexets hemsida. Han/hon har uppskattningsvis 30 minuter till förfogande att jobba med sina uppgifter.

Görs ofta: bockar av betalade bokningar, skickar påminnelser till obetalda bokningar, tar bort gamla bokningar.

Görs mer sällan: kollar på utförlig statistik.

2.5 Antaganden och beroenden

Bokningssystemet som utvecklas för Kårspexet är beroende av datorkraft från webbservrar där mjukvaran körs. Mjukvaran och systemet i sin helhet ställer krav på yttre faktorer för att systemet skall bli användbart. De yttre faktorerna är framför allt bandbredd och serverprestanda.

Bandbredden talar om i vilken hastighet webbservern kan kommunicera med omvärlden. Omvärlden består av ett flertal användare som var och en kräver en viss del av den totala bandbredden då en användare är aktiv. Med andra ord beror behovet av bandbredd på hur många som använder systemet samtidigt.

Serverprestanda talar om hur många anrop till ett system en server kan hantera samtidigt. Varje aktiv användare kräver en del av den totala prestanda som finns tillgänglig. Behovet på serverprestanda beror precis som bandbredd på hur många som använder systemet vid en och samma tidpunkt.

Uppskattat möjliga samtida användare beror på de antaganden vi gör om systemet. Utifrån denna uppskattning samt dess användning vill vi bestämma hur mycket prestanda och bandbredd som systemet maximalt kan kräva.

2.5.1 Avgörande faktorer

- A. Hur många platser en föreställning har i medeltal.
- B. Hur många föreställningar som släpps för biljettköp åt gången.
- C. Hur stor del av platserna som säljs per tidsenhet då efterfrågan är som störst.
- D. Hur många anrop (sidladdningar) det krävs från bokningsgränssnittet för användaren till servern under en bokning i medeltal (första inladdningen utesluten).
- E. På vilken tid antalet anrop är fördelade vid en bokning (hur lång tid det tar att boka).
- F. Hur mycket trafik som överförs vid första inladdningen av bokningsgränssnittet för användaren.
- G. Hur mycket trafik som överförs vid ett anrop (första inladdningen utesluten) i medeltal.
- H. Hur många platser som bokas vid en bokning i medeltal.

2.5.2 Antaganden

- a. En föreställning har inte mer än 800 platser.
- b. Biljettsläpp görs inte för mer än 4 föreställningar i taget.

- c. Efterfrågan är maximalt 30% av platserna per timme.
- d. Bokningsgränssnittet för användaren behöver inte anropa servern mer än 10 gånger per bokning (första inladdningen ej inräknad).
- e. En bokning tar 4 minuter och bokningens anrop till servern är jämnt fördelat över tiden.
- f. Trafiken vid första inladdningen av bokningsgränssnittet för användaren är 100KiB.
- g. Trafiken för ett anrop (första inladdningen utesluten) är 30KiB stort i medeltal.
- h. Varje bokning omfattar 2 platser i medeltal.

2.5.3 Beräkningar

0,046 (anrop/sekund) för varje bokning under den tid det tar att boka: $(d+1)/(e*60)$

3200 bokningsbara platser vid varje biljettsläpp: $a*b$

0,27 (platser/sekund) som hanteras då efterfrågan är maximal: $((a*b*c)/(100*60*60))$

0,14 (bokningar/sekund) som hanteras då efterfrågan är maximal: $(a*b*c)/(100*60*60*h)$

1,47 (anrop/sekund) till servern då efterfrågan är maximal: $(a*b*c*(d+1))/(100*60*60*h)$

0,42 (MiBit/sekund) i trafik då efterfrågan är maximal: $(a*b*c*(d*g+f)*8)/(100*60*60*h*1024)$

2.5.4 Slutsats

Utifrån antagandena skall bandbredden minst vara **0,42** MiBit/sekund och webbservern måste klara av att hantera **1,47** anrop/sekund. Vad gäller bandbredden så motsvarar **0,42** MiBit/sekund en mindre del av en vanlig uppkoppling i hemmet. Det låga antalet **1,47** anrop/sekund mot bokningssystemet gör att prestanda från en vanlig persondator räcker till.

Antagandena om användandet av systemet har diskuterats med Kårspexet. De antaganden som gjorts är väl tilltagna gentemot Kårspexets uppfattning av användandet. Antagandena är tilltagna på ett sådant sätt att kraven för bandbredd och serverprestanda blir större. Med andra ord kommer Kårspexets användande av systemet sannolikt ha lägre krav på den befintliga hårdvaran än med angivna antagandena ovan.

2.6 Plattform

Biljettsystemet kommer använda flera externa system. Till att börja med kommer MVC-ramverket *Ruby on Rails* för webbapplikationer i Ruby användas. Det ger oss funktionalitet som underlättar webbutveckling, databashantering samt rendering av HTML.

Vi kommer även att använda databasmotorn *MySQL* för lagring av data. Databasen görs tillgänglig för systemet med hjälp av SQL. Dock kommer Rails att sköta mycket av den kommunikationen åt oss och i slutändan kommer databasen vara tillgänglig genom modeller i form av klasser i koden.

Systemet kommer dessutom vara beroende av Apache med modulen *Passenger* för att sköta inladdningen av applikationen och all HTTP-kommunikation mellan webbapplikationen och besökarna.

När vi implementerar kortbetalning i systemet kommer vi även vara beroende av ett externt system för hantering av korttransaktioner. Hur gränssnitt mot det systemet ser ut vet vi inte, eftersom inga beslut har tagits angående vilket system som ska användas.

3 Specifika krav

3.1 Funktionalitetskrav

3.1.1 Administration.

| | |
|-----------------------|---|
| Krav | UR1.1 Lägg till föreställning |
| Beskrivning | I bokningssystemets administratörens gränssnitt ska det gå att lägga till nya föreställningar som tillhör en omgång. En föreställning är en av de bokningsbara tillfällena för en omgång. |
| Motivering | Kårspexet gör nya omgångar med föreställningar varje år. Därför måste Kårspexet kunna lägga till föreställningar. |
| Behov | Standard. |
| Prioritet | Normal. |
| Stabilitet | Stabilt. |
| Källa | Kårspexet. |
| Verifierbarhet | Testa att lägga till en ny föreställning från administrationsvyn. |

| | |
|-----------------------|--|
| Krav | UR1.2 Redigera föreställningar |
| Beskrivning | Man ska kunna redigera tillagda föreställningar från administratörs vyn. |
| Motivering | Man kan behöva uppdatera med ny information som är korrekt. |
| Behov | Standard. |
| Prioritet | Normal. |
| Stabilitet | Stabilt. |
| Källa | Kårspexet. |
| Verifierbarhet | Testa att redigera föreställningen. |

| | |
|-----------------------|--|
| Krav | UR1.3 Enklare statistik |
| Beskrivning | Det skall vara möjligt att från administratörens och ekonomichefens gränssnitt kunna se enklare statistik från systemet. Detta inkluderar: totalt antal utgivna biljetter per föreställning, omgång och spelår, antal utgivna biljetter som är gratis/student/ordinarie per föreställning, omgång och spelår, antal bokade biljetter per föreställning, omgång och spelår. |
| Motivering | Underlättar arbetet för administratör och ekonomichef inför framtida planerianering av nya föreställningar, omgångar och spex. |
| Behov | Standard. |
| Prioritet | Låg. |
| Stabilitet | Stabilt. |
| Källa | Kårspexet. |
| Verifierbarhet | Kontrollera att man kan se den enklare statistiken från administratörens och ekonomichefens gränssnitt. |

| | |
|-----------------------|--|
| Krav | UR1.4 Omfattande statistik |
| Beskrivning | Det skall vara möjligt att från administratörens och ekonomichefens gräns-sning se mycket utförlig statistik från bokningssystemet. Detta inkluderar, men är ej begränsat till: enklare statistik inom vissa tidsintervall. För före-ställning innebär det möjlighet att välja vilken start/sluttid bokning samt utlämning av biljetter skedde. Det skall även vara möjligt att sortera antalet bokningar/utlämningar per dag och efter eventuell rabattklass. |
| Motivering | För bättre förståelse i bokningen vilket underlättar framtida planering och arbete av nya föreställningar, omgångar och spex. |
| Behov | Plus. |
| Prioritet | Normal. |
| Stabilitet | Stabilt. |
| Källa | Kårspexet. |
| Verifierbarhet | Man verifierar att statistiken stämmer jämfört med testdata. Sedan kon-trollerar man att det går att filtrera efter tid och eventuell rabattklass. |

| | |
|-----------------------|---|
| Krav | UR1.5 Filtrera bokningar |
| Beskrivning | Administratörsgränssnittet ska erbjuda möjligheten att filtrera bokningar efter betalnings- och placeringsstatus liksom bokningsnummer, föreställ-ning samt kontaktpersonens namn. |
| Motivering | För att underlätta administrationen av bokningar. |
| Behov | Standard. |
| Prioritet | Normal. |
| Stabilitet | Stabilt. |
| Källa | Kårspexet. |
| Verifierbarhet | Lista bokningar och verifiera att det finns ett formulär i vilket man kan välja hur bokningarna ska filtreras. Välj att filtrera på något sätt och verifiera att bokningslistan uppdateras korrekt. |

| | |
|-----------------------|--|
| Krav | UR1.6 Grafisk statistik |
| Beskrivning | Grafisk framställning av den statistik som finns tillgänglig. |
| Motivering | En grafisk representation gör det enklare att få en överblick över statistiken. |
| Behov | Deluxe. |
| Prioritet | Normal. |
| Stabilitet | Stabilt. |
| Källa | Kårspexet. |
| Verifierbarhet | Kontrollera att man från administratörens och ekonomichefens gränssnitt kan se den grafiska representationen av statistik. |

| | |
|-----------------------|---|
| Krav | UR1.7 Hantera utskick |
| Beskrivning | Administratör och ekonomichef ska kunna hantera de massutskick av epost som görs av systemet. |
| Motivering | Det är bra att kunna kontakta olika grupper av slutkunder, till exempel de som inte betalat sina biljetter. |
| Behov | Plus. |
| Prioritet | Låg. |
| Stabilitet | Stabilt. |
| Källa | Kårspexet. |
| Verifierbarhet | Man testar att göra utskick samt kontrollerar att de blev korrekta. |

| | |
|-----------------------|--|
| Krav | UR1.8 Sortera bokningar |
| Beskrivning | Administrationsgränsnittets listning av bokningar ska gå att sortera efter någon av de kolumner som visas. För att sortera efter en viss kolumn ska det gå att klicka på kolumnrubriken. |
| Motivering | För att underlätta administrationen av bokningar. |
| Behov | Plus. |
| Prioritet | Normal. |
| Stabilitet | Stabilt. |
| Källa | Nyx. |
| Verifierbarhet | Klicka på de olika kolumnrubrikerna i bokningslistan och verifiera att bokningarna sorteras i korrekt ordning. |

| | |
|-----------------------|--|
| Krav | UR1.9 Förhandsgranskning av utskick |
| Beskrivning | Det ska gå att förhandsgranska utskick innan de genomförs. Innan utskicket görs ska en förhandsgranskning i form av ett av (eventuellt) flera utskick visas. |
| Motivering | För att kunna kontrollera att utskick blir korrekta. |
| Behov | Plus. |
| Prioritet | Låg. |
| Stabilitet | Stabilt. |
| Källa | Nyx. |
| Verifierbarhet | Kontrollera att inget utskick görs utan att en förhandsgranskning först visas. |

| | |
|-----------------------|---|
| Krav | UR1.10 Ekonomigränssnitt |
| Beskrivning | Ett gränssnitt specifikt anpassat för att administrera ekonomin. Gränssnittets huvuduppgift är att registrera betalningar. Om detta inte hinns med kommer ekonomiansvariga administrera via administratörsgränssnittet. |
| Motivering | Ekonomichefen behöver ett gränssnitt som är anpassat för att registrera betalningar. |
| Behov | Plus. |
| Prioritet | Låg. |
| Stabilitet | Stabilt. |
| Källa | Kårspexet. |
| Verifierbarhet | Det går att logga in som ekonomichef och presenteras med ett gränssnitt som är anpassat specifikt för ekonomichefen. |

| | |
|-----------------------|--|
| Krav | UR1.11 Administratörsgränssnitt |
| Beskrivning | Administratören ska ha ett gränssnitt för att administrera hela systemet. |
| Motivering | Funktionalitet specifik för administratörer behöver presenteras i ett separat gränssnitt. |
| Behov | Standard. |
| Prioritet | Hög. |
| Stabilitet | Stabilt. |
| Källa | Kårspexet. |
| Verifierbarhet | Logga in som administratör och kontrollera att gränssnittet som presenteras inkluderar funktionalitet för administratörer. |

| | |
|-----------------------|--|
| Krav | UR1.12 Lägga till Teater |
| Beskrivning | Det ska gå att lägga till nya teatrar. De ska ha platser, sektioner och en salongsskiss, som visar hur platser och sektioner är placerade i teatern. I version Standard behöver det inte vara en smidig process. |
| Motivering | Eftersom Kårspexet kan komma att spela på nya teatrar behöver de kunna lägga till de i bokningssystemet. |
| Behov | Standard. |
| Prioritet | Hög. |
| Stabilitet | Stabilt. |
| Källa | Kårspexet. |
| Verifierbarhet | Möjlighet att lägga till en ny teater utöver existerande. Det ska gå att boka en biljett på en föreställning som går på den nya teatern. Bokningen ska kunna tilldelas platser i teatern. |

| | |
|-----------------------|--|
| Krav | UR1.13 Lägga till Omgång |
| Beskrivning | Man ska kunna lägga till nya omgångar i systemet från administratörens gränssnitt. En omgång är en mängd föreställningar som går på samma teater för samma priser. |
| Motivering | Kårspexet har många föreställningar på olika teatrar för olika priser. Omgångar behövs för att hantera detta. |
| Behov | Standard. |
| Prioritet | Hög. |
| Stabilitet | Stabilt. |
| Källa | Kårspexet. |
| Verifierbarhet | Det ska gå att från administratörens gränssnitt lägga till en ny omgång. Denna ska sedan synas i databasen. |

| | |
|-----------------------|---|
| Krav | UR1.14 Redigera omgång |
| Beskrivning | Det ska gå att ändra på samtliga egenskaper för en omgång via administratörsgränssnittet, dessa ändringar ska synas i de gränssnitt som berörs. |
| Motivering | Om något skulle bli fel vid uppläggning av en ny omgång. |
| Behov | Standard. |
| Prioritet | Normal. |
| Stabilitet | Stabilt. |
| Källa | Kårspexet. |
| Verifierbarhet | Efter att information har ändrats, ska man kunna se ändringen via bokningssidan. |

3.1.2 Bokning.

| | |
|-----------------------|---|
| Krav | UR2.1 Kontantbetalning från säljarvyn |
| Beskrivning | En säljare ska ha möjlighet att ta emot kontant betalning av en kund, och därefter säkerställa en bokning, med placering av biljetten till en plats. Säljaren ska kunna bekräfta betalningen och sedan ge ut biljetten till kunden. Det ska även vara möjligt för den kund som redan bokat att välja kontant betalning, och ta kontakt med en säljare för att betala sin biljett. Säljaren ska då kunna hitta den bokade biljetten, placera den, ta betalt, och sedan lämna ut den. |
| Motivering | Kårspexet vill kunna stå på offentliga platser och sälja och lämna ut biljetter till kunder. |
| Behov | Standard. |
| Prioritet | Normal. |
| Stabilitet | Stabilt. |
| Källa | Kårspexet. |
| Verifierbarhet | Kolla så att det går att göra en ny bokning via säljarens gränssnitt, och att den kan slutföras. Kolla så att säljaren kan hitta en obetald bokning, och registrera att bokningen är betald. |

| | |
|-----------------------|--|
| Krav | UR2.2 Säljare ska kunna lämna ut biljetter |
| Beskrivning | Säljare ska i sin vy kunna hitta en kund och se vilka biljetter han/hon ska lämna ut till honom/henne och därefter registrera bokningen som uthämtade. |
| Motivering | Säljaren behöver ha ett gränssnitt för att lämna ut biljetter. |
| Behov | Standard. |
| Prioritet | Normal. |
| Stabilitet | Stabilt. |
| Källa | Kårspexet. |
| Verifierbarhet | Söka upp en befintlig betalad biljett i säljarens interface och markera den som utlämnad. |

| | |
|-----------------------|--|
| Krav | UR2.3 Interaktiv översiktsbild |
| Beskrivning | Möjlighet för den som bokar att interagera med översiktsbilden. När man har muspekaren över en rad skall rätt sektion markeras i översiktsbilden och sektionstabellen. |
| Motivering | Det skulle göra det lättare för besökare att förstå vilken sektion de ska boka platser till för att hamna på ett visst ställe i salongen. |
| Behov | Plus. |
| Prioritet | Normal. |
| Stabilitet | Stabilt. |
| Källa | Nyx. |
| Verifierbarhet | Vid bokningssteget där man väljer sektioner skall det gå att interagera med översiktsbilden. |

| | |
|-----------------------|---|
| Krav | UR2.4 Avbokning |
| Beskrivning | En kund skall kunna avboka sin bokning om den inte är markerad som betalad. Det sker genom en länk i bekräftelsemailet för bokningen. |
| Motivering | Användare ska kunna avboka biljetter de inte önskar betala eller hämta ut. |
| Behov | Standard. |
| Prioritet | Låg. |
| Stabilitet | Stabilt. |
| Källa | Kårspexet. |
| Verifierbarhet | I bekräftelsemailet finns det en länk till en sida där man kan ta bort sin bokning. Efter att man avbokat ska bokningen vara borta ur systemet. |

| | |
|-----------------------|---|
| Krav | UR2.5 Bokning Administratör |
| Beskrivning | Administratören skall ha möjlighet att göra bokningar via administratörsinterfacet. Administratören har full tillgång till rabattklasserna, även gratis, och kan placera ut de valda platserna direkt samt sätta status som betald. En gratisbokning är detsamma som att registrera en gratisbiljett. |
| Motivering | Kravet behövs för att kårspexet ska kunna ge bort gratisbiljetter och ha koll på att det är just gratisbiljetter. |
| Behov | Standard. |
| Prioritet | Normal. |
| Stabilitet | Stabilt. |
| Källa | Kårspexet. |
| Verifierbarhet | Administratören kan boka och registrera gratisbiljetter. |

| | |
|-----------------------|--|
| Krav | UR2.6 Boka |
| Beskrivning | Det ska gå att som kund genomföra en bokning med valfritt antal biljetter för en vald föreställning. Detta görs i en flerstegsprocess där kunden först väljer föreställning, sedan vilka platser som ska bokas och slutligen anger sina kontaktuppgifter samt hur betalning ska ske. När bokningen är genomförd presenteras en sammanfattning av den för kunden. |
| Motivering | Huvudsyftet med systemet. |
| Behov | Standard. |
| Prioritet | Hög. |
| Stabilitet | Stabilt. |
| Källa | Kårspexet. |
| Verifierbarhet | Boka en biljett och verifiera sedan att den skapats i administrationsgränssnittet. |

| | |
|-----------------------|---|
| Krav | UR2.7 Färgkodning |
| Beskrivning | Den bild som ger kunden en översikt av platsfördelningen på den valda föreställningen ska ha en färgskala, som anger till vilken grad sektionerna är lediga. För att underlätta för färgblinda bör lämpliga färger väljas. Färgerna ändras dynamiskt allt eftersom fler bokningar görs. |
| Motivering | Ger snabb överblick för kunden i början av bokningen. |
| Behov | Plus. |
| Prioritet | Normal. |
| Stabilitet | Stabilt. |
| Källa | Nyx. |
| Verifierbarhet | Kolla så att bilden har korrekta startfärger när det inte finns några bokningar i systemet. Kolla gradvis efter övergångar mellan färger, allt eftersom fler bokningar görs. |

| | |
|-----------------------|---|
| Krav | UR2.8 Kortköp |
| Beskrivning | Möjlighet för kund att betala med kort vid bokning. |
| Motivering | Kårspexet vill ha kortbetalning som ett smidigt betalsätt för slutanvändaren. |
| Behov | Deluxe. |
| Prioritet | Normal. |
| Stabilitet | Stabilt. |
| Källa | Kårspexet. |
| Verifierbarhet | Välja att betala med kort vid slutförande av en bokning. |

| | |
|-----------------------|---|
| Krav | UR2.9 Ändra betalningsstatus |
| Beskrivning | Det skall vara möjligt att från administratören och ekonomichefens gränssnitt att ändra betalningsstatus för en bokning från obetald till betald eller tvärtom. |
| Motivering | Det är viktigt för Kårspexet att de kan registrera inkomna betalningar. |
| Behov | Standard. |
| Prioritet | Normal. |
| Stabilitet | Stabilt. |
| Källa | Kårspexet. |
| Verifierbarhet | Från administratörens och ekonomichefens gränssnitt kontrolleras att en bokning kan ändra betalningsstatus. |

| | |
|-----------------------|---|
| Krav | UR2.10 Bekräftelsemeddelande |
| Beskrivning | Efter att en kund genomfört en bokning ska en bekräftelse skickas via epost till den epostadress som kunden angett i bokningen. Denna bekräftelse ska inkludera nödvändig information om bokningen, såsom: betalningsinformation, bokningsnummer, aktuella datum och tider samt en fungerande länk för avbokning. |
| Motivering | Det är viktigt att kunden får en bekräftelse av bokningen med aktuell information. |
| Behov | Standard. |
| Prioritet | Normal. |
| Stabilitet | Stabilt. |
| Källa | Kårspexet. |
| Verifierbarhet | Genomför en bokning och verifiera att ett korrekt bekräftelsemeddelande har skickats ut till rätt epostadress. |

| | |
|-----------------------|---|
| Krav | UR2.11 Säljargränssnitt |
| Beskrivning | Säljarens gränssnitt ska vara enkelt att lära sig och svårt att göra felbokningar i. Från gränssnittet ska det gå att göra kontantköp och registrera uthämtningar av biljetter. |
| Motivering | Säljare behöver ett enklare gränssnitt för minska risken för fel. |
| Behov | Standard. |
| Prioritet | Normal. |
| Stabilitet | Stabilt. |
| Källa | Kårspexet. |
| Verifierbarhet | Det ska gå att logga in som säljare och utföra kontantköp samt registrering av uthämtade biljetter. |

| | |
|-----------------------|---|
| Krav | UR2.12 Studentbiljetter |
| Beskrivning | Vid en bokning skall det gå att boka studentbiljetter som är en typ av specialbiljett. Det som skiljer en studentbiljett från en ordinarie biljett är priset. |
| Motivering | Kårspexet vill att det ska gå att boka rabatterade studentbiljetter. |
| Behov | Standard. |
| Prioritet | Normal. |
| Stabilitet | Stabilt. |
| Källa | Kårspexet. |
| Verifierbarhet | Genomföra en bokning som inkluderar minst en studentbiljett och verifiera att studentbiljetten syns i administrationsgränssnittet. |

| | |
|-----------------------|--|
| Krav | UR2.13 Utplacering av platser för bokningar |
| Beskrivning | I administratörsgränssnittet ska det gå att tilldela stolar till bokningar. I detta gränssnitt ska det gå att se vilka stolar som är upptagna och vilka som finns tillgängliga för utplacering för den aktuella föreställningen. |
| Motivering | Kårspexet behöver ha möjlighet att manuellt placera ut bokningar. |
| Behov | Standard. |
| Prioritet | Hög. |
| Stabilitet | Stabilt. |
| Källa | Kårspexet. |
| Verifierbarhet | Via administratörsgränssnittet ska man kunna tilldela platser för en ännu oplacerad bokning. |

| | |
|-----------------------|--|
| Krav | UR2.14 Redigering av bokningar |
| Beskrivning | Administratörer ska kunna redigera befintliga bokningar. Detta inkluderar att ändra betalningsstatus (hur mycket som betalats in), om biljetterna är uthämtade samt övriga egenskaper, dock inte nödvändigtvis alla. |
| Motivering | Det måste gå att uppdatera bokningsstatus. |
| Behov | Standard. |
| Prioritet | Normal. |
| Stabilitet | Stabilt. |
| Källa | Kårspexet. |
| Verifierbarhet | Välj att redigera en enskild bokning, verifiera att egenskaper hos bokningen går att ändra och att de sparas korrekt. |

3.1.3 Diverse.

| | |
|-----------------------|---|
| Krav | UR3.1 Möjlighet att navigera i systemen |
| Beskrivning | I varje vy (kund, säljare, ekonomichef, administatör), ska alla tillhörande funktioner kunna nås via länkar på webbsidorna. |
| Motivering | Användaren måste kunna komma åt all funktionalitet. |
| Behov | Standard. |
| Prioritet | Normal. |
| Stabilitet | Stabilt. |
| Källa | Kårspexet. |
| Verifierbarhet | Man testar att alla länkarna finns, och att de inte är trasiga. |

| | |
|-----------------------|---|
| Krav | UR3.2 Tidsmätning |
| Beskrivning | För varje anrop till systemet, ska tiden det tar att svara mätas. Sedan ska tiden skrivas till en logg-fil. |
| Motivering | För att verifiera att kravet om svarstid har uppfyllts. |
| Behov | Standard. |
| Prioritet | Låg. |
| Stabilitet | Stabilt. |
| Källa | Kalle Arvidsson, Johan Stjernberg. |
| Verifierbarhet | Kontrollera att tiden står i loggen. |

3.1.4 Loginsystem.

| Krav | UR4.1 Loginsystem |
|-----------------------|---|
| Beskrivning | För att komma åt säljarnas, ekonomichefens och administratörens gränssnitt och funktioner måste användare identifiera sig med användarnamn samt tillhörande lösenord. Det ska alltså finnas någon form av system som hanterar användare och inloggningar. |
| Motivering | För att uppfylla kravet säkerhet, se 3.2.2. |
| Behov | Standard. |
| Prioritet | Normal. |
| Stabilitet | Stabilt. |
| Källa | Kårspexet. |
| Verifierbarhet | Testat genom att försöka komma åt gränssnitten, kontrollera att lösenord efterfrågas och att det rätta lösenordet ger tillgång till gränssnitten. |

| Krav | UR4.2 Kontohantering |
|-----------------------|--|
| Beskrivning | Administratörer ska kunna hantera de olika kontona och byta lösenord på dem. De ska även kunna sätta tidsbegränsade lösenord för användarna. |
| Motivering | Administratörer måste kunna hantera kontona. |
| Behov | Standard. |
| Prioritet | Normal. |
| Stabilitet | Instabilt. Osäkert hur kontohanteringen ska fungera i dagsläget. |
| Källa | Kårspexet. |
| Verifierbarhet | Kontrollera att det går att gå in som administratör och redigera/skapa/hantera konton. |

| Krav | UR4.3 Lösenordsgenerator |
|-----------------------|--|
| Beskrivning | När administratören skapar nya lösenord för säljare ska det finnas möjlighet att generera slumpmässiga lösenord. |
| Motivering | För att på ett enkelt sätt ändra säljarens lösenord till ett säkert lösenord. |
| Behov | Deluxe. |
| Prioritet | Låg. |
| Stabilitet | Stabilt. |
| Källa | Kårspexet. |
| Verifierbarhet | Det går att generera nytt lösenord när man administrerar säljarkontot. |

3.2 Begränsande krav

3.2.1 Prestanda

| | |
|-----------------------|--|
| Krav | UR5.1 Serverbelastning |
| Beskrivning | Systemet ska utan märkbara problem hantera minst åtta typiska användare samtidigt. Hur väl kravet uppfylls beror mycket på serverns prestanda. |
| Motivering | Det kommer förekomma fall då flera använder systemet samtidigt. Se Antaganden och beroenden för uppskattningar. |
| Behov | Standard |
| Prioritet | Låg |
| Stabilitet | Stabilt |
| Källa | Johan Stjernberg, Kalle Arvidsson |
| Verifierbarhet | Testkörning av systemet på Kårspexets server med åtta eller fler användare. |

| | |
|-----------------------|---|
| Krav | UR5.2 Svarstid |
| Beskrivning | Systemet får inte ta för lång tid på sig att svara på användarens anrop. Vi kan dock inte ansvara för fördröjningar i nätverket mellan systemet och användaren. Olika operationer kan ha olika långa maximala svarstider. Vid alla operationer i alla gränssnitt som enbart gäller en enskild bokning ska systemet svara på max 1 sekund. |
| Motivering | Svarstiden är viktig för användarens upplevelse av systemet och vid väldigt långa svarstider försämras systemets användbarhet. |
| Behov | Standard |
| Prioritet | Låg |
| Stabilitet | Stabilt |
| Källa | Nyx |
| Verifierbarhet | Tiderna mäts och skrivs till en logg. Kontrollera att tiderna är tillräckligt små. |

3.2.2 Säkerhet

| | |
|-----------------------|---|
| Krav | UR6.1 Autentisering |
| Beskrivning | De funktioner som hör till säljarna, ekonomichefen eller administratören, ska bara kunna användas om man angett ett lösenord. |
| Motivering | Bara Kårspexets personal ska kunna använda dessa funktioner. |
| Behov | Standard |
| Prioritet | Låg |
| Stabilitet | Stabilt |
| Källa | Kårspexet |
| Verifierbarhet | Verifiera kravet Inloggningssystem |

| | |
|-----------------------|--|
| Krav | UR6.2 Datasäkerhet |
| Beskrivning | Data och lösenord i systemet ska inte gå att komma åt av obehöriga |
| Motivering | Användare av systemet ska kunna lita på att data och lösenord är trygga i systemet |
| Behov | Standard |
| Prioritet | Låg |
| Stabilitet | Stabilt |
| Källa | Kårspexet |
| Verifierbarhet | Verifiera att det inte går att komma åt data i systemet oaktoriserat |

3.2.3 Miljö

| | |
|-----------------------|---|
| Krav | UR7.1 Webb |
| Beskrivning | Kunder såväl som Kårspexets personal ska kunna använda bokningssystemet genom webbgränssnitt. |
| Motivering | Smidigast eftersom det innebär maximal tillgänglighet. |
| Behov | Standard |
| Prioritet | Hög |
| Stabilitet | Stabilt |
| Källa | Kårspexet |
| Verifierbarhet | Provkörning av systemet via webbläsare. |

| | |
|-----------------------|--|
| Krav | UR7.2 Rails |
| Beskrivning | Ruby on Rails är ett ramverk för utveckling av webbapplikationer. Bokningssystemet ska huvudsakligen vara byggt med detta ramverk. |
| Motivering | Gruppen tycker det verkar passande för projektet och vill arbeta i ramverket. |
| Behov | Standard |
| Prioritet | Hög |
| Stabilitet | Stabilt |
| Källa | Nyx |
| Verifierbarhet | Undersökning av serverns konfiguration samt källkoden. |

| | |
|-----------------------|---|
| Krav | UR7.3 Webbläsarkompatibilitet |
| Beskrivning | Det ska gå använda bokningssystemets alla funktioner med följande webbläsare: <i>Firefox 3</i> , <i>Internet Explorer 8</i> . |
| Motivering | Dessa webbläsare är stora på marknaden just nu och bör stödas av vårt system. Vi refererar till punkt 1.4 för statistik på webbläsaranvändande. |
| Behov | Standard |
| Prioritet | Låg |
| Stabilitet | Instabilt |
| Källa | Nyx |
| Verifierbarhet | Provkörning av systemet i dessa webbläsare. |

| | |
|-----------------------|---|
| Krav | UR7.4 Visuell webbläsarkompatibilitet |
| Beskrivning | Det ska inte vara skillnad på hur systemets webbsidor ser ut i de webbläsare som nämndes i kravet webbläsarkompatibilitet. Mindre avvikelser får förekomma, till exempel i teckensnitt. |
| Motivering | Om gränssnitten inte ser ut som de är tänkta att se ut, är de troligen svårare att använda. Se även motiveringen för webbläsarkompatibilitet. |
| Behov | Plus |
| Prioritet | Låg |
| Stabilitet | Instabilt |
| Källa | Nyx |
| Verifierbarhet | Provkörning av systemet i de olika webbläsarna. |

| | |
|-----------------------|--|
| Krav | UR7.5 Internet Explorer 7 |
| Beskrivning | Kravet webbläsarkompatibilitet uppfylls även för webbläsaren <i>Internet Explorer 7</i> . |
| Motivering | <i>Internet Explorer 7</i> är en webbläsare som används, men som skiljer sig från de andra webbläsarna så att extra arbete krävs för att sidorna ska visas korrekt. Därför finns detta krav inte i Standard. |
| Behov | Plus |
| Prioritet | Låg |
| Stabilitet | Instabilt |
| Källa | Nyx |
| Verifierbarhet | Provkörning av systemet med <i>Internet Explorer 7</i> . |

3.2.4 Användbarhet

| | |
|-----------------------|---|
| Krav | UR8.1 Bokningstid |
| Beskrivning | En typisk kund ska kunna genomföra sin första bokning på mindre än fem minuter. |
| Motivering | Det ska vara enkelt och smidigt att använda systemet. |
| Behov | Standard |
| Prioritet | Hög |
| Stabilitet | Stabilt |
| Källa | Johan Stjernberg, Kalle Arvidsson |
| Verifierbarhet | En urvalsgrupp som inte tidigare använt systemet provbokar under tidsmätning. |

| | |
|-----------------------|---|
| Krav | UR8.2 Inlärningsstid |
| Beskrivning | En typisk KTH-student ska, på en dag, kunna sätta sig in i administrationsgränssnittets huvudsakliga funktioner. |
| Motivering | Det ska gå smidigt för Kårspexets personal att använda systemet. |
| Behov | Standard |
| Prioritet | Medel |
| Stabilitet | Stabilt |
| Källa | Johan Stjernberg, Kalle Arvidsson |
| Verifierbarhet | Undersökning av hur lång tid det tar för Kårspexets personal eller andra KTH-studenter att sätta sig in i systemet. |

| | |
|-----------------------|--|
| Krav | UR8.3 Introduktion |
| Beskrivning | Vid leverans av produkt ska en introduktion till systemet ges vid ett tillfälle. Vi utlovar ingen vidare kundhjälp efter leverans. |
| Motivering | Det är nödvändigt att ge instruktioner till Kårspexet, dock kan inte gratis hjälp utlovas efter leverans. |
| Behov | Standard |
| Prioritet | Låg |
| Stabilitet | Stabilt |
| Källa | Johan Stjernberg, Kalle Arvidsson |
| Verifierbarhet | Kårspexet kan ombedas intyga att de fått instruktioner för systemet. |

| | |
|-----------------------|---|
| Krav | UR8.4 Förbättring |
| Beskrivning | Kårspexets personal såväl som deras kunder ska ha ett bättre bokningssystem än det tidigare. |
| Motivering | Om inte vårt bokningssystem är bättre än det befintliga har vi misslyckats med vårt uppdrag. |
| Behov | Standard |
| Prioritet | Hög |
| Stabilitet | Stabilt |
| Källa | Kårspexet |
| Verifierbarhet | Kårspexets personal ombedes lämna en muntlig eller skriftlig jämförelse av systemen, med särskilt fokus på användbarhet och effektivitet. |

3.2.5 Externa system

| | |
|-----------------------|--|
| Krav | UR9.1 MySQL |
| Beskrivning | Bokningssystemet ska använda databashanteraren MySQL. |
| Motivering | Gruppen vill använda MySQL och Kårspexet har samtyckt. |
| Behov | Standard |
| Prioritet | Hög |
| Stabilitet | Stabilt |
| Källa | Nyx |
| Verifierbarhet | Uppvisning av databas eller kontroll av källkod. |

| | |
|-----------------------|---|
| Krav | UR9.2 Apache |
| Beskrivning | Bokningssystemet ska använda webbservern Apache. |
| Motivering | Gruppen vill att webbplatsen ska använda Apache. Apache har stöd för Ruby on Rails. |
| Behov | Standard |
| Prioritet | Hög |
| Stabilitet | Stabilt |
| Källa | Nyx |
| Verifierbarhet | Visa att Apache körs på servern. |

| | |
|-----------------------|---|
| Krav | UR9.3 Kortbetalningssystem |
| Beskrivning | Bokningssystemet ska använda sig av ett externt system för kortbetalning. |
| Motivering | Vi kan inte ta på oss att hantera säkra kortbetalningar själva, ett externt system behövs. |
| Behov | Deluxe |
| Prioritet | Medel |
| Stabilitet | Stabilt |
| Källa | Kårspexet |
| Verifierbarhet | Om kortbetalning fungerar används ett externt system. För verifiering av att kortbetalning fungerar, se kravet kortköp. |

A Databasmodell

Varje box motsvarar en tabell i databasen, varje rad i en box motsvarar fält i tabellen. En rad i en box kan representera flera fält i en tabell, t.ex. kontaktuppgifter som skulle motsvara fälten Namn, Telefon, Adress osv. Pilarna indikerar att det finns en referens mellan två tabeller, 'FK' visar vilket fält som refererar till den andra tabellen. En fetmarkerad rad i en box innebär att fältet i tabellen måste ha ett värde. 'PK' innebär att ett eller flera fält identifierar en rad i tabellen, alltså gör den unik.