



Bokningssystem för Kårspexet

User Requirements Document (URD)

Tarandi, Andreas – taran@kth.se
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

Nyx

Version 0.3
December 6, 2010

Abstract

This document aims to describe the user requirements for Kårspexets online booking system, developed by Nyx. It contains the systems capabilities, its 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.

Innehåll

1	Introduktion	3
1.1	Syfte	3
1.2	Mjukvarans omfattning	3
1.3	Definitioner, akronymer och förkortningar	3
1.4	Källor	6
1.5	Dokumentöversikt	7
2	Allmän beskrivning	8
2.1	Produktperspektiv	8
2.2	Allmän funktionalitet	8
2.3	Allmänna begränsningar	9
2.4	Användarbeskrivning	10
2.5	Antaganden och beroenden	12
2.6	Plattform	14
3	Specifika krav	15
3.1	Begränsande krav	23

Ändringslogg

Version	Ändringar
0.1	Första sammanställd version av dokumentet.
0.2	Texter under bearbetning men i stort sätt färdigkomponerade. Texter i dokumentet numreras numera med sitt revisionsnummer i dokumenthanteringssystemet för spårbarhet. Kravdatabas innehåller kundens funktionalitetskrav, men beskrivande texter saknas till vissa av dem.
0.3	Texter ska vara färdigskrivna. Version som skickas för granskning av URD:n. Revisionsnummer flyttade till egen punkt.

Dokumentversioner

Dokumentet har genererats från följande deldokument.

User_Requirements_Document version 1.

URDabstract version 3.

URDÄndringslogg version 5.

URDIntroduktion version 4.

URDIntroduktionSyfte version 5.

URDIntroduktionMjukvarans__omfattning version 3.

URDIntroduktionDefinitioner__akronymer__och__förkortningar version 56.

URDIntroduktionKällor version 12.

URDIntroduktionDokumentöversikt version 5.

URDAllmän__beskrivning version 2.

URDAllmän__beskrivningProduktperspektiv version 24.

URDAllmän__beskrivningGeneral__Capabilities version 7.

URDAllmän__beskrivningAllmänna__begränsningar version 37.

URDAllmän__beskrivningAnvändarbeskrivning version 25.

URDAllmän__beskrivningAntaganden__och__beroenden version 21.

URDAllmän__beskrivningOperational__environment version 7.

URDSpecifika__krav version 2.

URDSpecifika__kravKravbegränsning version 36.

1 Introduktion

1.1 Syfte

Det här dokumentet innehåller en detaljerad sammanfattning av produktens krav och dess motiveringar inom vårt projekt. Dokumentets tänkta läsare är tekniskt datakunniga personer samt utvecklare.

1.2 Mjukvarans omfattning

Produkten består av ett webbaserat biljettbokningssystem med ett enkelt användargränssnitt och ett kraftfullt administrationsverktyg. Administrationsgränssnittet utgörs av tre delar; ett för säljare, ett för ekonomiansvariga och ett för administratörer.

1.3 Definitioner, akronymer och förkortningar

Behöver alla akronym referenser? Någon som vet? //André

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 webbrowser. Är gratis att använda. [1.3.1]

Apache Software Foundation Organisation som stödjer ett antal öppen-källkods-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.

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

Epostklient Även kallat Epostprogram. Datorprogram för att hantera/läsa/skicka epost.

Epostprogram Se Epostklient.

FK Foreign Key, främmande nyckel, ett värde i en databas som refererar till en rad i en annan

tabell. (Används denna förkortning på enbart i vårt diagram?)

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. Tidskillnaden 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

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

MVC framework *Mjukvarubibliotek designade efter MVC-konceptet.*

MySQL *En typ av relationsdatabas baserad på SQL-standarden. Ett relationsdatabas hanteringsystem 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.*

PK *Primary Key, primärnyckel, det fält (i en databas) som används för att identifiera en rad i en tabell. (Används denna förkortning enbart i diagrammet?)*

Processor *Den komponent i en dator som genomför beräkningar och andra 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 *Humoristisk studentamatörteaterföreställning, förkortning av spektakel.*

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 URD 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).

victor: *borde kanske ange referensnumret till vänster om titeln för att göra det enkelt att direkt hitta en viss referens? dvs [1.3.1] Apache HTTP Server. Dessa rubriker bör för övrigt INTE vara latex-numrerade.*

1.4.1 Apache HTTP Server

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

Hänvisning till källan görs från referenserna: [1.3.1].

1.4.2 Apache Software Foundation

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

Hänvisning till källan görs från referenserna: [1.3.2].

1.4.3 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].

1.4.4 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].

1.4.5 Undersökning av prestanda för Rails

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

Hänvisning till källan görs från referenserna: [2.3.2].

1.4.6 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]

1.5 Dokumentöversikt

Systemet som Nyx utvecklar åt Kårspexet ersätter ett gammalt system, detta behandlas sektion 2.1. Sektion 2.2 presenterar systemets användare kort, och går igenom användarscenarion för dessa. Projektets begränsningar går igenom 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, medan de mer operationella kraven beskrivs i 2.6.

I sektion 3 specificeras alla features 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 så de kan vidareutveckla systemet vid behov. Lösningen måste vara enkel så att Kårspexet slipper lägga mer tid än nödvändigt på administrationen, vilket ger dem mer tid att fokusera på andra aktiviteter som marknadsföring och 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 resten av hemsidan. Systemet körs på en extern server, som Kårspexet ej har tillgång till. Det system som vi kommer att konstruera skall ha olika vyer för kunder, administratörer och säljare på Kårspexets hemsida. Varje vy kommer att anpassas för 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 som uppmanar kunden att hämta ut sina biljetter.

2.2.4 Lämna ut bokade biljetter

När kunden kommer till säljaren ska denna verifiera att biljetterna är redo att hämtas samt registrera att biljetterna har hämtats. Gränssnittet för säljaren ska vara lätt att lära sig eftersom säljarna ofta byts ut.

2.2.5 Sälja biljetter direkt

Säljaren ska även kunna sälja biljetter kontant. Då är det säljaren som väljer vilka stolar som 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 kunder.

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 den 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 som en bilaga.

2.3.2 Resurser

Vi kommer vara begränsade i vilka och hur många funktioner vi kommer kunna implementera främst då vi totalt är fem programmerare. På ett fåtal veckor ska vi hinna implementera fyra gränssnitt mot bokningssystemets användare. Gränssnitten kommer behöva testas men då vi har nästan lika många testare som vi har utvecklare kommer inte detta utgöra ett hinder för tidsplanen. Det som kommer vara vårt största hinder under utvecklingen är administratörsgränssnittet då det är där de flesta funktionerna kommer finnas och även de mest avancerade.

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 tidsramen vi har 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. Administrationsgränssnittet 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 att köra vår webbapplikation. Vi har ingen kontroll över dess hårdvara men har verifierat att operativsystemet som körs på den är kompatibelt med Apache, Ruby med Rails [2.3.1] samt MySQL.

Applikationen kommer inte att 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 kabelanslutningar.

Systemet kommer enligt våra uppskattningar exponeras för upp till tio simultana anslutningar. Rails under Apache kommer i detta fall att, under godtycklig tidpunkt, använda uppskattningsvis 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 samt en processor med klockfrekvensen 1 GHz eller högre. Kårspexets server har en processor med klockfrekvensen 2,6 GHz samt 1 Gb systemminne, vilket alltså bör vara tillräckligt.

2.4 Användarbeskrivning

Produkten kommer att ha 4 olika användare: Kund, säljare, administratör och ekonomiansvarig.

2.4.1 Kund

Teknisk bakgrund för kund Kunder är uppdelad i två läger; de som pluggar på en teknisk högskola eller ett universitet, och släktingar eller bekanta till Kårspexets medlemmar som inte är associerade med en teknisk högskola eller ett universitet.

h5. Tekniska högskolestudenter

Använder datorer dagligen, antingen som del av sin utbildning och/eller för privat bruk. Van användare av emailklienter, och anpassar sig i behaglig takt till nya sidlayouter eller program.

h5. Släktingar och bekanta

Den tekniska kunnigheten varierar stort inom denna grupp, från datorvana tonåringar till pensionärer som aldrig rört en dator. Kan använda emailklienter till viss mån, och tar lång tid på sig att anpassa sig till nya sidlayouter eller program.

Typsscenario för kunden Kunden går in på Kårspexets hemsida, och trycker på boka biljett. Kunden behöver en snabb och genomförlig överblick av vilka föreställningar som finns, och hur många platser som finns i respektive sektioner. Kunden får snabbt återkoppling på sina val och går igenom flera steg av bokningen; val av föreställning, val av sektion, betalningsuppgifter, och bokningsbekräftelse. Kunden får sedan ett mail från Kårspexet om betalmedel, och betalar sin biljett. Kunden får sedan ett mail från Kårspexet om att hans/hennes biljett finns att hämta hos ombud.

2.4.2 Säljare

Teknisk bakgrund för säljare Säljare är medlemmer i Kårspexet, är därmed med stor sannolikhet kårmedlemmar vid en teknisk högskola. De är först och främst aktiva med kårspexets arrangemang, och säljare i andra hand. Det är därför viktigt att säljare snabbt lär sig använda bokningssystemet, då de inte ska kräva någon tidigare utbildning i systemet.

Typsscenario för säljare Säljaren står i Kårhuset eller på utsatt plats, och loggar in på Kårspexets hemsida. Antingen så säljs biljetter på plats, med kunder som står i kö; eller så kommer en kund som bokad sin biljett på hemsidan och valt att betala kontant. I båda fallen så 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 mail från Kårspexet om att deras biljett finns att hämta.

2.4.3 Administratör och ekonomiansvarig

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

h5. Mindre teknisk kunrig

Organisatören kan ha en bakgrund i matematik, biologi, kemi, eller liknande ämne, där programmering 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 tekniskt kunnige att använda nya program eller anpassa sig till nya sidlayouter.

h5. Mer teknisk kunrig

Organisatören kan ha en bakgrund i datalogi, teknisk fysik, eller liknande ämne, där programmering har varit del av utbildningen. Organisatören förstår objektorientering, och kanske html eller webbprogrammering. En sida söks igenom systematiskt, och organisatören lär sig snabbt nya sidlayouter eller program.

Typscenario för administratören Administratören loggar in på Kårspexets hemsida. Han/hon har uppskattningsvis 15 minuter till 2 timmar till förfogande att jobba med sina uppgifter.

Gör ofta: placera ut betalda bokningar, planera föreställningar och ta hand om specialbokningar, t.ex. handikappsbokningar eller stora företagsbokningar.

Gör mer sällan: lägga till en ny teater, lägga till en ny omgång, skicka massutskick till b.la. kundbokningar, kolla på statistik.

Typscenario för ekonomiansvarige Ekonomiansvarige loggar in på Kårspexets hemsida. Han/hon har uppskattningsvis 15 minuter till 2 timmar till förfogande att jobba med sina uppgifter.

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

Gör mer sällan: Kolla 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 bandbreddsuppkoppling och serverprestanda.

Bandbreddsuppkopplingen 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 bandbreddsuppkoppling på hur många som använder systemet samtidigt.

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

Antalet simultiga användare beror på en rad olika antaganden om systemet och användandet av systemet. Utifrån antagandena vill vi bestämma hur mycket prestanda och bandbredd som

systemet maximalt kan kräva. De avgörande antagandena berör:

- 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.

Vi antar att:

- a. att en föreställning inte har mer än 800 platser.
- b. att biljettsläpp inte görs för mer än 4 föreställningar i taget.
- c. att efterfrågan är maximalt 30
- d. att bokningsgränssnittet för användaren inte behöver anropa servern mer än 10 gånger per bokning (första inladdningen ej inräknad).
- e. att en bokning tar 4 minuter och att bokningens anrop till servern är jämt fördelat över tiden.
- f. att trafiken vid första inladdningen av bokningsgränssnittet för användaren är 100KiB.
- g. att trafiken för ett anrop (första inladdningen utesluten) är 30KiB stort i medeltal.
- h. att varje bokning omfattar 2 platser i medeltal.

Våra antaganden ger:

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 $((a*b*c*(d*g+f)*8)/(100*60*60*h*1024))$.

Utifrån antagandena så skall bandbreddsuppkopplingen minst vara 0,42 MiBit/sekund och

webbservern måste klara av att hantera 1,47 anrop/sekund. Vad gäller bandbreddsuppkopplingen 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 bandbreddsuppkoppling och serverprestanda blir större. Med andra ord kommer Kårspexets användande av systemet 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 ett MVC-framework för webbapplikationer i Ruby on Rails att användas. Det ger oss funktionalitet som gör webbutveckling smidig, databashantering, rendering av HTML och mycket hjälpfunktioner som underlättar webbutveckling.

Vi kommer även att använda databasmotorn MySQL för datalagring. 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 2 med modulen Passenger för att sköta inladdningen av applikationen och all HTTP-kommunikation i produktion. Gränssnitt mot Apache är i form av konfigurationsfiler på servern.

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

3 Specifika krav

Kravnummer	1
Krav	Loginsystem
Beskrivning	För att komma åt säljarnas, ekonomichefens och administratörens gränssnitt och funktioner måste användare identifiera sig via ett användarnamn med tillhörande lösenord. Det ska alltså finnas någon form av säkerhetssystem som hanterar användare och inloggningar.
Motivering	För att uppfylla kravet säkerhet.
Behov	Standard
Prioritet	High
Källa	Kårspexet
Verifierbarhet	Försök komma åt gränssnitten, kontrollera att lösenord efterfrågas och att det rätta lösenordet ger tillgång till gränssnitten.

Kravnummer	2
Krav	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	High
Källa	Kårspexet
Verifierbarhet	Boka en biljett och verifiera sedan att den skapats i administrationsgränssnittet.

Kravnummer	3
Krav	Bekräftelsemail
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 Kårspexet får en bekräftelse av bokningen med aktuell information.
Behov	Standard
Prioritet	Normal
Källa	Kårspexet
Verifierbarhet	Genomför en bokning och verifiera att ett korrekt bekräftelsemail har skickats ut till rätt epostadress.

Kravnummer	4
Krav	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
Källa	Kårspexet
Verifierbarhet	Genomföra en bokning som inkluderar minst en studentbiljett och verifiera att studentbiljetten syns i administrationsgränssnittet.

Kravnummer	5
Krav	Utplacering av platser för bokningar
Beskrivning	I administrationsgränssnittet ska det gå att placera ut bokningar på stolsnivå. I detta gränssnitt ska det gå att se vilka platser 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	High
Källa	Kårspexet
Verifierbarhet	Vi administratörsgränssnittet ska man kunna tilldela platser för en ännu oplacerad bokning.

Kravnummer	6
Krav	Redigering av bokningar
Beskrivning	Administratörer ska kunna redigera befintliga bokningar. Detta inkluderar att ändra betalningsstatus (hur mycket som betalats in), om biljetterna är uthämtade eller inte samt flera andra egenskaper hos bokningarna, dock inte nödvändigtvis alla.
Motivering	Det måste gå att uppdatera bokningsstatus
Behov	Standard
Prioritet	Normal
Källa	Kårspexet
Verifierbarhet	Gå in och välj att redigera en enskild bokning, verifiera att status går att ändra och att den sparas.

Kravnummer	7
Krav	Lägga till Teater
Beskrivning	Det ska gå att bygga nya teatrar med varierande mängd platser och sektioner. Platserna ska kunna placeras ut så att de överensstämmer med salongsskissen. I standardutförandet 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	High
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.

Kravnummer	8
Krav	Lägga till Omgång
Beskrivning	Man ska kunna lägga till nya omgångar i systemet från administratörsinterfacet. Eftersom man annars inte kan visa samma föreställningar flera gånger utan att behöva skapa en ny föreställning för varje visning. Vilket skulle vara väldigt opraktiskt för kunden.
Motivering	Kårspexet vill kunna skapa nya omgångar.
Behov	Standard
Prioritet	High
Källa	Kårspexet
Verifierbarhet	Det ska gå att från administratörens interface lägga till en ny omgång. Denna ska sedan synas i databasen.

Kravnummer	9
Krav	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
Källa	Kårspexet
Verifierbarhet	Efter att information har ändrats, ska man kunna se ändringen via bokningssidan.

Kravnummer	10
Krav	Lägg till föreställning
Beskrivning	I bokningssystemets administratörsvy skall det gå att lägga till nya föreställningar som tillhör en omgång. En omgång är ett vist spex på en viss teater, en viss period. En föreställning är en av de bokbar tillfällena för en omgång.
Motivering	Kårspexet gör nya omgångar med föreställningar varje år. Därför måste Kårspexet kunna lägga till föreställningar.
Behov	Standard
Prioritet	Normal
Källa	Kårspexet
Verifierbarhet	Testa att lägga till en ny föreställning från administrationsvyn. Kontrollera att en föreställning kan visas och döljas i kundens och säljarens vy, genom att markera föreställningen som bokningsbar/inte bokningsbar.

Kravnummer	11
Krav	Redigera föreställningar
Beskrivning	Man ska kunna redigera tillagda föreställningar från administratörs vyn. Annars kan man inte uppdatera dem med ny information som är mera väsentlig.
Motivering	Utan kravet kan kunden inte använda systemet på tänkt sätt
Behov	Standard
Prioritet	Normal
Källa	Kårspexet
Verifierbarhet	Efter att man har lagt till en föreställning ska man kunna redigera den.

Kravnummer	12
Krav	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 betalning och sedan ge ut biljetten till kunden. Det ska även gå 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
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 bekräfta att betalningen är gjord.

Kravnummer	13
Krav	Möjlighet att navigera i systemen
Beskrivning	I varje vy (kund, säljare, ekonomichef, administratör), ska alla tillhörande funktioner kunna nås via länkar på webbsidorna.
Motivering	Användaren måste kunna komma åt all funktionalitet.
Behov	Standard
Prioritet	Normal
Källa	Kårspexet
Verifierbarhet	Man testar att alla länkarna finns, och att de inte är trasiga.

Kravnummer	14
Krav	Säljare ska kunna lämna ut biljetter
Beskrivning	Säljare ska i sin vy kunna hitta en kund och se vilka biljetter han ska lämna ut till honom och därefter registrera dem som uthämtade.
Motivering	Säljaren behöver ha ett interface för att lämna ut förköpta biljetter.
Behov	Standard
Prioritet	Normal
Källa	Kårspexet
Verifierbarhet	Man kollar att systemet ger de korrekta biljetterna som svar.

Kravnummer	15
Krav	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
Källa	Nyx
Verifierbarhet	Vid bokningssteget där man väljer sektioner skall det gå att interagera med översiktsbilden.

Kravnummer	16
Krav	Avbokning
Beskrivning	En kund skall kunna avboka sin bokning. 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	Low
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.

Kravnummer	17
Krav	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	Low
Källa	Kårspexet
Verifierbarhet	Kontrollera att man kan se den enklare statistiken från administratörens och ekonomichefens gränssnitt.

Kravnummer	18
Krav	Omfattande statistik
Beskrivning	Det skall vara möjligt att från administratörens och ekonomichefens gränssnitt se mycket utförlig statistik från bokningssystemet. Detta inkluderar, men är ej begränsat till: enklare statistik inom vissa tidsintervall. För föreställning 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 föreståelse i bokningen vilket underlättar framtida planering och arbete av nya föreställningar, omgångar och spex.
Behov	Plus
Prioritet	Normal
Källa	Kårspexet
Verifierbarhet	Vi verifierar att statistiken stämmer jämfört med testdata. Kontrollerar att det går att filtrera efter tid och eventuell rabattklass.

Kravnummer	19
Krav	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
Källa	Kårspexet
Verifierbarhet	Administratören kan göra bokningar, även registrera gratisbiljetter. Osv Förklara hur Karl ska kolla att (1) kravet finns i designen och (2) att mjukvaran implementerar kravet.

Kravnummer	20
Krav	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
Källa	Nyx
Verifierbarhet	Kolla så att bilden har korrekt startfärger när det inte finns några bokningar i systemet. Kolla gradvis efter övergångar mellan färger, allt eftersom fler bokningar görs.

Kravnummer	21
Krav	Filtrera bokningar
Beskrivning	Administratörsgränsnitten ska erbjuda möjligheten att filtrera bokningar efter betalnings- och placeringsstatus liksom bokningsnummer, föreställning samt kontaktpersonens namn.
Motivering	För att underlätta administrationen av bokningar.
Behov	Standard
Prioritet	Normal
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.

Kravnummer	22
Krav	Kortköp
Beskrivning	Möjlighet för kund att välja att betala med kort vid bokning.
Motivering	Kårspexet vill ha kortbetalning som ett smidigt betalsätt för slutanvändaren.
Behov	Deluxe
Prioritet	Normal
Källa	Kårspexet
Verifierbarhet	Välja att betala med kort vid slutförande av en bokning.

Kravnummer	23
Krav	Kontohantering
Beskrivning	Administratörer ska kunna hantera de olika kontona och byta lösenord på dem. Hur mycket som kan göras här beror på hur kontona utformas.
Motivering	Administratörer måste kunna hantera kontona.
Behov	Standard
Prioritet	Normal
Källa	Kårspexet
Verifierbarhet	Kontrollera att det går att gå in som administratör och redigera/skapa/hantera konton.

Kravnummer	24
Krav	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
Källa	Kårspexet
Verifierbarhet	Kontrollera att man från administratörens och ekonomichefens gränssnitt kan se den grafiska representationen av statistik.

Kravnummer	25
Krav	Hantera utskick
Beskrivning	Admin/Ekonomichef ska kunna hantera de massutskick av mail som görs av systemet.
Motivering	Det är bra att kunna kontakta olika grupper av slutkunder, till exempel de som inte betalat sina biljetter.
Behov	Standard
Prioritet	Low
Källa	Kårspexet
Verifierbarhet	Kontroll av modell och design. Man kan testa att göra mailutskick.

Kravnummer	26
Krav	Sortera bokningar
Beskrivning	Administrationsgränssnittets 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	
Behov	Standard
Prioritet	Normal
Källa	Nyx
Verifierbarhet	Klicka på de olika kolumnrubrikerna i bokningslistan och verifiera att bokningarna sorteras i korrekt ordning.

Kravnummer	28
Krav	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
Källa	Kårspexet
Verifierbarhet	Från Administratörens och Ekonomichefens gränssnitt kontrolleras att en bokning kan ändra betalningsstatus.

Kravnummer	29
Krav	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	Standard
Prioritet	Low
Källa	Nyx
Verifierbarhet	Kontrollera att inget utskick görs utan att en förhandsgranskning först visas.

Kravnummer	30
Krav	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	Low
Källa	Kårspexet
Verifierbarhet	Det går att generera nytt lösenord när man administrerar säljarkontot.

Kravnummer	31
Krav	Tidsmätning
Beskrivning	För varje anrop till systemet, ska tiden det tar att svara mätas. Tiden ska skrivas till en logg.
Motivering	För att verifiera att kravet om svarstid har uppfyllts.
Behov	Standard
Prioritet	Low
Källa	Kalle Arvidsson, Johan Stjernberg
Verifierbarhet	Kontrollera att tiden står i loggen.

3.1 Begränsande krav

3.1.1 Prestanda

Krav	serverbelastning
Beskrivning	Systemet ska utan märkbara problem hantera minst tio typiska användare samtidigt. Hur väl kravet uppfylls beror mycket på serverns prestanda.
Motivering	Det kommer förekomma fall då flera använder systemet samtidigt. Hänvisa till antagande i sektion 2, Allmänna begränsningar > tekniska begränsningar eller Antaganden och beroenden.
Behov	Standard
Prioritet	Låg
Stabilitet	Stabilt
Källa	Johan Stjernberg, Kalle Arvidsson
Verifierbarhet	Testkörning av systemet på Kårspexets server med tio eller fler användare.

Krav	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.

3.1.2 Säkerhet

Krav	säkerhet
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

3.1.3 Miljö

Krav	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	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	webbläsarkompatibilitet
Beskrivning	Det ska gå använda bokningssystemets alla funktioner med följande webbläsare: _Firefox 3_, _Internet Explorer 8_, _Google Chrome (version ?)_, _Safari 3_, Opera (?). Inga inställningar ska behöva göras utifrån deras standard-konfigurationer(?).
Motivering	Dessa webbläsare är stora på marknaden just nu och bör stödas av vårt system. Referera till statistik!
Behov	Standard
Prioritet	Låg
Stabilitet	Instabilt
Källa	Nyx
Verifierbarhet	Provkörning av systemet i dessa webbläsare.

Krav	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 gränssnittet 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	Internet Explorer 7
Beskrivning	Kravet webbläsarkompatibilitet uppfylls även för webbläsaren _Internet Explorer 7_.
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	Victor Hallgren(?)
Verifierbarhet	Provkörning av systemet med _Internet Explorer 7_.

3.1.4 Användbarhet

Krav	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	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	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	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.

3.1.5 Externa system

Krav	mysql
Beskrivning	Bokningssystemet ska använda databashanteraren MySQL.
Motivering	Gruppen vill använda MySQL och Kårspexet har samtyckt.
Behov	Standard
Prioritet	Hög
Stabilitet	Stabilt
Källa	Nyx, Kårspexet(?)
Verifierbarhet	Uppvisning av databas eller kontroll av källkod.

Krav	apache
Beskrivning	Bokningssystemet ska använda webbservern Apache.
Motivering	Kårspexets webbplats använder Apache (?).
Behov	Standard
Prioritet	Hög
Stabilitet	Stabilt
Källa	Kårspexet(?)
Verifierbarhet	Visa att Apache körs på servern.

Krav	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	Nyx, Kårspexet(?)
Verifierbarhet	Om kortbetalning fungerar används ett externt system. För verifiering av att kortbetalning fungerar, se kravet kortbetalning(?).