



Bokningssystem för Kårspexet

Software Requirements Document (SRD)

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 0.5
February 3, 2011

Abstract

This document describes the requirements of Nyx's software for the client Kårspexet. The main purpose of this document is to facilitate the requirements for Nyx's developers, by interpreting and expanding on the user requirements document (URD) for the same project. The document details the software's functionality and goes more in depth with how that is to be implemented in the product.

Ändringslogg

Version	Ändringar
0.5	Revidering av krav. Granskningsversion inför inlämning.
0.4	Vissa kravförändringar. Citat från PSS-standarderna borttagna.
0.3	Ny version inför granskning.
0.2	Prototypbilder inkluderade i dokumentet.
0.1	Första sammanställd version av dokumentet.

Dokumentversioner

Dokumentet har genererats från följande deldokument.

SRD/abstract version: 3.

SRD/Ändringslogg version: 8.

Gruppmedlemmar version: 3.

SRD/Introduktion version: 2.

SRD/Introduktion/Syfte version: 8.

SRD/Introduktion/Mjukvarans omfattning version: 6.

SRD/Introduktion/Definitioner akronymer och förkortningar version: 50.

SRD/Introduktion/Källor version: 9.

SRD/Introduktion/Dokumentöversikt version: 9.

SRD/Allmän beskrivning version: 3.

SRD/Allmän beskrivning/Relation till aktuella projekt version: 10.

SRD/Allmän beskrivning/Relation till tidigare och kommande projekt version: 11.

SRD/Allmän beskrivning/Funktion och syfte version: 7.

SRD/Allmän beskrivning/Miljömässiga betänkanden version: 11.

SRD/Allmän beskrivning/Relation till andra system version: 19.

SRD/Allmän beskrivning/Generella begränsningar version: 26.

SRD/Allmän beskrivning/Modellbeskrivning version: 22.

SRD/Specifika krav version: 12.

SRD/Specifika krav/Funktionalitetskrav version: 138.

SRD/Specifika krav/Prestandakrav version: 37.

SRD/Specifika krav/Gränssnittskrav version: 52.

SRD/Specifika krav/Körbarhetskrav version: 11.

SRD/Specifika krav/Resurskrav version: 17.

SRD/Specifika krav/Verifikationskrav version: 19.

SRD/Specifika krav/Test av användaracceptans version: 15.

SRD/Specifika krav/Dokumentationskrav version: 30.

SRD/Specifika krav/Säkerhetskrav version: 20.

SRD/Specifika krav/Portabilitetskrav version: *12*.

SRD/Specifika krav/Kvalitetskrav version: *40*.

SRD/Specifika krav/Tillförlitlighetskrav version: *13*.

SRD/Specifika krav/Underhållskrav version: *10*.

SRD/Specifika krav/Trygghetskrav version: *13*.

SRD/Spårningsmatris mellan användarkrav och mjukvarukrav version: *16*.

SRD/appendix version: *1*.

Gruppmedlemmar

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	6
1.1	Syfte	6
1.2	Mjukvarans omfattning	6
1.3	Definitioner akronymer och förkortningar	6
1.4	Källor	8
1.5	Dokumentöversikt	9
2	Allmän beskrivning	10
2.1	Relation till aktuella projekt	10
2.2	Relation till tidigare och kommande projekt	10
2.3	Funktion och syfte	10
2.3.1	Boka biljetter på webben	10
2.3.2	Registrera betalningar	10
2.3.3	Administrera biljetter	11
2.3.4	Lämna ut bokade biljetter	11
2.3.5	Sälja biljetter direkt	11
2.3.6	Administrera mailutskick	11
2.3.7	Administrera föreställningar och teatrar	11
2.3.8	Administrera konton	11
2.3.9	Statistik	11
2.4	Miljömässiga betänkanden	12
2.4.1	Användarna	12
2.5	Relation till andra system	12
2.6	Generella begränsningar	13
2.6.1	Datamodell	13
2.6.2	Resurser	13
2.6.3	Tekniska begränsningar	14
2.7	Modellbeskrivning	14
2.7.1	Kundgränssnittet	14
2.7.2	Säljargränssnittet	15
2.7.3	Ekonomichefsgränssnittet	15
2.7.4	Admingränssnittet	16
2.7.5	Systemkomponenter	16
3	Specifika krav	18
3.1	Funktionalitetskrav	18
3.1.1	Funktionalitet för autentisering	18
3.1.2	Funktionalitet för kund	18
3.1.3	Funktionalitet för säljaren	19
3.1.4	Funktionalitet för administratör	20
3.1.5	Funktionalitet som föregående funktionalitet beror på	24
3.1.6	Övrig funktionalitet	27
3.2	Prestandakrav	28
3.3	Gränssnittskrav	29
3.3.1	Mjukvaruinterface	29
3.3.2	Kommunikationsinterface	30
3.3.3	Arbetsflödesanalys av användarscenarion	30
3.3.4	Interfaceprototyper	33

3.4	Körbarhetskrav	35
3.5	Resurskrav	36
3.6	Verifikationskrav	37
3.7	Test av användaracceptans	38
3.8	Dokumentationskrav	39
3.8.1	Kund	39
3.8.2	Kårspexet	39
3.8.3	Systemutvecklare	40
3.9	Säkerhetskrav	41
3.10	Portabilitetskrav	43
3.11	Kvalitetskrav	43
3.12	Tillförlitlighetskrav	45
3.13	Underhållskrav	46
3.14	Trygghetskrav	46
4	Kravspårningsmatris	48
4.1	Administration	48
4.2	Bokning	48
4.3	Diverse	49
4.4	Loginsystem	49
4.5	Prestanda	49
4.6	Säkerhet	50
4.7	Miljö	50
4.8	Användbarhet	50
4.9	Externa system	51
4.10	Dokumentreferenser	51

1 Introduktion

1.1 Syfte

Detta dokumentets syfte är att ge projektets utvecklare en överblick av produktens omfattning och mer detaljerad genomgång av dess funktionalitet.

Mjukvarans omfattning (sektion 1.2) bör först läsas igenom, tillsammans med förklaringar av tekniska begreppen (sektion 1.3). Först efter detta bör de mer tekniskt detaljerade sektion 2 och sektion 3 gås igenom.

1.2 Mjukvarans omfattning

Produkten som Nyx utvecklar består av ett webbaserat biljettbokningssystem med ett enkelt användargränssnitt för besökare och administrationsverktyg för Kårspexets personal. Administrationsverktygen består av tre gränssnitt; ett för säljare, ett för ekonomiansvariga och ett för administratörer.

1.3 Definitioner akronymer och förkortningar

Active record *Teknik för att kommunicera med databaser i objekt-orienterade språk. Objekten i databasen kopplas till objekt i programmet.*

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]*

Arbetsminne *Även kallat primärminne. En fysisk komponent i en dator. Används för att lagra program och data då programmet körs.*

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. [1.3.2]*

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

Firefox (Mozilla Firefox) *En webbläsare som går att köra på de populäraste operativsystemen.*

Foreign key *Inom relationsdatabaser en begränsning som kräver att information på ett ställe finns definierat tidigare på ett annat ställe (kolumner i tabeller).*

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.*

GUI (Graphical User Interface) *Se gränssnitt.*

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.*

HTTPS (HTTP Secure) *En kombination av HTTP och SSL/TLS med syftet att förhindra avlyssning av HTTP-trafiken*

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

Internet Explorer *En webbläsare utvecklad av Microsoft för operativsystemet Windows.*

InvalidAuthenticityToken *Ett fel som kan uppstå i rails om användaren backar på sidor innehållande formulär. Uppstår på grund av Rails skydd mot XSS.*

KiB (kibibyte) 2^{10} byte.

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

MiB (mebibyte) 2^{20} byte.

Mib (mebibit) 2^{20} bit.

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

MVC (Model-View-Controller) *Se Model-View-Controller.*

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

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

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

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

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

RDoc (Ruby Doc) *Verktyg för att generera dokumentation för Ruby-källkod i HTML-format.*

Ruby *Ett objektorienterat programmeringsspråk.*

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

Sjöslaget *Årligen återkommande studentfest på Finlandsfärja.*

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

SQL-injection *En metod för att förändra eller komma åt data i en databas genom att ange strängar i användarinterfacet som förändrar betydelsen av en SQL-fråga.*

SHA-1 *En hashfunktion.*

SSL/TLS (Secure Socket Layer/Transport Layer Security) *Ett kryptografiskt protokoll för att sätta upp säkra kommunikationskanaler över internet.*

Testkod *Kod som används för att testa funktioner i programmet så att det returnerar förväntat svar för att försäkra sig om att mjukvaran fungerar på ett tillfredsställande sätt.*

Tutorial *En metod för att överföra kunskap som ofta används vid inläring.*

URD (User Requirements Document) *Dokument inom PSS050 standarden där användarens krav specificeras.*

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.*

XSS (Cross site scripting) *En teknik som utnyttjar svagheter i en webbsida genom att låta en auktoriserad användare accessa en länk som modifierar sidan på ett sätt som den auktoriserade användaren inte önskar [1.3.3].*

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 “Relation till andra system” (5) under sektion “Allmän beskrivning” (2).

Apache HTTP Server

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

Hänvisning till källan görs från referenserna: [1.3.1].

CentOS

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

Hänvisning till källan görs från referenserna: [1.3.2].

Cross-site-scripting

http://en.wikipedia.org/wiki/Cross-site_scripting

Hänvisning till källan görs från referenserna: [1.3.3].

1.5 Dokumentöversikt

Detta dokument innehåller fyra distinkta delar. Sektion 1 är denna introduktion till projektet. Sektion 2 beskriver produktens funktionalitet och presenterar dessutom externa faktorer kring produkten. Sektion 2.7 är intressant för utvecklare, ty den innehåller en översikt av systemets komponenter och typiska användarscenarion. Sektion 3 beskriver de specifika krav på produkten, som härletts ifrån URD:n. Sektion 4 innehåller en matris där mjukvarukraven kan spåras tillbaka till användarens krav från URD:n.

2 Allmän beskrivning

2.1 Relation till aktuella projekt

Nyx produkt ska bli en del av Kårspexets webbplats och byta ut det gamla bokningssystemet som finns där. Till vår kännedom är det ingen annan som gör några större ändringar på Kårspexets webbplats eller det gamla bokningssystemet.

2.2 Relation till tidigare och kommande projekt

För tillfället använder sig Kårspexet av Fysikalens bokningssystem. Det skulle vara möjligt att få tillgång till det systemet för att se hur det fungerar. Detta för att få en bättre inblick i vad som är problematiskt och vad som fungerar bra som vi bör återskapa.

Kårspexet har uttryckligen bett om att det nya bokningssystemet ska passa in på deras nuvarande webbplats utan att designen krockar för mycket. Detta gäller främst kundens gränssnitt, men inte de övriga.

Victor Hallberg har gjort Sjöslagets bokningssystem som relaterar till detta projekt eftersom det också är ett bokningssystem. Vissa delar av det projektet skulle vi kunna använda oss av här. Validering av input med javascript och navigation mellan bokningssteg är ett exempel på detta.

Andreas Tarandi har jobbat med kortköp på andra sidor. Om vi väljer att implementera kortköp även här så kan vi ha användning av det.

2.3 Funktion och syfte

Bokningssystemet ska användas av fyra typer av användare: kund, säljare, ekonomichef och administratör. Dessa har olika roller som interagerar med varandra. Varje typ av användare har ett eget gränssnitt till systemet men många av funktionerna är delade.

2.3.1 Boka biljetter på webben

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

2.3.2 Registrera betalningar

Ekonomichefen ska kunna registrera betalningar för bokningar som kunder gjort.

2.3.3 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.

2.3.4 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.

2.3.5 Sälja biljetter direkt

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

2.3.6 Administrera mailutskick

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

2.3.7 Administrera föreställningar och teatrar

Vid nya föreställningar är det administratören som för in dessa 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 skapas. Administratören ska kunna välja vilka föreställningar det går att boka/köpa biljetter till.

2.3.8 Administrera konton

Administratören ska kunna ändra lösenord för alla användarkonton. Säljarnas lösenord ska kunna genereras automatiskt och vara giltigt en begränsad tid.

2.3.9 Statistik

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

2.4 Miljömässiga betänkanden

2.4.1 Användarna

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

Kunder

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 och är vana vid både mail och webbsidor. Hos släktingar och bekanta varierar den tekniska kompetensen stort, där finns både tonåringar och pensionärer. Vi antar att användarna har viss vana vid mail och webben men att de kan ta lång tid på sig att anpassa sig till nya gränssnitt. Kunderna kommer att använda systemet relativt sällan, det är därför viktigt att kundens gränssnitt är enkelt, även första gången man använder det.

Säljare

Säljare är medlemmar i Kårspexet och är därmed sannolikt kårmedlemmar vid KTH, eller har varit det. 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.

Administratör och ekonomichef

Både administratören och ekonomiansvarige är aktiva medlemmar i Kårspexet. De är därmed vana användare av mailklienter och det nuvarande bokningssystemet och kommer använda systemet ofta. Det är acceptabelt att viss utbildning krävs. Användarvänlighetskraven på administratörens gränssnitt är relativt låga.

Systemmiljö

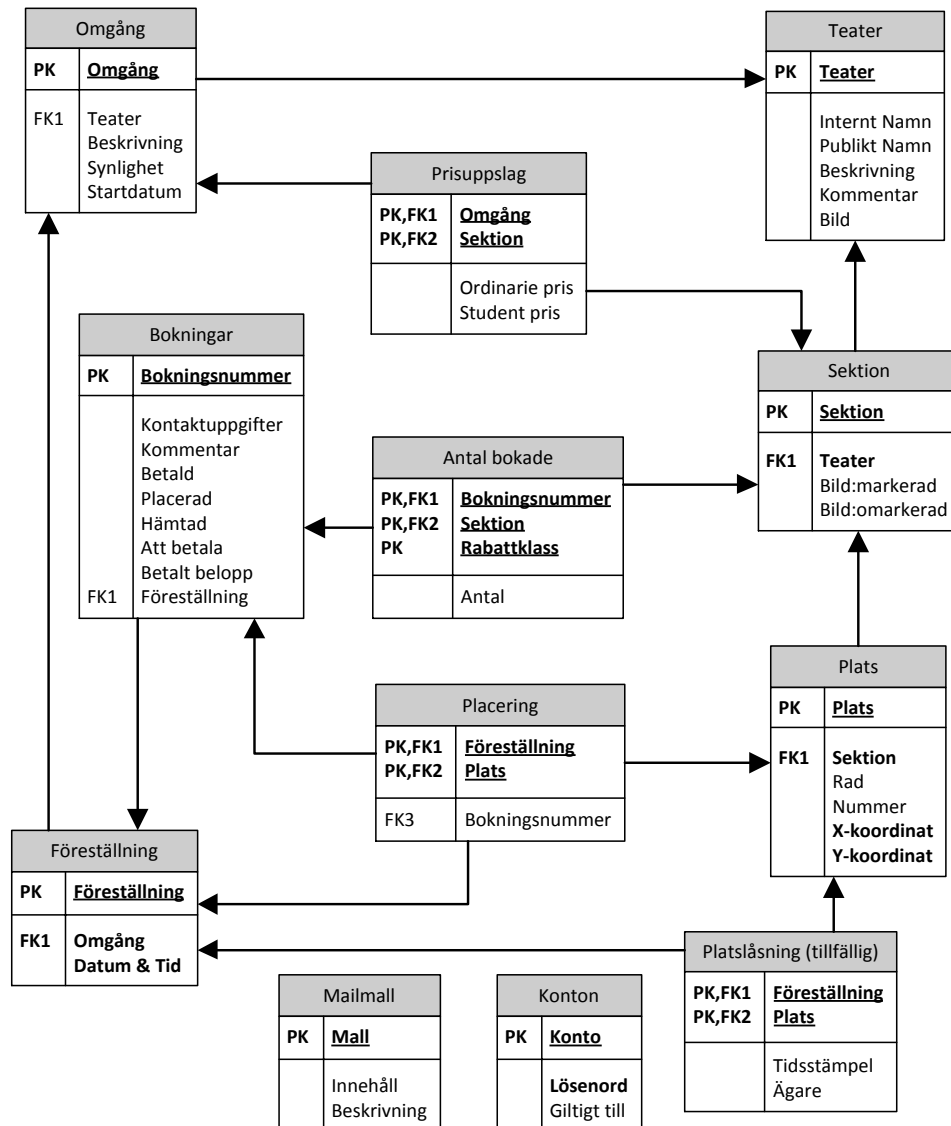
Systemet kommer att köras på Kårspexets server med operativsystemet CentOS och kommer administreras av personer med stor eller mycket stor kunskap om Linux-servrar. De kan antas kunna hantera administreringen och underhållet av systemet själva.

2.5 Relation till andra system

Nyx system kommer att köras på en server som Kårspexet har tillgång till. På servern körs webbserver, databashanterare och ett operativsystem. Vårt program använder webbservern och databasen för att kommunicera med användarna och databasen för att lagra data. I grundutförandet kommer vårt program i övrigt att vara fristående från andra system, men om vi har tid så kommer det även att använda sig av externa system för att implementera kortbetalningar.

2.6 Generella begränsningar

2.6.1 Datamodell



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.

2.6.2 Resurser

Nyx kommer vara begränsade i vilka och hur många funktioner vi kommer kunna implementera då vi totalt har 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.

Kundbehov

Vårt 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 kvalitén 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 därför 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.

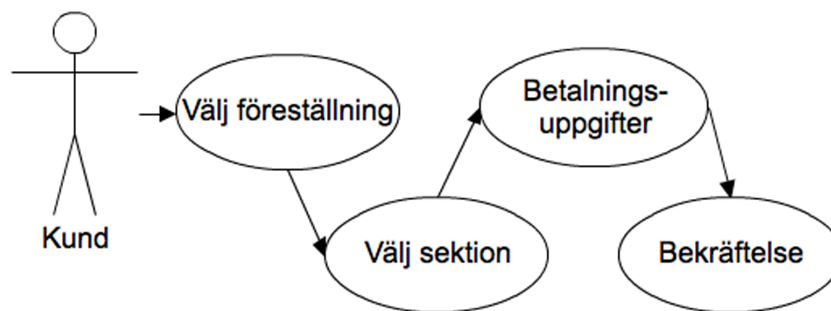
2.6.3 Tekniska begränsningar

Kårspexet står för den server som kommer köra vår webbapplikation. Vi har ingen kontroll över dess hårdvara, men vi har verifierat att operativsystemet som körs på servern är kompatibelt med Apache, Ruby on Rails 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 MiB systemminne. CentOS anger 256 MiB minne samt en klockfrekvens på minst 500 MHz som minimikrav för datorer som kör operativsystemet. Med MySQL och Apache körandes samtidigt utöver dessa bör servern ha minst 1 GiB systemminne och en processor med klockfrekvensen 1 GHz eller högre. Kårspexets server har en processor med klockfrekvensen 2,6 GHz och 1 GiB systemminne, vilket beräknas vara tillräckligt.

2.7 Modellbeskrivning

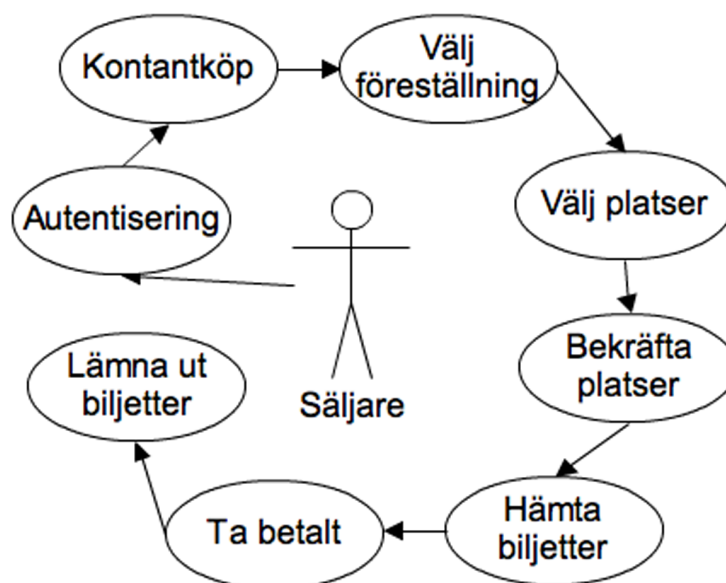
För att bättre förklara hur bokningssystemet fungerar följer nedan användarscenario som på en övergripande nivå visar de fyra gränssnitt vi kommer implementera för systemets användare.

2.7.1 Kundgränssnittet



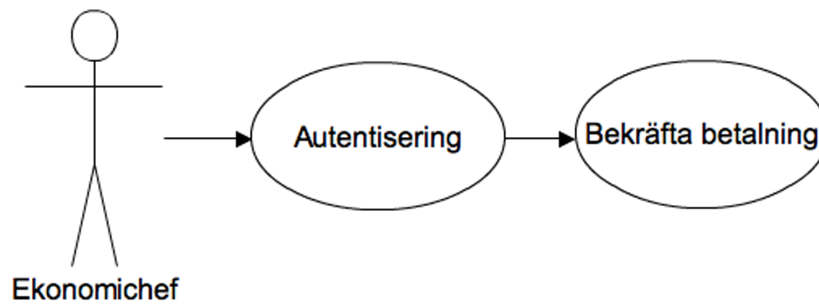
I ett typiskt användarscenario så kommer kundens vy att användas på följande sätt; kunden väljer en föreställning och sedan en sektion att sitta i. I nästa steg ger kunden ut sina betalningsuppgifter och får ett bekräftelsemail.

2.7.2 Säljargränssnittet



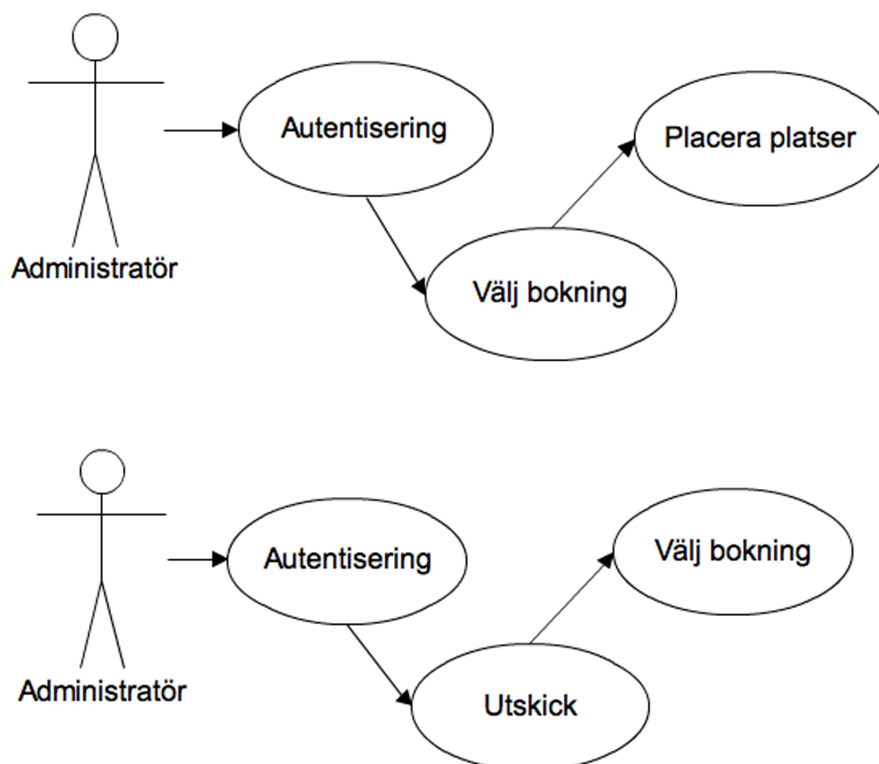
Säljare loggar in i systemet och möts av vyn för kontantköp. Där väljs en föreställning och sedan önskade platser. Säljaren bekräftar platserna och hämtar berörda biljetter till kunden. Sedan tar säljaren betalt och överlämnar biljetterna till kunden.

2.7.3 Ekonomichefsgränssnittet



Ekonomichefen loggar in i systemet och kan därifrån bekräfta inkomna betalningar.

2.7.4 Admingränssnittet



Administratören loggar in i systemet och väljer en bokning där platserna placeras.

Administratören kan också välja att göra mailutskick för en eller flera bokningar.

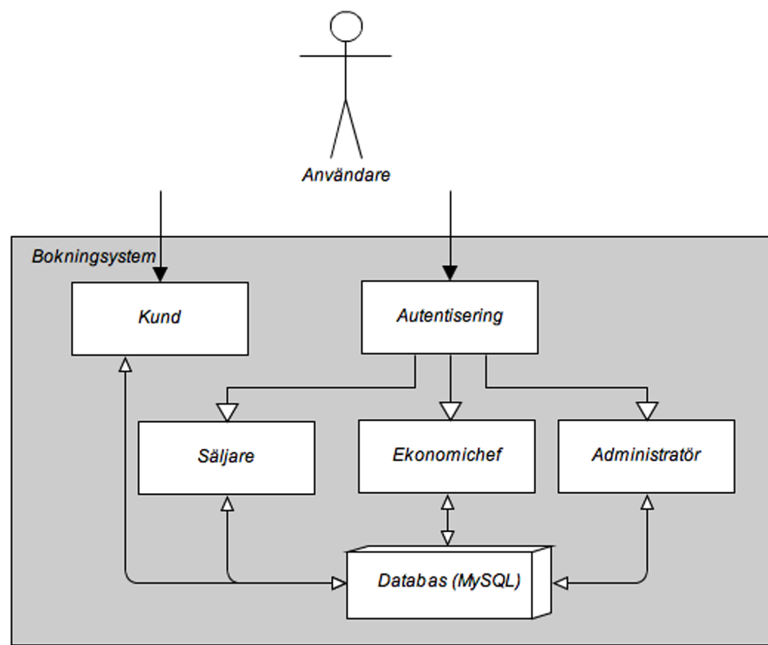
2.7.5 Systemkomponenter

Systemets komponenter består på en översiktsnivå av fyra olika gränssnitt, en autentiseringskomponent och en databas. Tre av de fyra gränssnitten är kopplade till autentiseringskom-

ponenten och kräver att användaren har fått behörighet innan gränssnittet kan nås. Ett av de fyra gränssnitten är inte kopplad till autentiseringskomponenten och är tillgängligt för alla användare.

Alla gränssnittskomponenter har dubbelriktad kommunikation med databasen. Autentiseringskomponenten hämtar endast information från databasen, den sparar aldrig något.

När man ansluter till systemet kommer man normalt in i kundgränssnittet där man kan boka en föreställning. Från kundgränssnittet kan man välja att logga in som säljare, ekonomichef eller administratör. Inloggningen hanteras av autentiseringskomponenten som ger behörighet till de övriga gränssnitten.



Detta ger en bild över hur de logiska delarna i systemet kommer att fungera ihop med den tidigare modellen av databasen som beskrevs i URD 2.7.1.

3 Specifika krav

3.1 Funktionalitetskrav

3.1.1 Funktionalitet för autentisering

Krav	SR1.1 Autentisering
Beskrivning	Säljarens, administratörens och ekonomichefens funktionaliteter ska endast kunna användas av autentiserade användare, i enlighet med SR9.2. Autentisering sker genom inloggning med användarnamn och ett lösenord. Den inloggade användaren ska kunna logga ut, det vill säga att upphäva sin behörighet tills nästa inloggning. För att uppfylla detta behöver systemet hantera användarkonton med olika användarroller.
Behov	Standard
Prioritet	Hög
Stabilitet	Stabilt
Källa	UR4.1
Verifierbarhet	Kontrollera att systemet hänvisar användaren till inloggning och att man efter inloggning får tillgång till användarrollens funktionalitet. Prova även att logga ut och att man därefter inte längre har tillgång till funktionaliteten

3.1.2 Funktionalitet för kund

Krav	SR1.2 Kund: Boka
Beskrivning	En kund ska kunna göra en bokning så som beskrivs i sektion 2.7. Detta inkluderar att välja föreställning, sektion och antal biljetter med student- respektive ordinariepris. Vid slutförd bokning ska kunden få en bokningsbekaftelse både som webbsida och som mail.
Behov	Standard
Prioritet	Hög
Stabilitet	Stabilt
Källa	UR2.6, UR2.10
Verifierbarhet	Pröva att genomföra användarscenerierna för kunden. Kontrollera i administrationsgränssnittet att bokningen finns i systemet.

Krav	SR1.3 Kund: Visa/avboka bokning
Beskrivning	Kunden ska kunna visa eller avboka sin bokning, från och med att bokningen görs och minst fram tills föreställningens avbokningsdeadline. Förslagsvis kan kunden göra detta via en länk i bokningens bekräftelsemail.
Behov	Standard
Prioritet	Låg
Stabilitet	Stabilt
Källa	UR2.4, UR2.6
Verifierbarhet	Pröva att genomföra en bokning och sedan visa och avboka den. Kontrollera avbokningen i administratörsgränssnittet.

Krav	SR1.4 Kund: Kortbetalning
Beskrivning	Vid bokning ska kunden kunna betala sina biljetter med betalkort.
Behov	Deluxe
Prioritet	Medel
Stabilitet	Stabilt
Källa	UR2.8
Verifierbarhet	Man genomför en kortbetalning med ett testkort och en testbetalning

3.1.3 Funktionalitet för säljaren

Krav	SR1.5 Säljare: Nytt kontantköp
Beskrivning	En säljare ska kunna genomföra ett kontantköp. Detta inkluderar att skapa ny bokning, välja föreställning och sittplatser, registrera betalning och registrera biljetterna som uthämtade.
Behov	Standard
Prioritet	Medel
Stabilitet	Stabil
Källa	UR2.1
Verifierbarhet	Testa att genomföra ett kontantköp enligt användarscenariot. Kontrollera i administrationsgränssnittet att en korrekt bokning skapats.

Krav	SR1.6 Säljare: Lämna ut bokade biljetter
Beskrivning	Säljaren ska kunna lämna ut biljetterna för en bokning. Detta innebär, förutom att bekräfta biljettutlämning, eventuellt att välja sittplatser och att ta betalt.
Behov	Standard
Prioritet	Medel
Stabilitet	Stabil
Källa	UR2.2
Verifierbarhet	Pröva att genomföra biljettutlämningen, med bokningar som är placerade respektive oplacerade och betalda respektive obetalda. Vi ska även verifiera att det inte går att lämna ut oplacerade biljetter.

3.1.4 Funktionalitet för administratör

Administration av användarkonton

Krav	SR1.7 Administratör: Hantera konton
Beskrivning	Administratören ska kunna ändra lösenord och deras giltighetstid för konton.
Behov	Standard
Prioritet	Medel
Stabilitet	Stabilt
Källa	UR4.2
Verifierbarhet	Testa att ändra lösenord och att det nya lösenordet fungerar men inte det gamla. Testa att välja en giltighetstid sådan att den gäller vid ändringen. Verifiera att det går att logga in. Vänta tills giltighetstiden har gått ut, testa att logga in igen och verifiera att det inte längre fungerar.

Krav	SR1.8 Administratör: Lösenordsgenerator
Beskrivning	Administratören ska kunna välja ett automatiskt genererat lösenord vid kontohanteringen (SR1.7). Detta för att lösenorden ska bli säkra och för att förenkla administratörens arbete.
Behov	Deluxe
Prioritet	Låg
Stabilitet	Stabilt
Källa	UR4.3
Verifierbarhet	Testa att skapa lösenordet automatiskt vid skapandet av ett konto.

Administration av teatrar och omgångar

Krav	SR1.9 Administratör: Hantera teatrar
Beskrivning	En teater är en lokal där föreställningar i en omgång kan spelas. Administratören ska kunna lägga till, ändra och ta bort teatrar i databasen (ändringar och borttagningar måste vara begränsade så att felaktiga länkar i databasen undviks). Data som förknippas med teatern är namn, beskrivning, etc, samt sektioner, platser och bild av sektioner och platser.
Behov	Standard
Prioritet	Hög
Stabilitet	Stabilt
Källa	UR1.12
Verifierbarhet	Testa att skapa och sedan redigera en teater. Kontrollera att teatern kan väljas när en omgång skapas.

Krav	SR1.10 Administratör: Hantera omgångar
Beskrivning	Administratören ska kunna lägga till, ta bort och ändra omgångar och deras föreställningar.
Behov	Standard
Prioritet	Medel
Stabilitet	Stabilt
Källa	UR1.1, UR1.2, UR1.13, UR1.14
Verifierbarhet	Testa att skapa en omgång, skapa och ta bort föreställningar för omgången, ändra info om omgången och att ta bort omgången.

Administration av bokningar

Krav	SR1.11 Administratör: Placera bokningar
Beskrivning	Administratören ska kunna välja sittplatser åt bokningar.
Behov	Standard
Prioritet	Hög
Stabilitet	Stabilt
Källa	UR2.13
Verifierbarhet	Verifiera SR1.25 genom administratörens gränssnitt.

Krav	SR1.12 Administratör: Registrera betalningar
Beskrivning	Administratören ska kunna registrera att bokningar har betalats. Det ska också gå att ändra en boknings betalningsstatus från betald till obetald.
Behov	Standard
Prioritet	Medel
Stabilitet	Stabilt
Källa	UR2.9
Verifierbarhet	Testa att registrera en betalning. Sök sedan upp bokningen igen för att kontrollera att den blivit betald.

Krav	SR1.13 Administratör: Hantera bokningar
Beskrivning	Administratören ska kunna lägga till, ändra och ta bort bokningar. Bokningar ska kunna markeras som gratis. Alla ändringar ska bekräftas med en bekräftelsesida där ändringarna visas.
Behov	Standard
Prioritet	Medel
Stabilitet	Stabilt
Källa	UR2.5, UR2.14
Verifierbarhet	Testa att skapa gratisbiljetter, ändra någon uppgift och sedan ta bort bokningarna. Notera att bekräftelsesidor visas för varje ändring. Genom att hitta bokningen igen och visa, ändra och ta bort den, har man verifierat att bokningen lagrats i databasen.

Krav	SR1.14 Administratör: Mailutskick
Beskrivning	Administratören ska kunna söka upp bokningar och göra ett mailutskick till kunderna, baserat på en mail-mall. Det ska också gå att skapa nya mail-mallar.
Behov	Standard
Prioritet	Medel
Stabilitet	Stabilt
Källa	UR1.7
Verifierbarhet	Testa att skapa en ny mail-mall och att göra ett utskick med denna. Kontrollera att man får förhandsgranska utskicket. Kontrollera även att mailen kommer fram och att mallarna använts korrekt.

Krav	SR1.15 Administratör: Mailutskick förhandsgranskning
Beskrivning	Mailutskick (SR1.14) ska förhandsgranskas innan de genomförs. Innan utskicket görs ska en förhandsgranskning i form av ett av (eventuellt flera) meddelanden visas.
Behov	Plus
Prioritet	Låg
Stabilitet	Stabilt
Källa	UR1.9
Verifierbarhet	Kontrollera att inget utskick görs utan att en förhandsgranskning först visats.

Statistik

Krav	SR1.16 Administratör: Statistik
Beskrivning	Administratören ska få statistik om bokningarna, som hjälp för att planera aktuellt och kommande arbete. Statistiken ska inkludera antal bokade respektive utgivna biljetter per föreställning, omgång och spelår och rabattklass.
Behov	Standard
Prioritet	Låg
Stabilitet	Stabilt
Källa	UR1.3
Verifierbarhet	Kontrollera att statistiken följer beskrivningen och att den förefaller korrekt.

Krav	SR1.17 Administratör: Statistik Plus
Beskrivning	Mer utförlig statistik än enligt SR1.16. Detta inkluderar, statistik över tidsintervall, bokningar beroende på föreställnings start-/sluttid, tidpunkter för biljettutlämningar.
Behov	Plus
Prioritet	Medel
Stabilitet	Stabilt
Källa	UR1.4
Verifierbarhet	Kontrollera att statistiken följer beskrivningen och att den förefaller korrekt.

Krav	SR1.18 Administratör: Statistik Deluxe
Beskrivning	Statistiken (SR1.16, SR1.17) ska framställas grafiskt.
Behov	Deluxe
Prioritet	Medel
Stabilitet	Stabilt
Källa	UR1.6
Verifierbarhet	Kontrollera att statistiken följer beskrivningen och att den förefaller korrekt.

Ekonomichefens gränssnitt

Krav	SR1.19 Administratör: Ekonomichefens gränssnitt
Beskrivning	Det ska finnas en begränsad version av administrationsgränssnittet, som är speciellt anpassat för den ekonomichefen. Följande funktioner ska inkluderas: registrera betalningar, statistik.
Behov	Plus
Prioritet	Låg
Stabilitet	Stabilt
Källa	UR1.10
Verifierbarhet	Pröva att logga in med ett ekonomichef-konto och kontrollera att man får tillgång till de tre funktionerna som nämns i beskrivningen.

3.1.5 Funktionalitet som föregående funktionalitet beror på

Krav	SR1.20 Sök bokningar
Beskrivning	För många av uppgifterna systemets användare ska utföra behöver man kunna söka upp och välja ut bokningar. Detta gäller de funktioner som beskrivs av SR1.6 och SR1.11-SR1.14. Detta ska lösas av en flexibel modul som kan anpassas till de olika användningarna. Olika uppsättningar av egenskaper hos bokningarna ska kunna visas och bokningarna ska kunna sorteras och filtreras på de olika egenskaperna. Användaren ska själv kunna anpassa sorteringen och filtreringen.
Behov	Standard
Prioritet	Hög
Stabilitet	Stabilt
Källa	UR1.5, UR1.8
Verifierbarhet	För det administrativa gränssnittet ska det gå att sortera på bokningarnas olika attribut och utföra olika filtreringar.

Krav	SR1.21 Föreställningsväljare
Beskrivning	Vid skapandet av en bokning måste en föreställning väljas. Det ska bara gå att välja en föreställning i en aktuell omgång.
Behov	Standard
Prioritet	Medel
Stabilitet	Stabilt
Källa	UR2.1, UR2.5, UR2.6
Verifierbarhet	Testa de funktioner som är beroende av denna funktion, förslagsvis SR1.2 Kund: Boka. Om det går att välja bland de aktuella omgångarna är kravet uppfyllt.

Krav	SR1.22 Sektionsplacering
Beskrivning	För att låta kunden välja sektion vid bokningen och eventuellt för att låta administratören och säljaren välja sittplatser, behövs funktionalitet för att visa en bild av teaterns sektioner och låta användaren välja en sektion.
Behov	Standard
Prioritet	Medel
Stabilitet	Stabilt
Källa	UR2.6
Verifierbarhet	Gör en bokning som kund. Notera om valet av sektion visas i en bild.

Krav	SR1.23 Sektionsplacering färgkodning
Beskrivning	I Sektionsplaceringen (SR1.22) ska antal lediga platser i de olika sektionerna visas med en färgskala. Detta för att kunden snabbt ska förstå vilka sektioner han kan få plats i.
Behov	Plus
Prioritet	Medel
Stabilitet	Stabilt
Källa	UR2.7
Verifierbarhet	Låt bokningar tillkomma (eller försvinna). Kontrollera att korrekta färger visas vid sektionsvalet.

Krav	SR1.24 Interaktiv sektionsplacering
Beskrivning	I Sektionsplaceringen (SR1.22) ska kundens val av sektion visas i bilden. Med stor sannolikhet kommer valet av sektion att göras i en lista utanför bilden (i ett HTML-formulär). Den sektion som muspekaren befinner sig över i listan eller bilden, ska markeras både i listan och bilden.
Behov	Plus
Prioritet	Medel
Stabilitet	Stabilt
Källa	UR2.3
Verifierbarhet	Kontrollera att sektionsvalet visas interaktivt, genom att sektioner som pekas på markeras.

Krav	SR1.25 Platsplacering
Beskrivning	Säljare och administratören ska kunna välja sittplatser åt bokningarna. Därför behöver en mängd platser (stolar) i en teater kunna visas och väljas. För en viss föreställning ska det visas vilka platser som är upptagna och bara lediga platser ska kunna väljas.
Behov	Standard
Prioritet	Medel
Stabilitet	Stabil
Källa	UR2.13
Verifierbarhet	Testa att välja platser för en bokning. Kontrollera att upptagna platser markeras och inte kan väljas. Kontrollera att bokningen har fått platser genom att söka upp den i administrationsgränssnittet.

Krav	SR1.26 Bekräftelse
Beskrivning	Vid de flesta inmatningar ska, för att säkerställa att korrekta uppgifter ges, en bekräftelsesida visas för användaren. Datan som angetts ska visas och användaren ska kunna välja att bekräfta att uppgifterna är korrekta.
Behov	Standard
Prioritet	Medel
Stabilitet	Stabilt
Källa	UR2.10
Verifierbarhet	Kravet är uppfyllt om bekräftelsesidor kan användas. Testa till exempel att göra en bokning.

Krav	SR1.27 Mail-skickare
Beskrivning	Systemet ska kunna göra mail-utskick till kunderna, automatiskt (vid bokning SR1.2) och manuellt (SR1.14). Givet en mängd bokningar och en mailmall skapas meddelanden och skickas till de epostadresser som hör till bokningarna. Mail-mallarna ska lagras i databasen. Genom mallen anpassas mailen till de olika bokningarna, uppgifter om namn, bokningsnummer, föreställningens datum, bokade platser, betalstatus, etc, läggs in i meddelandet.
Behov	Standard
Prioritet	Låg
Stabilitet	Stabilt
Källa	UR1.7, UR2.10
Verifierbarhet	Verifiera SR1.14 (Administratör: Mail-utskick).

Krav	SR1.28 Meny
Beskrivning	Administratören, säljaren och eventuellt ekonomiansvarige ska ha gränssnitt där de från en meny kan nå funktioner som hör till respektive användare.
Behov	Standard
Prioritet	Hög
Stabilitet	Stabilt
Källa	UR1.10, UR1.11, UR2.11, UR3.1
Verifierbarhet	Ta en titt på de olika användarnas första-sidor. Kontrollera att det finns länkar till sidor för alla funktionerna och att dessa länkar pekar rätt.

Krav	SR1.29 Databas
Beskrivning	Systemet har data som behöver lagras permanent och säkert. Det behövs en databas. Vi har bestämt oss för att använda databashanteraren MySQL genom ett Active record-gränssnitt.
Behov	Standard.
Prioritet	Hög
Stabilitet	Stabilt.
Källa	UR3.2, UR9.1, UR6.2
Verifierbarhet	Starta om systemet och kontrollera att data inte förlorats. Teatrar, föreställningar, bokningar, användarkonton, mail-mallar och all data dessa beror på ska finnas kvar.

3.1.6 Övrig funktionalitet

Krav	SR1.30 Tidsmätning
Beskrivning	För att kunna kontrollera systemets svarstid (SR2.5), ska för varje anrop till systemet, tiden för att generera svaret skrivas till en logg-fil.
Behov	Standard
Prioritet	Låg
Stabilitet	Stabilt
Källa	UR3.2
Verifierbarhet	Gör ett eller flera anrop och kontrollera att uppmätta tider står i loggen.

3.2 Prestandakrav

Krav	SR 2.1 Trafikmängd för en bokning
Beskrivning	Den maximala datatrafikmängd som får krävas för att en bokning skall kunna genomföras är 4 MiB i var riktning.
Behov	Standard
Prioritet	Medel
Stabilitet	Stabilt.
Källa	UR5.1, UR5.4
Verifierbarhet	Mäta trafiken i båda riktningar mellan klienten och servern då en bokning utförs.

Krav	SR 2.2 Anropsantal för en bokning
Beskrivning	Det maximala antalet anrop som får krävas för att en bokning skall kunna genomföras är 11.
Behov	Standard
Prioritet	Låg
Stabilitet	Stabilt.
Källa	UR5.1, UR5.2
Verifierbarhet	Mäta antalet förfrågningar som görs mot server då en bokning utförs.

Krav	SR 2.3 Maximal svarstid
Beskrivning	Bokningssystemet måste generera webbsidor på högst 1 sekund, förutsatt att operationen berör enstaka poster. Överföringstiden till klienten ej inräknad.
Behov	Standard
Prioritet	Låg
Stabilitet	Stabil.
Källa	UR3.2, UR5.2
Verifierbarhet	Avläsa renderingstiderna i webbservers loggfil.

3.3 Gränssnittskrav

3.3.1 Mjukvaruinterface

Klient

Krav	SR3.1 Webbläsare
Beskrivning	Kårspexets bokningssystem ska stödja följande webbläsare; <i>Firefox 3+</i> , <i>Internet Explorer 8+</i> .
Behov	Standard
Prioritet	Låg
Stabilitet	Instabilt
Källa	UR7.3
Verifierbarhet	Provkörning av systemet i dessa webbläsare.

Krav	SR3.2 Internet Explorer 7
Beskrivning	Kårspexets bokningssystem ska även stödja <i>Internet Explorer 7</i> .
Behov	Plus
Prioritet	Låg
Stabilitet	Instabilt
Källa	UR7.5
Verifierbarhet	Provkörning av systemet i <i>Internet Explorer 7</i> .

Server

Krav	SR3.3 Serverinterface
Beskrivning	Apache används som webbserver för att göra bokningssystemet tillgängligt via HTTP-protokollet.
Behov	Standard
Prioritet	Hög
Stabilitet	Stabilt
Källa	UR9.2
Verifierbarhet	Visa att Kårspexets server kör Apache.

3.3.2 Kommunikationsinterface

Krav	SR3.4 Databasinterface
Beskrivning	Webbservern kommunicerar med MySQL-databasen via Rails med hjälp av Unix-sockets.
Behov	Standard
Prioritet	Hög
Stabilitet	Stabilt
Källa	UR9.1
Verifierbarhet	Verifiera att webbservern kommunicerar med databasen.

3.3.3 Arbetsflödesanalys av användarscenarion

Nyx produkts GUI är uppdelat i tre delar; Kundens, Säljarens och Administratörens GUI. Systemet har även en fjärde användare, Ekonomichefen, som använder en begränsad version av Administratörens GUI.

Kundens GUI

Kundens GUI består av en bokningssida på Kårspexets webbplats. När kunden trycker på "Boka biljett" eller dylik dirigeras han/hon till vår produkt.

Det första kunden får göra är att avgöra vilken föreställning i den nuvarande omgången som denne vill gå på. Här möts dessutom kunden av en grafisk representation av hur fulla de olika sektionerna är, som stöds av en färgskala. Nästa steg är att välja vilken sektion kunden vill sitta i. Kunden får ej välja vilken stol denne vill sitta i, ty det gör administratören. När kunden valt sektion så matar kunden in sina betalningsuppgifter, så som namn, telefonnummer, hur han/hon vill betala (kontant, postgiro, kort) och dylik. Sedan bekräftar kunden sina uppgifter och bokningen är klar.

Kunden kan dessutom i varje steg backa tillbaka till ett tidigare steg i bokningen, förutsatt att bokningen inte är bekräftad.

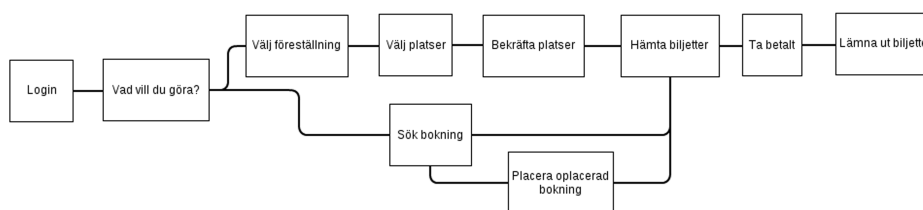
För kundens interfacestruktur, se figur 1.



Säljarens GUI

Betrodda medlemmar av Kårspexet kan agera säljare, och får då ett temporärt lösenord av administratören till en separat inloggningssida. Säljaren loggar in vid sitt säljtillfälle och möts då av två olika val; kontant köp och hämta bokad biljett.

För säljarens interfacestruktur, se figur 2.



Kontantköp

Kontantköp innebär att kunder står i kö för att köpa en biljett direkt från Kårspexet, istället för att boka sin biljett på Kårspexets webbplats först. Säljaren frågar då kunden vilken föreställning och vilken sektion som denne vill gå till. Säljaren placerar ut platser i den valda sektionen på den valda föreställningen och bekräftar sedan dessa uppgifter med kunden. Säljaren får en prompt om att gå och hämta de valda biljetterna, varpå säljaren tar betalt och lämnar ut biljetterna till kunden.

Hämta bokad biljett

Hämta bokad biljett innebär att kunden redan har bokat precis vilken föreställning och sektion som denne vill gå på, och behöver därmed endast hämta ut sina biljetter. Säljaren ber då kunden om dennes namn, bokningsnumret eller dylik, och söker sedan igenom databasen efter bokningen. Om bokningens platser är oplacerade så placerar säljaren dessa manuellt, och hämtar sedan biljetterna. Om kunden valt att betala kontant eller inte betalat tillräckligt för sina biljetter så tar säljaren betalt som vanligt och lämnar ut biljetterna.

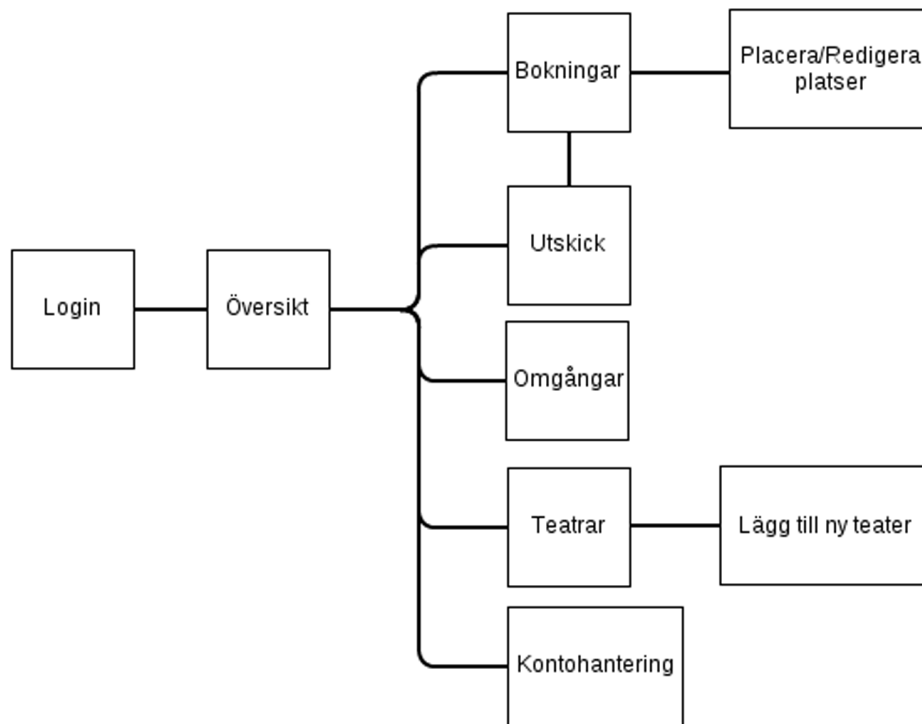
Administratör

En betrodd medlem av Kårspexet har som uppgift att agera administratör över hela Kårspexets bokningssystem. Denne sköter allt administrativt som behöver skötas både framför och bakom kulisserna.

Administratören loggar in på samma sida som säljaren, men med ett annat konto och lösenord. Det första som möter administratören är översiktssidan, som kan ses som fördefinierade sökfilter som till exempel visar hur många platser som är oplacerade, hur många platser det finns kvar per föreställning eller omgång och så vidare.

Administratören har tillgång till en meny som låter henne utföra diverse olika operationer. Administratören kan till exempel lägga till nya omgångar, föreställningar och teatrar via olika gränssnitt. Administratörens huvuduppgift är att placera ut bokningar, så detta gränssnitt får en hög prioritet.

För administratörens interfacestruktur, se figur 3.

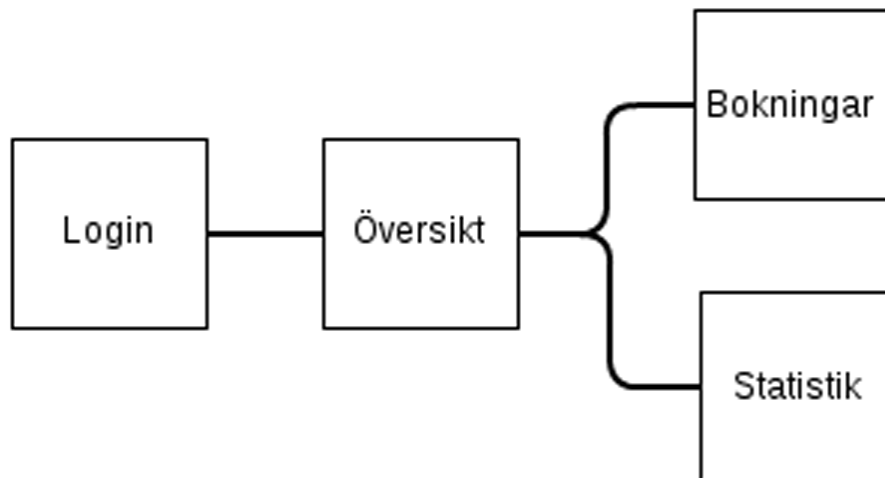


Ekonomichef

En betrodd medlem av Kårspexet agerar ekonomichef. Denne person har kontroll över Kårspexets ekonomi och har som huvuduppgift att hantera bokningsbetalningar. När en betalningen godtagits för biljetten, så kan därefter administratören placera ut platsen.

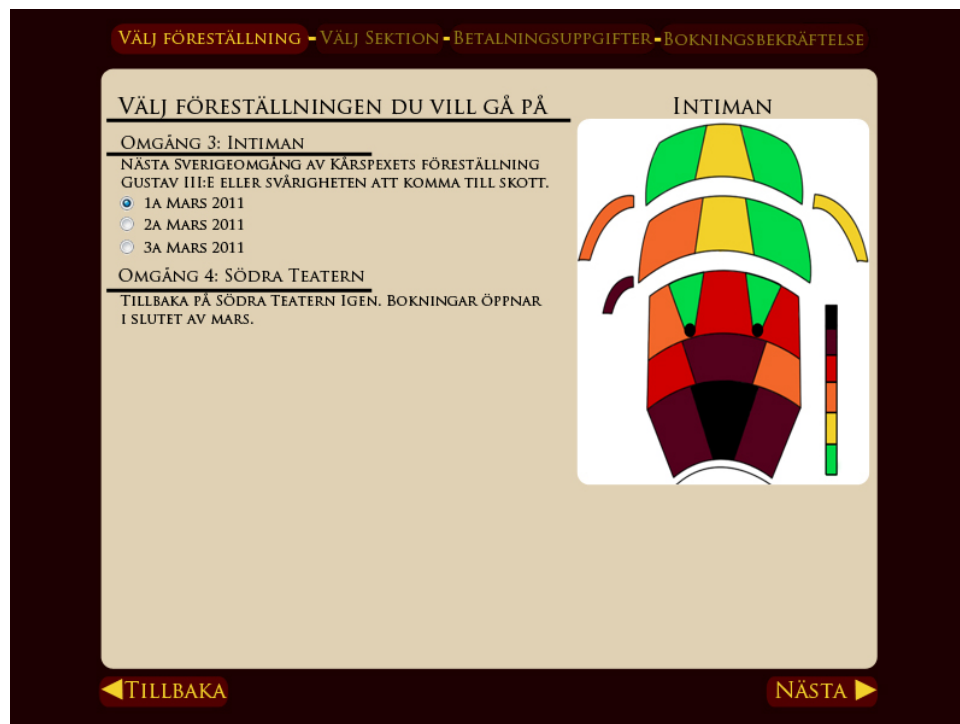
Ekonomichefen ska också ha möjlighet att se mer övergripande statistik om biljettsäljning, vinter och förluster och dylik. Detta kommer kunna ses i en separat flik, som även administratören har tillgång.

För ekonomichefens interfacestruktur, se figur 4.



3.3.4 Interfaceprototyper

Kundens prototyp



Säljarens prototyp

AVBRYT VÄLJ FÖRESTÄLLNING - VÄLJ PLATSER - HÄMTA BILJETTER - TA BETALT - LÄMNA UT BILJETTER

VÄLJ PLATSER I DEN SEKTION KUNDEN VILL HA

VALDA PLATSER FÖR: 11/2 SÖDRA TEATERN

SEKTION	RAD	STOL	RABATTKLASS	PRIS	TA BORT
PARKETT	2	57	<input checked="" type="checkbox"/> STUDENT	120 KR	<input checked="" type="checkbox"/>
PARKETT	2	58	<input type="checkbox"/> STUDENT	170 KR	<input checked="" type="checkbox"/>
TOTALT.....				290 KR	

TILLBAKA NÄSTA

Administratörens prototyp

3.4 Körbarhetskrav

Krav	SR4.1 Servern måste köra Ruby on Rails
Beskrivning	Vår mjukvara kommer kodas i rails, så ett stort krav för att kunna starta applikationen är att servern som används kan köra ruby on rails. Vi planerar att använda Apache och Passenger på Kårspexets server, men det behöver inte nödvändigtvis vara den enda möjliga lösningen för vår mjukvara.
Behov	Standard
Prioritet	Hög
Stabilitet	Stabilt
Källa	UR7.2
Verifierbarhet	Det går att starta applikationen och komma åt den från en webbläsare.

Krav	SR4.2 Mjukvaran ska kunna köras på apache
Beskrivning	Vår mjukvara ska gå att använda tillsammans med Apache. Denna lösning kommer vi använda på Kårspexets server.
Behov	Standard
Prioritet	Hög
Stabilitet	Stabilt
Källa	UR9.2
Verifierbarhet	Test av mjukvaran på en server som kör Apache som webbserver. Om man kan besöka sidan via en webbläsare är kravet uppfyllt.

Krav	SR4.3 Servern måste köra MySQL
Beskrivning	Databasmjukvaran MySQL måste finnas på servern för att vår mjukvara ska fungera.
Behov	Standard
Prioritet	Hög
Stabilitet	Stabilt
Källa	UR9.1
Verifierbarhet	Det går att komma åt applikationen i en webbläsare och göra något enkelt som att till exempel boka en biljett eller skapa en föreställning utan felmeddelanden.

Krav	SR4.4 Säker kortbetalning
Beskrivning	Det externa systemet för kortbetalning behöver fungera på ett säkert och smidigt sätt samt kommunicera med vårt system för att hantera betalningar av denna typ.
Behov	Deluxe
Prioritet	Låg
Stabilitet	Stabilt
Källa	UR2.8, UR9.3
Verifierbarhet	Gör en bokning och välj att betala med kort. Kontrollera kontoutdrag för kund och mottagare. Kontrollera att bokningen registrerats som betald, till exempel via administratörsgränssnittet. Detta krav kostar pengar att uppfylla och verifiera, vi tänker enbart göra det i samarbete med Kårspexet.

3.5 Resurskrav

Krav	SR5.1 Tillgänglig webbserver
Beskrivning	Bokningssystemet körs på en server som är åtkomstbar på internet för kunder såväl som säljare och andra administratörer.
Behov	Standard
Prioritet	Medel
Stabilitet	Stabilt
Källa	UR7.1
Verifierbarhet	Systemet går att använda från en dator som är ansluten till internet och inte befinner sig på samma lokala nätverk som servern.

Krav	SR5.2 Servers prestanda
Beskrivning	Kravet på servers hårdvara omfattande processor och arbetsminne som en helhet. Hårdvaran måste klara av att hantera 1,47 anrop/sekund. Det motsvarar 8 samtidiga användare.
Behov	Standard
Prioritet	Medel
Stabilitet	Stabilt
Källa	UR5.1
Verifierbarhet	Stresstesta servern med flera användare och observera processor- och minnesanvändningen i aktivitetshanterare på servern.

Krav	SR5.3 Servers uppkopplingshastighet
Beskrivning	Kravet på servers uppkopplingshastighet mot internet. Servern måste ha en uppkopplingshastighet på minst 4,27 Mib/s.
Behov	Standard
Prioritet	Medel
Stabilitet	Stabilt
Källa	UR5.4
Verifierbarhet	Stresstesta servern med flera användare och observera nätverkstrafiken i aktivitetshanterare på servern.

3.6 Verifikationskrav

Krav	SR6.1 Testkod
Beskrivning	För att testa att serverns operationer utförs korrekt.
Behov	Standard
Prioritet	Medel
Stabilitet	Stabilt
Källa	UR8.6
Verifierbarhet	Nyx testar manuellt att vår testkod ger tillbaka förväntat svar.

Krav	SR6.2 Användartestning.
Beskrivning	En användargrupp testar systemet i ett rimligt scenario.
Behov	Standard
Prioritet	Medel
Stabilitet	Stabilt
Källa	UR8.1, UR8.2
Verifierbarhet	Nyx genomför kontrollerade tester och observerar användargruppens respons.

3.7 Test av användaracceptans

Krav	SR7.1 Test av användaracceptans
Beskrivning	Alla gränssnitt måste gås igenom och testas mot alla applicerbara krav utlagda i URD:n. Separata testprotokoll för varje gränssnitt med instruktioner för att verifiera att det följer varje applicerbart krav i URD:n ska skapas samt gås igenom och fyllas i av testpersoner.
Motivering	Mjukvaran måste följa de krav som är utlagda av URD:n.
Behov	Plus
Prioritet	Låg
Stabilitet	Stabilt
Källa	UR8.6
Verifierbarhet	Granska ifyllda testprotokoll för kund-, säljar-, ekonomi- samt administratörsgränssnittet och verifiera att de överensstämmer med de krav som ställs i URD:n.

Krav	SR7.2 Spridning av testpersonal
Beskrivning	När testpersonal väljs för att utföra de separata testerna måste personal tilldelas så att varje gränssnitt testas av personer som inte har tidigare erfarenheter av gränssnitt de inte ska ha tillgång till i systemet.
Behov	Plus
Prioritet	Låg
Stabilitet	Instabilt
Källa	UR8.1, UR8.6
Verifierbarhet	Verifiera med hjälp av ifyllda testprotokoll och användarstudier, att det förekommer accepterade tester där testpersonen inte har erfarenhet av andra system än motsvarande roll skulle ha tillgång till. En testperson som testar kund-gränssnittet ska inte ha tidigare erfarenhet av säljar-, ekonomi- eller administratörsgränssnittet. När de administrativa gränssnitten testas får dock testpersonen ha tidigare erfarenhet av kund-gränssnittet, då alla personer har tillgång till kund-gränssnittet.

3.8 Dokumentationskrav

3.8.1 Kund

I det här sammanhanget är en kund den person som skall utföra en bokning på Kårspexets webbplats.

Krav	SR8.1 Bokningsinstruktioner
Beskrivning	Instruktioner för hur man bokar en biljett på bokningswebbsidan. Varje steg i bokningsprocessen skall vara synligt både innan och under en bokning. Beskrivningen skall innehålla vilken information som kunden uppmanas att ange under varje steg.
Behov	Standard
Prioritet	Medel
Stabilitet	Stabil
Källa	UR2.6, UR8.1
Verifierbarhet	Kontrollera att det finns instruktioner för hur bokningen går till på bokningswebbsidan.

3.8.2 Kårspexet

För betrodda medlemmar i kårspexet

Krav	SR8.2 Instruktioner för säljare
Beskrivning	De sysslor som kan utföras av en säljare skall beskrivas var för sig där de utförs i gränssnittet.
Behov	Standard
Prioritet	Medel
Stabilitet	Stabil
Källa	UR2.11
Verifierbarhet	Observera att det finns synliga instruktioner för varje syssla i säljargränssnittet.

Krav	SR8.3 Dokumentation för den ekonomiansvarige
Beskrivning	Instruktioner för hur den ekonomiansvarige kan använda gränssnittets alla funktioner. Utvalda sysslor kan utformas som lathundar. Instruktionerna skall kunna skrivas ut på behagligt pappersformat och hittas på webbplatsen i relation till gränssnittet.
Behov	Plus
Prioritet	Låg
Stabilitet	Stabil
Källa	UR1.10, UR8.5
Verifierbarhet	Kontrollera att dokumentationen kan nås från ekonomigränssnittet.

Krav	SR8.4 Dokumentation för administratören
Beskrivning	Instruktioner för hur administratören kan använda gränssnittets alla funktioner. Utvalda sysslor kan utformas som lathundar. Återkommande sysslor som utförs sällan bör ha en tutorial. Exempel på sådan syssla är "planering av ny föreställning". Instruktionerna skall kunna skrivas ut på behagligt pappersformat och hittas på webbplatsen i relation till gränssnittet.
Behov	Plus
Prioritet	Låg
Stabilitet	Stabil
Källa	UR1.11, UR8.2, UR8.5
Verifierbarhet	Kontrollera att instruktionerna kan nås från administrationsgränssnittet.

3.8.3 Systemutvecklare

För utvecklare som skall anpassa eller berika produktens funktionalitet.

Krav	SR8.5 Databasens struktur
Beskrivning	Dokumentation om hur databasen är strukturerad. Dokumentationen skall omfatta: var information sparas, hur poster mellan tabeller länkas (foreign keys), vilka regler och begränsningar (constraints) som finns. Det bör finnas exempelbeskrivningar för SQL-uttryck. Dokumentationen skall finnas tillgänglig från administratörens gränssnitt och i källkodsmappen för bokninssystemet.
Behov	Plus
Prioritet	Låg
Stabilitet	Stabil
Källa	UR8.5, UR9.1
Verifierbarhet	Kontrollera att dokumentationen ligger där den skall och att dokumentet beskriver den aktuella databasen.

Krav	SR8.6 Webbplatsens struktur
Beskrivning	Dokumentation om hur webbplatsens är strukturerad. Dokumentationen skall omfatta: en webbkarta, vilka webbsidor som använder gemensamma bibliotek/funktioner, hur en sida är uppbyggd med MVC. Dokumentationen skall finnas tillgänglig från administratörens gränssnitt och i källkodsmappen för bokningssystemet.
Behov	Plus
Prioritet	Låg
Stabilitet	Stabil
Källa	UR8.5
Verifierbarhet	Kontrollera att dokumentationen ligger där den skall och att dokumentet beskriver den aktuella webbplatsen.

Krav	SR8.7 Kommenterad källkod
Beskrivning	Den implementerade källkoden skall kommenteras. Funktioner skall beskrivas för framtida vidareutveckling.
Behov	Standard
Prioritet	Låg
Stabilitet	Stabil
Källa	UR8.5
Verifierbarhet	Kontrollera att källkodens funktioner och delar är dokumenterad i källfilen.

3.9 Säkerhetskrav

Krav	SR9.1 Skydd av lösenord
Beskrivning	Lösenord ska inte sparas i klartext i databasen. De bör sparas som saltade SHA-1 lösenord.
Behov	Standard
Prioritet	Hög
Stabilitet	Stabilt
Källa	UR6.1, UR6.2
Verifierbarhet	Verifiera att lösenorden i databasen sparas som saltade SHA-1 strängar

Krav	SR9.2 Användarroller
Beskrivning	Användare ska ges olika roller, som har tillgång till olika delar av systemet
Behov	Standard
Prioritet	Hög
Stabilitet	Stabilt
Källa	UR4.1, UR1.10, UR1.11, UR2.11
Verifierbarhet	Verifiera att en de olika användarrollerna endast kommer åt de delar av systemet som de ska komma åt

Krav	SR9.3 Datasäkerhet
Beskrivning	Alla frågor till databasen ska använda funktioner som förhindrar SQL-injections.
Behov	Standard
Prioritet	Hög
Stabilitet	Stabilt
Källa	UR6.2
Verifierbarhet	Funktionaliteten finns redan i rails, men vi bör verifiera att ev. egenskriva SQL-frågor inte är sårbara

Krav	SR9.4 HTTPS
Beskrivning	HTTPS ska användas för den inloggade delen av systemet och för ev. kortbetalning
Behov	Deluxe
Prioritet	Låg
Stabilitet	Stabilt
Källa	UR6.2
Verifierbarhet	Verifiera att de delar av sidan som använder inloggning eller kortbetalning kräver https

Krav	SR9.5 Förhindra XSS
Beskrivning	För att förhindra XSS bör POST med autentitets-verifiering användas
Behov	Standard
Prioritet	Hög
Stabilitet	Stabilt
Källa	UR6.2
Verifierbarhet	Funktionaliteten finns i rails. Vi bör dock verifiera att vi inte lägger in några egna svagheter

3.10 Portabilitetskrav

Krav	SR10.1 Stöd för CentOS
Beskrivning	Systemet ska ha stöd för att köras och installeras på CentOS.
Behov	Standard
Prioritet	Hög
Stabilitet	Stabilt
Källa	UR9.1, UR9.2, UR7.2, URD 2.3.4
Verifierbarhet	Genom att installera och köra systemet.

3.11 Kvalitetskrav

Krav	SR11.1 Kontroll av dokumentation
Beskrivning	All dokumentation skall ses över av minst en person i Nyx som inte är ansvarig för koden som dokumenterats. Vid eventuella åtgärder meddelas ansvarig för koden. När dokumentationen anses klara kvalitetskraven skriver granskaren sitt namn och datum under kommentarer.
Behov	Plus
Prioritet	Medel
Stabilitet	Stabilt
Källa	UR8.5
Verifierbarhet	Genom att skriva namn och datum i kommentarer efter granskning är det lätt att kontrollera vem som granskat och godkänt dokumentationen samt när detta gjordes.

Krav	SR11.2 Ansvarsfördelning för testkod
Beskrivning	En policy för mjukvarukvalitetsom tydligt hanterar ansvarsfördelning av testkod för vem som har ansvar för att testkod skrivs och körs samt när denna skall vara skriven och köras.
Behov	Standard
Prioritet	Medel
Stabilitet	Instabilt, policyn är ännu inte skriven
Källa	UR8.6
Verifierbarhet	Kontrollera att policy finns att tillgå för alla medlemmar i Nyx.

Krav	SR11.3 Stresstest av systemet
Beskrivning	Testning för den typ av användning systemet skall klara av. Vid felaktigheter analyseras dessa för åtgärd i ett testprotokoll.
Behov	Standard
Prioritet	Medel
Stabilitet	Stabilt
Källa	UR5.1
Verifierbarhet	Testprotokoll med tillhörande resultat finns.

Krav	SR11.4 Användartestning av gränssnitt
Beskrivning	Systemet testas av personer utanför Nyx som faller in under respektive användarkategori. Testarnas åsikter och kommentarer kommer analyseras för att sedan förbättra systemet.
Behov	Standard
Prioritet	Medel
Stabilitet	Stabilt
Källa	UR3.1, UR8.1-4, UR8.6
Verifierbarhet	Testprotokoll med tillhörande resultat finns. Dessa protokoll är tillgängliga efter testfasen med tydliga resultat. Av dessa skall en tydlig åtgärd X finnas om undermåligt resultat Y finns för fråga Z.

Krav	SR11.5 Prestandaövervakning
Beskrivning	Övervakning av prestanda under användartester för att sedan undersöka avvikelser.
Behov	Standard
Prioritet	Hög
Stabilitet	Stabilt
Källa	UR5.1-4
Verifierbarhet	Inga avvikande resultat noterade efter användartester i dessa protokoll.

Krav	SR11.6 Policy för kontroll av kod
Beskrivning	En policy som säger hur och av vem koden skall granskas samt hur detta skall dokumenteras.
Behov	Plus
Prioritet	Medel
Stabilitet	Instabilt, policy ännu inte skriven.
Källa	UR8.6
Verifierbarhet	Den av policyn rekommenderade dokumentationen av granskningen kontrolleras.

3.12 Tillförlitlighetskrav

Krav	SR12.1 Stabilitet
Beskrivning	Systemet ska vara stabilt och bör därför ge inte serverfel oftare än var 1000:e sidladdning. Användarorsakade fel så som InvalidAuthenticityToken räknas ej.
Behov	Standard
Prioritet	Medium
Stabilitet	Stabilt
Källa	UR5.3
Verifierbarhet	Serverfel bör loggas tydligt. Man kan sedan jämföra antalet serverfel med ungefärligt antal sidladdningar för att verifiera detta krav.

3.13 Underhållskrav

Krav	SR13.1 Dokumentation
Beskrivning	Dokumentation ska finnas (och uppdateras vid ändringar) för skrivna Ruby-klasser (däribland Rails-kontroller, -modeller och -mailers) och dess metoder.
Behov	Standard
Prioritet	Låg
Stabilitet	Stabilt
Källa	UR8.5
Verifierbarhet	Granska källkoden till bokningssystemet och verifiera att ingen klass eller metod saknar dokumentation.

Krav	SR13.2 Skrivsätt
Beskrivning	Dokumentation av metoder ska beskriva vad metoden gör i presens samt följa vanliga skrivregler (fullständiga meningar, punkt som avslutning, etc.). Klasser ska dokumenteras på ett sätt som ger en överblick över vad de gör (kontroller) eller beskriver (modeller).
Behov	Plus
Prioritet	Låg
Stabilitet	Stabilt
Källa	UR8.5
Verifierbarhet	Undersök dokumentationen och kontrollera att inga delar av den är formulerade på felaktigt sätt.

3.14 Trygghetskrav

Krav	SR14.1 Databasbackup
Beskrivning	Databasen ska backas upp minst dagligen till en annan disk i en annan lokal så att data aldrig riskerar att gå förlorad
Behov	Standard
Prioritet	Låg
Stabilitet	Stabilt
Källa	UR6.3
Verifierbarhet	Verifiera att backup av databasen är uppsatt korrekt

4 Kravspårningsmatris

Följande tabell listar användarkrav från URD:n samt vilka mjukvarukrav de givit upphov till. För en enklare överblick över vilka användarkrav ett specifikt mjukvarukrav härstammar ifrån, se de specifika kraven i sektion 3.

4.1 Administration

Kravkälla	Krav
UR1.1 Lägg till föreställning	?
UR1.2 Redigera föreställning	?
UR1.3 Enklare statistik	?
UR1.4 Omfattande statistik	?
UR1.5 Filtrera bokningar	?
UR1.6 Grafisk statistik	?
UR1.7 Hantera utskick	?
UR1.8 Sortera bokningar	?
UR1.9 Förhandsgranskning av utskick	?
UR1.10 Ekonomigränssnitt	?
UR1.11 Administratörsgrenssnitt	?
UR1.12 Lägga till teater	?
UR1.13 Lägga till omgång	?
UR1.14 Redigera omgång	?

4.2 Bokning

Kravkälla	Krav
UR2.1 Kontantbetalning från säljarvyn	?
UR2.2 Säljare ska kunna lämna ut biljetter	?
UR2.3 Interaktiv översiktsbild	?
UR2.4 Avbokning	?
UR2.5 Bokning Administratör	?
UR2.6 Boka	?
UR2.7 Färgkodning	?
UR2.8 Kortköp	?
UR2.9 Ändra betalningsstatus	?
UR2.10 Bekräftelsemeddelande	?
UR2.11 Säljargränssnitt	?
UR2.12 Studentbiljetter	?
UR2.13 Utplacering av platser för bokningar	?
UR2.14 Redigering av bokningar	?

4.3 Diverse

Kravkälla	Krav
UR3.1 Möjlighet att navigera i systemen	?
UR3.2 Tidsmätning	?

4.4 Loginsystem

Kravkälla	Krav
UR4.1 Loginsystem	?
UR4.2 Kontohantering	?
UR4.3 Lösenordsgenerator	?

4.5 Prestanda

Kravkälla	Krav
UR5.1 Serverbelastning	?
UR5.2 Svarstid	?
UR5.3 Pålitlighet	?
UR5.4 Serverns uppkopplingshastighet	?

4.6 Säkerhet

Kravkälla	Krav
UR6.1 Autentisering	?
UR6.2 Datasäkerhet	?
UR6.3 Datatrygghet	?

4.7 Miljö

Kravkälla	Krav
UR7.1 Webb	?
UR7.2 Rails	?
UR7.3 Webbläsarkompatibilitet	?
UR7.4 Visuellt webbläsarkompatibilitet	?
UR7.4 Internet Explorer 7	?

4.8 Användbarhet

Kravkälla	Krav
UR8.1 Bokningstid	?
UR8.2 Inlärningsstid	?
UR8.3 Introduktion	?
UR8.4 Förbättring	?
UR8.5 Dokumentation	?
UR8.6 Mjukvarutestning	?

4.9 Externa system

Kravkälla	Krav
UR9.1 MySQL	?
UR9.2 Apache	?
UR9.3 Kortbetalningssystem	?

4.10 Dokumentreferenser

Kravkälla	Krav
URD 2.3.4	?