



---

## **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 0.1

January 31, 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 functionality is to be implemented in the product.

## Ändringslogg

Version	Ändringar
0.1	Första sammanställd version av dokumentet.

## 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

## Dokumentversioner

Dokumentet har genererats från följande deldokument.

**SRD/abstract** version: 2.

**SRD/Ändringslogg** version: 4.

**Gruppmedlemmar** version: 3.

**SRD/Introduktion** version: 2.

**SRD/Introduktion/Syfte** version: 4.

**SRD/Introduktion/Mjukvarans omfattning** version: 4.

**SRD/Introduktion/Definitioner akronymer och förkortningar** version: 28.

**SRD/Introduktion/Källor** version: 2.

**SRD/Introduktion/Dokumentöversikt** version: 4.

**SRD/Allmän beskrivning** version: 3.

**SRD/Allmän beskrivning/Relation till aktuella projekt** version: 4.

**SRD/Allmän beskrivning/Relation till tidigare och kommande projekt** version: 8.

**SRD/Allmän beskrivning/Funktion och syfte** version: 3.

**SRD/Allmän beskrivning/Miljömässiga betänkanden** version: 6.

**SRD/Allmän beskrivning/Relation till andra system** version: 14.

**SRD/Allmän beskrivning/Generella begränsningar** version: 6.

**SRD/Allmän beskrivning/Modellbeskrivning** version: 5.

**SRD/Specifika krav** version: 8.

**SRD/Specifika krav/Funktionalitetskrav** version: 68.

**SRD/Specifika krav/Prestandakrav** version: 20.

**SRD/Specifika krav/Gränssnittskrav** version: 39.

**SRD/Specifika krav/Körbarhetskrav** version: 6.

**SRD/Specifika krav/Resurskrav** version: 5.

**SRD/Specifika krav/Verifikationskrav** version: 14.

**SRD/Specifika krav/Test av användaracceptans** version: 10.

**SRD/Specifika krav/Dokumentationskrav** version: 16.

**SRD/Specifika krav/Säkerhetskrav** version: 14.

**SRD/Specifika krav/Portabilitetskrav** version: *6*.

**SRD/Specifika krav/Kvalitetskrav** version: *19*.

**SRD/Specifika krav/Tillförlitlighetskrav** version: *5*.

**SRD/Specifika krav/Underhållskrav** version: *7*.

**SRD/Specifika krav/Trygghetskrav** version: *8*.

**SRD/Spårningsmatris mellan användarkrav och mjukvarukrav** version: *3*.

**SRD/appendix** version: *1*.

---

## Innehåll

<b>1</b>	<b>Introduktion</b>	<b>8</b>
<b>2</b>	<b>Syfte</b>	<b>9</b>
<b>3</b>	<b>Mjukvarans omfattning</b>	<b>10</b>
<b>4</b>	<b>Definitioner akronymer och förkortningar</b>	<b>11</b>
<b>5</b>	<b>Källor</b>	<b>15</b>
<b>6</b>	<b>Dokumentöversikt</b>	<b>16</b>
<b>7</b>	<b>Allmän beskrivning</b>	<b>17</b>
<b>8</b>	<b>Relation till aktuella projekt</b>	<b>18</b>
<b>9</b>	<b>Relation till tidigare och kommande projekt</b>	<b>19</b>
<b>10</b>	<b>Funktion och syfte</b>	<b>20</b>
10.1	Boka biljetter från hemsidan . . . . .	20
10.2	Registrera betalningar . . . . .	20
10.3	Administrera biljetter . . . . .	20
10.4	Lämna ut bokade biljetter . . . . .	20
10.5	Sälja biljetter direkt . . . . .	20
10.6	Administrera mailutskick . . . . .	20
10.7	Administrera föreställningar och teatrar . . . . .	21
10.8	Administrera konton . . . . .	21
10.9	Statistik . . . . .	21
<b>11</b>	<b>Miljömässiga betänkanden</b>	<b>22</b>
11.1	Användarna . . . . .	22
11.1.1	Kunder . . . . .	22
11.1.2	Säljare . . . . .	22
11.1.3	Administratör och ekonomiansvarig . . . . .	22
11.2	Systemmiljö . . . . .	22
<b>12</b>	<b>Relation till andra system</b>	<b>23</b>
<b>13</b>	<b>Generella begränsningar</b>	<b>24</b>
13.1	Datamodell . . . . .	24
13.2	Resurser . . . . .	24
13.3	Kundbehov . . . . .	24
13.4	Tekniska begränsningar . . . . .	25
<b>14</b>	<b>Modellbeskrivning</b>	<b>26</b>
14.1	Kundgränssnittet . . . . .	26
14.2	Säljargränssnittet . . . . .	26
14.3	Ekonomichefsgränssnittet . . . . .	26
14.4	Admingränssnittet . . . . .	26
14.5	Systemkomponenter . . . . .	26



<b>15 Specifika krav</b>	<b>27</b>
<b>16 Funktionalitetskrav</b>	<b>28</b>
16.1 Funktionalitet för autentisering . . . . .	28
16.2 Funktionalitet för kund . . . . .	28
16.3 Funktionalitet för säljaren . . . . .	29
16.4 Funktionalitet för administratör 1, 9 . . . . .	30
16.5 Funktionalitet för administratör och ekonomiansvarig 1 . . . . .	32
16.6 Funktionalitet som föregående funktionalitet beror på . . . . .	33
16.7 Övrig funktionalitet . . . . .	36
<b>17 Prestandakrav</b>	<b>37</b>
<b>18 Gränssnittskrav</b>	<b>39</b>
18.1 Mjukvaruinterface . . . . .	39
18.1.1 Klient . . . . .	39
18.1.2 Server . . . . .	40
18.2 Kommunikationsinterface . . . . .	40
18.3 Arbetsflödesanalys av användarscenarion . . . . .	40
18.3.1 Kundens GUI . . . . .	41
18.3.2 Säljarens GUI . . . . .	41
18.3.3 Administratör . . . . .	42
18.3.4 Ekonomichef . . . . .	42
18.4 Interfaceprototyper . . . . .	42
18.4.1 Kundens prototyp . . . . .	42
18.4.2 Säljarens prototyp . . . . .	42
18.4.3 Administratörens prototyp . . . . .	42
<b>19 Körbarhetskrav</b>	<b>43</b>
<b>20 Resurskrav</b>	<b>44</b>
<b>21 Verifikationskrav</b>	<b>46</b>
21.1 Test av användaracceptans . . . . .	46
<b>22 Dokumentationskrav</b>	<b>48</b>
22.1 Kund . . . . .	48
22.2 Kårspexet . . . . .	48
22.3 Systemutvecklare . . . . .	49
<b>23 Säkerhetskrav</b>	<b>51</b>
<b>24 Portabilitetskrav</b>	<b>53</b>
<b>25 Kvalitetskrav</b>	<b>54</b>
<b>26 Tillförlitlighetskrav</b>	<b>57</b>
<b>27 Underhållskrav</b>	<b>58</b>
<b>28 Trygghetskrav</b>	<b>59</b>

**29 Spårningsmatris mellan användarkrav och mjukvarukrav**

**60**

## 1 Einführung

## 2 Syfte

Detta dokumentets syfte är att ge projektets utvecklare en överblick av produktens omfattning och mer detaljerad genomgång av dess funktionalitet. Dokumentet är speciellt skrivet för att underlätta för utvecklarna av produkten.

För att underlätta för läsaren så bör mjukvarans omfattning först läsas igenom, tillsammans med de tekniska begreppen som existerar i texten. Först efter detta bör de mer tekniskt detaljerade sektion 2 och sektion 3 gås igenom.

### 3 Mjukvarans omfattning

Produkten 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.

## 4 Definitioner akronymer och förkortningar

*May extend/delete information from URD Section 1.3.*

Alla akronym från URD är medtagna. Om du använder något av akronymen, var snäll och ta bort bindesträcket “ - ” framför akronymet så man vet att det används i SRD.

**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 webserver. Är gratis att använda. [1.3.1]

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

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

**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. [1.3.3]

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

**Firefox** (Mozilla Firefox) \_En gratis 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).

- **Mail** Elektroniskt brev (engelska: email).

- **Mailklient** Datorprogram för att hantera/läsa/skicka mail.

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

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

**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.*

- **Interface** *Se gränssnitt.*

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

**kB** (kilobyte) *Se kbyte.*

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

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

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

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

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

- **Latency** *Engelskt ord för Latens.*

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

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

**Mb** (Megabyte) *Se Mbyte.*

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

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

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

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

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

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

- **Modul** *Term för komponenter eller funktioner som går att separera från resten av systemet*

*och som inte nödvändigtvis krävs för att systemet ska fungera som helhet.*

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

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

- **MVC ramverk** *Mjukvarubibliotek designade efter MVC-konceptet.*

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

**N/A** (Not Applicable) *Inte applicerbart. Används bland annat i tabeller där alla fält förväntas fyllas i men fältet saknar applicerbart värde. (KOLLA DEN HÄR, SKRIV OM? ÄR DET FINNT? /PETER)*

- **Open Source** *Engelskt låneord för öppen källkod.*

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

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

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

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

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

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

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

- **Red Hat Enterprise Linux** *Variant av Linux.*

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

**Ruby** *Ett objektorienterat programmeringsspråk.*

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

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

- **Spex** (Spektakel) *Humoristisk studentamatörteaterföreställning.*

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

**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*

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

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

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

- **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.*

**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*

- **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.*

## 5 Källor

*May extend/delete information from URD Section 1.4*

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].

### **Apache Software Foundation**

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

Hänvisning till källan görs från referenserna: [1.3.2].

### **CentOS**

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

Hänvisning till källan görs från referenserna: [1.3.3].

## 6 Dokumentöversikt

*Similar to URD Section 1.5, but describes the SRD. However, it need not be assumed that readership on the customer side exists. In practise, this document may be internal and confidential to the development team.*

Detta dokument innehåller fyra distinkta delar. Sektion 1 är denna introduktion till projektet. Sektion 2 visar produktens funktionalitet och presenterar dessutom de externa faktorer kring produkten. Sektion 2.7 är extra intressant för utvecklare, ty den innehåller logiska datamodeller och användarscenarion (use cases). Sektion 3 beskriver de specifika krav från kunden såväl från Nyx som finns på produkten. Sektion 4 innehåller en matris där mjukvarans krav kan spåras tillbaka till användarens krav från URDn.

## **7 Allmän beskrivning**

## 8 Relation till aktuella projekt

*Describes the relationship with other current projects (either customer side or developer side). Customer side could be outsourced component of a larger project. Developer side could be related to similar development work allowing synergies in work, staff sharing, component re-use, software re-use, etc.*

Vårt projekt relaterar till Kårspexets hemsida och ska byta ut det gamla bokningssystemet som finns där. Till vår kännedom är det ingen annan som gör några större ändringar på Kårspexets hemsida eller det gamla bokningssystemet.

Vi har heller inga aktuella projekt som någon i gruppen arbetar med som relaterar till Kårspexet.

## 9 Relation till tidigare och kommande projekt

*Relation to predecessor and successor projects. Describes the relationship with past and future projects (either customer side or developer side). Similar to Section 2.1 in the SRD, but looking outside the present time.*

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

Kårspexet har uttryckligen bett om att det nya bokningssystemet ska passa in i deras gamla hemsida utan att designen krockar för mycket. Det innebär att kundens gränssnitt behöver vara anpassat till Kårspexets nuvarande hemsida. De andra gränssnitten behöver inte anpassas.

Victor 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. T.ex. validering av input med javascript och navigation mellan bokningssteg.

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

## 10 Funktion och syfte

*Describes the main functions the product must perform, gives an overview. (Details are set out in Section 3.) Takes a developer-centric approach. Similar to Section 2.2 in the URD.*

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.

### 10.1 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.

### 10.2 Registrera betalningar

Ekonomichefen ska kunna registrera betalningar för bokningar som kunder gjort.

### 10.3 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.

### 10.4 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.

### 10.5 Sälja biljetter direkt

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

### 10.6 Administrera mailutskick

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

### 10.7 Administrera föreställningar och teatrar

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

### 10.8 Administrera konton

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

### 10.9 Statistik

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



## 11 Miljömässiga betänkanden

*Environmental considerations. Describes where the product will be used (business environment and/or geographical location), who will use it (job roles, skill levels), who will operate and maintain it, hardware it will run on, operating system required.*

### 11.1 Användarna

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

#### 11.1.1 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 webb, men att de kan ta lång tid på sig att anpassa sig till nya gränssnitt. Kunderna kommer att använda systemet bara enstaka gånger. Det är därför viktigt att kundens gränssnitt är enkelt, även första gången man använder det.

#### 11.1.2 Säljare

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

#### 11.1.3 Administratör och ekonomiansvarig

Både administratören och ekonomiansvarige är studenter på en teknisk högskola, i Kårspexets fall KTH. De är därmed vana att navigera i personliga inloggningssidor, t.ex. Mina sidor, eller studera.nu. De är även vana användare av mailklienter.

De kommer använda systemet ofta. Det är acceptabelt att viss utbildning krävs. Användarvänlighetskraven på administratörens gränssnitt är relativt låga.

### 11.2 Systemmiljö

Systemet kommer att köras på en 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.

## 12 Relation till andra system

*Relation to other systems. Describes related external systems and subsystems. (A revision of URD Section 2.1).*

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

## 13 Generella begränsningar

*General constraints. Describes the main constraints that apply and why they exist. (A revision of URD Section 2.3).*

### 13.1 Datamodell

Datamodellen finns bifogad med beskrivning i appendix.

### 13.2 Resurser

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

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

### 13.3 Kundbehov

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

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

### 13.4 Tekniska begränsningar

Kårspexet står för den server som kommer köra vår webbapplikation. Vi har ingen kontroll över deras hårdvara, men vi har verifierat att operativsystemet som körs på servern är kompatibelt med Apache, Ruby on Rails [2.6.1] och MySQL.

Applikationen kommer inte inkludera avancerade algoritmer utan till störst del involveras mycket

trafik till och från databasen. I och med att webbapplikationen och databasen körs på en och samma dator undviks eventuella begränsningar i nätverksanslutningen.

Systemet kommer enligt våra uppskattningar exponeras för upp till åtta samtidiga användare.

Rails under Apache kommer i detta fall att, under godtycklig tidpunkt, använda uppskattningsvis

c:a 250 MB systemminne [2.6.2]. CentOS anger 256 mb minne samt en klockfrekvens

på minst 500 MHz som minimikrav för datorer som kör operativsystemet [2.6.3]. Med MySQL

och Apache körandes samtidigt utöver dessa bör servern ha minst en gigabyte systemminne och

en processor med klockfrekvensen 1 GHz eller högre. Kårspexets server har en processor med

klockfrekvensen 2,6 GHz och 1 Gb systemminne, vilket alltså bör vara tillräckligt.

## 14 Modellbeskrivning

*Model description. Describes the logical or conceptual model using a recognized (i.e. standardized) systems analysis method. Provides a top-level description of the logical model. (Details can be presented in Section 3.) This could be for example the results of an object-oriented analysis of the user requirements from the URD using UML, with a data dictionary, actor identification, use case analysis and an object model/class diagrams. May also include other kinds of model, such as state machines, collaboration diagrams, Petri net models, flow diagrams, business process analysis, abstract data type models, formal specifications, etc, etc.*

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

### 14.1 Kundgränssnittet

!kund-usecase.jpg!

### 14.2 Säljargränssnittet

!saljare-usecase.jpg!

### 14.3 Ekonomichefsgränssnittet

!ekonomi-usecase.jpg!

### 14.4 Admingränssnittet

!admin-usecase.jpg!

### 14.5 Systemkomponenter

!skiss-bokningssystem.jpg!

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].

## 15 Specifika krav

<b>Krav</b>	<i>.SRX.Y Lämplig rubrik (exempel: krav 5 under 3.6 har ID SR6.5)</i>
<b>Beskrivning</b>	
<b>Behov</b>	Essential software requirements <b>shall</b> be marked as such. Essential software requirements are non-negotiable; others may be less vitally important and subject to negotiation.
<b>Prioritet</b>	For incremental delivery, each software requirement <b>shall</b> include a measure of priority so that the developer can decide the production schedule.
<b>Stabilitet</b>	Some requirements may be known to be stable over the expected life of the software; others may be more dependent on feedback from the design phase, or may be subject to change during the software life cycle. Such unstable requirements <b>should</b> be flagged.
<b>Källa</b>	References that trace software requirements back to the URD <b>shall</b> accompany each software requirement.

|\_.Verifierbarhet| Each software requirement shall be verifiable. This means that it must be possible to:

- a) check that the requirement has been incorporated in the design;
- b) prove that the software will implement the requirement;
- c) test that the software does implement the requirement.

**Klarhet:** A requirement is clear if it has one, and only one, interpretation. Clarity implies lack of ambiguity. If a term used in a particular context has multiple meanings, the term should be qualified or replaced with a more specific term.

## 16 Funktionalitetskrav

*Functional requirements. Describes each logical component in terms of functional behaviour.*

*3.1 Functional requirements. What each logical component does.*

*Kravens nummer (SR1, SR2, ...) är ej slutgiltiga, det finns luckor*

*Ta ej bort de gamla siffrorna (1, 2, 3, ...), kraven refererar till varandra*

### 16.1 Funktionalitet för autentisering

<b>Krav</b>	<b>SR1.1 Autentisering</b>
<b>Beskrivning</b>	Säljarens, administratörens och ekonomichefens funktionaliteter ska endast kunna användas av autentiserade användare. Autentisering sker genom inloggning med användarnamn och ett lösenord. Den inloggade användaren ska kunna logga ut, dvs att upphäva sin behörighet tills nästa inloggning. För att uppfylla detta behöver systemet hantera användarkonton med olika behörigheter.
<b>Motivering</b>	För att hindra att obehöriga får tillgång till systemet. <i>Hänvisa till annat krav på autentisering?</i>
<b>Behov</b>	Standard
<b>Prioritet</b>	Låg (Hög ?)
<b>Stabilitet</b>	Stabilt
<b>Källa</b>	URD4.1
<b>Verifierbarhet</b>	Kontrollera att det dokumenterats att autentisering har implementerats. Kontrollera att systemet hänvisar användaren till inloggning.

### 16.2 Funktionalitet för kund

<b>Krav</b>	<b>SR1.3 Kund: boka, 3, 4, 6, 11, 12, 7, 10?</b>
<b>Beskrivning</b>	En kund ska kunna göra en bokning så som beskrivs i stycke <i>just nu i Gränssnittskrav</i> . Detta inkluderar att välja föreställning, sektion,... och att få en bokningsbekräftelse både som webbsida och som mail.
<b>Motivering</b>	Detta hör till systemets huvuduppgifter.
<b>Behov</b>	Standard
<b>Prioritet</b>	Hög
<b>Stabilitet</b>	Stabilt
<b>Källa</b>	UR2.6
<b>Verifierbarhet</b>	

<b>Krav</b>	<b>SR1.4 Kund: visa/avboka bokning, 13, 6, 7?</b>
<b>Beskrivning</b>	Kunden ska kunna visa eller avboka sin bokning, från och med att bokningen görs och minst fram tills föreställningens början/slut(?). Förslagsvis kan kunden göra detta via en länk i bokningens bekräftelsemail.
<b>Motivering</b>	För att tillåta avbokningar.
<b>Behov</b>	Standard
<b>Prioritet</b>	Låg
<b>Stabilitet</b>	Stabilt
<b>Källa</b>	UR2.4
<b>Verifierbarhet</b>	

### 16.3 Funktionalitet för säljaren

<b>Krav</b>	<b>SR1.5 Säljare: nytt kontantköp 3, 5, 6, 11</b>
<b>Beskrivning</b>	En säljare ska kunna genomföra ett kontantköp i enlighet med <i>vad?</i> . Detta inkluderar att skapa ny bokning, välja sittplatser, registrera betalning och registrera biljetterna som uthämtade.
<b>Motivering</b>	Kårspexet vill kunna stå på offentliga platser och sälja och lämna ut biljetter till kunder.
<b>Behov</b>	Standard
<b>Prioritet</b>	Medel
<b>Stabilitet</b>	Stabil
<b>Källa</b>	UR2.1
<b>Verifierbarhet</b>	

<b>Krav</b>	<b>SR1.6 Säljare: lämna ut bokade biljetter, 2, 5, 6</b>
<b>Beskrivning</b>	Säljaren ska kunna lämna ut biljetterna en bokning, så som beskrivs i <i>referera</i> . Detta innebär, förutom att registrera biljetternas utlämnande, eventuellt att välja sittplatser och att ta betalt.
<b>Motivering</b>	Kårspexet vill kunna stå på offentliga platser och sälja och lämna ut biljetter till kunder.
<b>Behov</b>	Standard
<b>Prioritet</b>	Medel
<b>Stabilitet</b>	Stabil
<b>Källa</b>	UR2.2
<b>Verifierbarhet</b>	

<b>Krav</b>	<b>SR1.7 Säljare: ändra bokning</b>
<b>Beskrivning</b>	Säljaren ska kunna visa/ändra vissa av en boknings uppgifter, vilka? (?)
<b>Motivering</b>	
<b>Behov</b>	
<b>Prioritet</b>	
<b>Stabilitet</b>	
<b>Källa</b>	
<b>Verifierbarhet</b>	



## 16.4 Funktionalitet för administratör 1, 9

<b>Krav</b>	<b>SR1.9 Administratör: hantera teatrar 14</b>
<b>Beskrivning</b>	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 ska 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.
<b>Motivering</b>	Teatrar behövs för att hantera omgångar på olika teatrar med olika uppsättningar stolar.
<b>Behov</b>	Standard
<b>Prioritet</b>	Hög(?)
<b>Stabilitet</b>	Stabilt
<b>Källa</b>	UR1.12
<b>Verifierbarhet</b>	

<b>Krav</b>	<b>SR1.10 Administratör: hantera omgångar 15, 16</b>
<b>Beskrivning</b>	Administratören ska kunna lägga till, ta bort och ändra omgångar och deras föreställningar.
<b>Motivering</b>	Omgångar behöver hanteras av systemet.
<b>Behov</b>	Standard
<b>Prioritet</b>	Hög
<b>Stabilitet</b>	Stabilt
<b>Källa</b>	UR1.13, UR1.14
<b>Verifierbarhet</b>	

<b>Krav</b>	<b>SR1.11 Administratör: mailutskick 7, 10, 6, 2</b>
<b>Beskrivning</b>	Administratören ska kunna söka upp bokningar och göra ett mailutskick, baserat på en mail-mall. Administratören ska också kunna skapa nya mail-mallar.
<b>Motivering</b>	För att smidigt kunna göra mailutskick till kunderna.
<b>Behov</b>	Standard
<b>Prioritet</b>	Medel
<b>Stabilitet</b>	Stabilt
<b>Källa</b>	UR1.7
<b>Verifierbarhet</b>	

<b>Krav</b>	<b>SR1.12 Administratör: hantera konton 18, 6</b>
<b>Beskrivning</b>	Administratören ska kunna lägga till och ta bort användarkonton, samt ändra deras lösenord, kontotyp(behörighet) och giltighetstid.
<b>Motivering</b>	Konton behövs för autentisering, SR autentisering
<b>Behov</b>	Standard
<b>Prioritet</b>	Medel
<b>Stabilitet</b>	Stabilt
<b>Källa</b>	UR4.2
<b>Verifierbarhet</b>	

<b>Krav</b>	<b>SR1.12 Administratör: Lösenordsgenerator</b>
<b>Beskrivning</b>	Administratören ska kunna välja ett automatiskt genererat lösenord vid kontohanteringen (SR Administratör: konton).
<b>Motivering</b>	För att lösenorden ska bli säkra och för att förenkla administratörens arbete.
<b>Behov</b>	Deluxe
<b>Prioritet</b>	Medel
<b>Stabilitet</b>	Stabilt
<b>Källa</b>	UR4.3
<b>Verifierbarhet</b>	

<b>Krav</b>	<b>SR1.13 Administratör: placera bokningar 2, 5, 13, 7, 6, 10?</b>
<b>Beskrivning</b>	Administratören ska kunna välja sittplatser åt bokningar.
<b>Motivering</b>	Detta är ett grundläggande steg i hanteringen av bokningar, enligt Kårspexets önskemål.
<b>Behov</b>	Standard
<b>Prioritet</b>	Medel
<b>Stabilitet</b>	Stabilt
<b>Källa</b>	UR2.13
<b>Verifierbarhet</b>	

<b>Krav</b>	<b>SR1.14 Administratör: hantera bokningar 2, 13, 6, 11, 12</b>
<b>Beskrivning</b>	Administratören ska kunna ändra eller lägga till bokningar. Bokningar ska kunna markeras som gratis.
<b>Motivering</b>	För att kunna rätta till misstag manuellt. För att kunna hantera gratisbiljetter.
<b>Behov</b>	Standard
<b>Prioritet</b>	Medel
<b>Stabilitet</b>	Stabilt
<b>Källa</b>	UR2.5, UR2.14
<b>Verifierbarhet</b>	

## 16.5 Funktionalitet för administratör och ekonomiansvarig 1

<b>Krav</b>	<b>SR1.15 Administratör och ekonomiansvarig: översikt, 2</b>
<b>Beskrivning</b>	En sida som visar kortfattad statistik. Vad? antal bokningar i olika tillstånd.
<b>Motivering</b>	
<b>Behov</b>	
<b>Prioritet</b>	
<b>Stabilitet</b>	
<b>Källa</b>	
<b>Verifierbarhet</b>	

<b>Krav</b>	<b>SR1.16 Administratör och ekonomiansvarig: Statistik</b>
<b>Beskrivning</b>	Statistik om bokningarna ska visas. Detta ska inkludera, antal bokade respektive utgivna biljetter per föreställning, omgång och spelår och rabattklass.
<b>Motivering</b>	Att ge admin och ekochef överblick och hjälpa dem att planera verksamheten nu och i framtiden.
<b>Behov</b>	Standard
<b>Prioritet</b>	Låg
<b>Stabilitet</b>	Stabilt
<b>Källa</b>	UR1.3
<b>Verifierbarhet</b>	

<b>Krav</b>	<b>SR1.16b Administratör och ekonomiansvarig: Statistik Plus</b>
<b>Beskrivning</b>	Mer utförlig statistik än SR Statistik. Detta inkluderar, statistik över tidsintervall, bokningar beroende på föreställnings start-/sluttid, tidpunkter för biljettutlämningar.
<b>Motivering</b>	Se SR Statistik
<b>Behov</b>	Plus
<b>Prioritet</b>	Medel
<b>Stabilitet</b>	Stabilt
<b>Källa</b>	UR1.4
<b>Verifierbarhet</b>	

<b>Krav</b>	<b>SR1.16c Administratör och ekonomiansvarig: Statistik Deluxe</b>
<b>Beskrivning</b>	Utöver SR Statistik ska statistiken framställas grafiskt.
<b>Motivering</b>	Se SR Statistik, för att göra det enklare att förstå statistiken.
<b>Behov</b>	Deluxe
<b>Prioritet</b>	Medel
<b>Stabilitet</b>	Stabilt
<b>Källa</b>	UR1.6
<b>Verifierbarhet</b>	

<b>Krav</b>	<b>SR1.17 Administratör och ekonomiansvarig: registrera betalningar, 2</b>
<b>Beskrivning</b>	Administratören och ekonomiansvarig ska kunna registrera betalningar.
<b>Motivering</b>	
<b>Behov</b>	Standard
<b>Prioritet</b>	Medel
<b>Stabilitet</b>	Stabilt
<b>Källa</b>	UR Ändra betalningsstatus (#28)
<b>Verifierbarhet</b>	

## 16.6 Funktionalitet som föregående funktionalitet beror på

<b>Krav</b>	<b>SR1.18 2. Sök bokningar</b>
-------------	--------------------------------

|\_..Beskrivning|Användaren ska kunna välja ut bokningar med hjälp av filtrering och sortering. Följande egenskaper hos bokningarna ska kunna användas: ...

Sökningen ska anpassas till de olika användarna och användaren ska kunna förfinas ytterligare genom att visa olika egenskaper, och välja olika filter och sorteringar.|

<b>Motivering</b>	Behövs av SR ...
<b>Behov</b>	Standard
<b>Prioritet</b>	Hög
<b>Stabilitet</b>	Stabilt
<b>Källa</b>	UR1.8
<b>Verifierbarhet</b>	

<b>Krav</b>	<b>SR1.19 3. Föreställningsväljare</b>
<b>Beskrivning</b>	Det behövs funktionalitet som låter användaren välja en föreställning i en aktuell omgång. Kund, säljare och administratör behöver detta.
<b>Motivering</b>	Vid skapandet av en bokning måste föreställning väljas.
<b>Behov</b>	Standard
<b>Prioritet</b>	Medel
<b>Stabilitet</b>	Stabilt
<b>Källa</b>	URD #2, URD #12, URD #19
<b>Verifierbarhet</b>	Verifiera att de funktioner som är beroende av denna funktion. SR1.X ...

<b>Krav</b>	<b>SR1.20 4. Sektionsplacering</b>
<b>Beskrivning</b>	En bild av teaterns sektioner visas. Eventuellt kan användaren välja en sektion. Hur "ledig" en sektion är visas med färger.
<b>Motivering</b>	För att låta kunden välja sektion vid en bokning. För att låta administratören och säljaren placera ut en bokning på en plats i den sektion kunden valt.
<b>Behov</b>	Standard
<b>Prioritet</b>	Medel
<b>Stabilitet</b>	Stabilt
<b>Källa</b>	URD #2, URD #20
<b>Verifierbarhet</b>	

<b>Krav</b>	<b>SR1.21 5. Platsplacering</b>
<b>Beskrivning</b>	Visa eller välj en mängd platser (stolar) i en teater. Visa upptagna platser vid en viss föreställning.
<b>Motivering</b>	För att säljare eller administratör ska kunna välja platser åt bokningarna.
<b>Behov</b>	Standard
<b>Prioritet</b>	Medel
<b>Stabilitet</b>	Stabil
<b>Källa</b>	
<b>Verifierbarhet</b>	

<b>Krav</b>	<b>SR1.22 6. Bekräftelse</b>
<b>Beskrivning</b>	Användaren visas en sida med data och kan välja att bekräfta att uppgifterna är korrekta. Datan visas som godtycklig HTML-kod (?).
<b>Motivering</b>	Bekräftelser ska användas vid de flesta inmatningar, för att säkerställa att korrekta uppgifter ges.
<b>Behov</b>	Standard
<b>Prioritet</b>	Medel
<b>Stabilitet</b>	
<b>Källa</b>	
<b>Verifierbarhet</b>	

<b>Krav</b>	<b>SR1.23 7. Mail-skickare</b>
<b>Beskrivning</b>	Givet en mängd bokningar och en mailmall skapas meddelanden och skickas till de epostadresser som hör till bokningarna.
<b>Motivering</b>	För att kunna skicka mejl till kunderna, automatiskt eller manuellt. Se SR Administratör: mailutskick, SR Kund: Boka.
<b>Behov</b>	Standard
<b>Prioritet</b>	Låg
<b>Stabilitet</b>	Stabilt
<b>Källa</b>	UR1.7
<b>Verifierbarhet</b>	

<b>Krav</b>	<b>SR1.24 10. Mail-mallar</b>
-------------	-------------------------------

|\_.Beskrivning|I databasen finns mailmallar. Uppgifter om en bokning kan sättas in i mallen. Mailen är avsedda för kunderna. Mailet adressera

s till den epost-adress som förknippas med bokningen.

Följande uppgifter går att lägga genom mallen: namn, bokningsnummer, föreställningens datum, bokade platser, betalstatus, ...|

<b>Motivering</b>	Se SR Mail-skickare
<b>Behov</b>	Standard
<b>Prioritet</b>	Medel
<b>Stabilitet</b>	Stabilt
<b>Källa</b>	UR1.7
<b>Verifierbarhet</b>	

<b>Krav</b>	<b>SR1.25 11. Skapa bokning</b>
<b>Beskrivning</b>	Skapar en ny bokning i databasen. Följande data krävs ...
<b>Motivering</b>	
<b>Behov</b>	
<b>Prioritet</b>	
<b>Stabilitet</b>	
<b>Källa</b>	
<b>Verifierbarhet</b>	

<b>Krav</b>	<b>SR1.26 13. Ändra bokning</b>
<b>Beskrivning</b>	Ändrar uppgifter i en bokning. Behörighet krävs (?).
<b>Motivering</b>	
<b>Behov</b>	
<b>Prioritet</b>	
<b>Stabilitet</b>	
<b>Källa</b>	
<b>Verifierbarhet</b>	

<b>Krav</b>	<b>SR1.27 9. Menyer</b>
<b>Beskrivning</b>	Menyn ska hjälpa administratören (och ekonomichefen och säljaren?) att ta sig till de olika webbsidorna/funktionerna.
<b>Motivering</b>	

|\_.Behov||

<b>Prioritet</b>	
<b>Stabilitet</b>	
<b>Källa</b>	UR navigationsmöjlighet (#13)
<b>Verifierbarhet</b>	
<b>Krav</b>	<b>SR1.28 12. Progress bar</b>
<b>Beskrivning</b>	Visar en “progress bar”, en överblick över en serie handlingar som användaren utför.
<b>Motivering</b>	
<b>Behov</b>	
<b>Prioritet</b>	
<b>Stabilitet</b>	
<b>Källa</b>	
<b>Verifierbarhet</b>	

## 16.7 Övrig funktionalitet

<b>Krav</b>	<b>SR1.29 Tidsmätning</b>
<b>Beskrivning</b>	För varje anrop till systemet, ska tiden det tar att svara mätas. Sedan ska tiden skrivas till en logg-fil.
<b>Motivering</b>	För att verifiera att SR2.5 Maximal svarstid uppfylls.
<b>Behov</b>	Standard.
<b>Prioritet</b>	Låg.
<b>Stabilitet</b>	Stabilt.
<b>Källa</b>	UR3.2
<b>Verifierbarhet</b>	Gör ett eller flera anrop och kontrollera att uppmätta tider står i loggen.

## 17 Prestandakrav

*Performance requirements. Describes specific performance requirements on components such as: time, space, load, reliability etc.*

Krav	SR 2.1 Trafikmängd för en bokning
Beskrivning	Den maximala datatrafikmängd som får krävas för att en bokning skall kunna genomföras är 3,125 MiBit. Trafikmaximumet är räknat i en riktning (ex. server till klient).
Behov	Standard
Prioritet	Medel
Stabilitet	Stabilt så länge informationsinnehållet i bokningsprocessen inte utökas.
Källa	UR5.1, UR5.2
Verifierbarhet	Mäta trafiken i båda riktningar mellan klienten och servern då en bokning.

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 (ett första anrop och 10 nominella anrop).
Behov	Standard
Prioritet	Medel
Stabilitet	Stabilt så länge flödet för bokningsprocessen inte ändras.
Källa	UR5.1, UR5.2
Verifierbarhet	Mäta antalet förfrågningar som görs mot server då en bokning utförs.

Krav	SR 2.3 Största sidladdningen
Beskrivning	Den största sidladdningen för bokningsprocessen i datatrafik räknat får inte vara större än 100KiB.
Behov	Standard
Prioritet	Medel
Stabilitet	Stabilt så länge informationsinnehållet i bokningsprocessen inte utökas.
Källa	UR5.1, UR5.2
Verifierbarhet	Mäta och observera den största sidladdningen då en bokning utförs.

Krav	SR 2.4 Nominell sidladdning
Beskrivning	Medelvärde för de nominella anropen under bokningsprocessen i datatrafik räknat får inte överstiga 30KiB.
Behov	Standard
Prioritet	Medel
Stabilitet	Stabilt så länge informationsinnehållet i bokningsprocessen inte utökas.
Källa	UR5.1, UR5.2
Verifierbarhet	Mäta och observera medelvärde för alla anrop, exklusivt det största, då en bokning utförs.



Krav	SR 2.5 Maximal svarstid
Beskrivning	Bokningssystemet får inte ha en svarstid till klienten som är större än 1 sekund.
Behov	Standard
Prioritet	Medel
Stabilitet	Stabil
Källa	URD #31
Verifierbarhet	Under ett stresstest motsvarande maxintensiteten för bokningsanrop mäta eller observera tiden för ett anrop. Webbserverns loggfil är hjälpsam.

## 18 Gränssnittskrav

*Interface requirements. Proposal for a user interface structure, including information structure and organisation, workflow analysis for certain use cases, design philosophy including any ergonomic issues. Presentation should include prototype designs.*

**Victor:** Resurskrav är ganska likt delar av denna punkt, vi borde typ diskutera vem som tar vad.

\_Interface requirements. These specify hardware, software or database elements with which the system, or system component, must interact or communicate. Interface requirements should be classified into software, hardware and communications interfaces. Software interfaces could include operating systems, software environments, file formats, database management systems and other software applications. Hardware

interface requirements may specify the hardware configuration. Communications interface requirements constrain the nature of the interface to other hardware and software. They may demand the use of a particular network protocol, for example. External interface requirements should be described, or referenced in ICDs . User interface requirements should be specified under ‘Operational Requirements’ (see below). Interface requirements can be illustrated with system block diagrams (e.g. to show the hardware configuration).\_\_

<b>Krav</b>
<b>Beskrivning</b>
<b>Motivering</b>
<b>Behov</b>
<b>Prioritet</b>
<b>Stabilitet</b>
<b>Spår</b>
<b>Källa</b>
<b>Verifierbarhet</b>

### 18.1 Mjukvaruinterface

#### 18.1.1 Klient

<b>Krav</b>	<b>SR3.1 Kundens GUI</b>
<b>Beskrivning</b>	Kårspexets bokningssystem ska stödja följande webbläsare; <i>Firefox 3</i> , Internet Explorer 8.
<b>Motivering</b>	Dessa webbläsare är stora på marknaden just nu och bör stödas av vårt system. Vi refererar till UR 1.4 för statistik på webbläsaranvändande.
<b>Behov</b>	Standard
<b>Prioritet</b>	Låg
<b>Stabilitet</b>	Instabilt
<b>Spår</b>	UR7.3
<b>Källa</b>	Nyx
<b>Verifierbarhet</b>	Provkörning av systemet i dessa webbläsare.

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

### 18.1.2 Server

<b>Krav</b>	<b>SR3.3 Serverinterface</b>
<b>Beskrivning</b>	Apache används som webbserver för att göra bokningssystemet tillgängligt via HTTP-protokollet.
<b>Motivering</b>	Apache möjliggör åtkomst till Kårspexets bokningssystem på internet via HTTP-protokollet.
<b>Behov</b>	Standard
<b>Prioritet</b>	Hög
<b>Stabilitet</b>	Stabilt
<b>Spår</b>	UR9.2
<b>Källa</b>	Nyx
<b>Verifierbarhet</b>	Visa att Kårspexets server kör Apache.

## 18.2 Kommunikationsinterface

<b>Krav</b>	<b>SR3.4 Databasinterface</b>
<b>Beskrivning</b>	Webbservern kommunicerar med MySQL-databasen via Rails med hjälp av Unix-sockets.
<b>Motivering</b>	Bokningssystemet behöver kommunicera med databasen.
<b>Behov</b>	Standard
<b>Prioritet</b>	Hög
<b>Stabilitet</b>	Stabilt
<b>Källa</b>	UR9.1
<b>Verifierbarhet</b>	Verifiera webbservern kommunicerar med databasen.

## 18.3 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.

### 18.3.1 Kundens GUI

Kundens GUI består av en bokningssida på Kårspexets hemsida. 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.

!kundInterfaceStruktur.jpg!

### 18.3.2 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.

!saljarInterfaceStruktur.jpg!

#### Kontant köp

Kontant köp innebär att kunder står i kö för att köpa en biljett direkt från Kårspexet, istället för att boka sin biljett på Kårspexets hemsida 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.

### 18.3.3 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 henom 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.

!adminInterfaceStruktur.jpg!

### 18.3.4 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.

!ekonomiInterfaceStruktur.jpg!

## 18.4 Interfaceprototyper

### 18.4.1 Kundens prototyp

!kundSRD.jpg!

### 18.4.2 Säljarens prototyp

!säljarPlaceringv1.jpg!

### 18.4.3 Administratörens prototyp

!adminBokning.jpg!

## 19 Körbarhetskrav

<b>Krav</b>	<b>SR 4.1 Servern måste köra Ruby on Rails</b>
<b>Beskrivning</b>	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.
<b>Behov</b>	Standard
<b>Prioritet</b>	Hög
<b>Stabilitet</b>	Stabilt
<b>Källa</b>	UR7.2
<b>Verifierbarhet</b>	Det går att starta applikationen och komma åt den från en webbläsare.

<b>Krav</b>	<b>SR 4.2 Mjukvaran ska kunna köras på apache</b>
<b>Beskrivning</b>	Vår mjukvara ska gå att använda tillsammans med Apache. Denna lösning kommer vi använda på Kårspexets server.
<b>Behov</b>	Standard
<b>Prioritet</b>	Hög
<b>Stabilitet</b>	Stabilt
<b>Källa</b>	UR9.2
<b>Verifierbarhet</b>	Test av mjukvaran på en server som kör Apache som webserver. Om man kan besöka sidan via en webbläsare är kravet uppfyllt.

<b>Krav</b>	<b>SR 4.3 Servern måste köra MySQL</b>
<b>Beskrivning</b>	Databasmjukvaran MySQL måste finnas på servern för att vår mjukvara ska fungera.
<b>Behov</b>	Standard
<b>Prioritet</b>	Hög
<b>Stabilitet</b>	Stabilt
<b>Källa</b>	UR9.1
<b>Verifierbarhet</b>	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.

## 20 Resurskrav

*Resource requirements. Platform, OS, network, browser requirements, etc.*

**Victor:** Är detta krav som vi ställer på all användning av vår produkt eller är det någon form av begränsande krav typ “med dessa resurser måste det fungera”?

ingen motsvarighet i URD, kårspexet har inte heller explicit sagt detta utan detta är något implicit. Inkludera ändå?

Krav	SR5.1 Tillgänglig webbserver
<b>Beskrivning</b>	Bokningssystemet körs på en server som är åtkomstbar för kunder såväl som säljare och andra administratörer.
<b>Motivering</b>	
<b>Behov</b>	Standard
<b>Prioritet</b>	Normal
<b>Stabilitet</b>	Stabilt
<b>Källa</b>	UR7.3

|\_.Verifierbarhet|Each software requirement shall be verifiable. This means that it must be possible to:

- a) check that the requirement has been incorporated in the design;
- b) prove that the software will implement the requirement;
- c) test that the software does implement the requirement.|

Krav	SR5.2 Webbläsare
<b>Beskrivning</b>	Användning av bokningssystemet görs i någon av de webbläsare vi valt att stödja; Firefox 3 samt Internet Explorer 8.
<b>Motivering</b>	
<b>Behov</b>	Standard
<b>Prioritet</b>	Normal
<b>Stabilitet</b>	Stabilt
<b>Källa</b>	UR7.3

|\_.Verifierbarhet|Each software requirement shall be verifiable. This means that it must be possible to:

- a) check that the requirement has been incorporated in the design;
- b) prove that the software will implement the requirement;
- c) test that the software does implement the requirement.|

<b>Krav</b>	<b>SR5.3 Serverns prestanda</b>
<b>Beskrivning</b>	Kravet på serverns hårdvara omfattande processor och arbetsminne som en helhet. Hårdvaran måste klara av att hantera 1,47 anrop/sekund. Det motsvarar 8 samtidiga användare.
<b>Behov</b>	Standard
<b>Prioritet</b>	Medel
<b>Stabilitet</b>	Stabil
<b>Källa</b>	UR31
<b>Verifierbarhet</b>	Stresstesta servern med flera användare och observera processor- och minnesanvändningen i aktivitetshanterare på servern.

<b>Krav</b>	<b>SR5.4 Serverns uppkopplingshastighet</b>
<b>Beskrivning</b>	Kravet på serverns uppkopplingshastighet mot internet. Servern måste ha en uppkopplingshastighet på 1,15 MiBit/sekund.
<b>Behov</b>	Standard
<b>Prioritet</b>	Medel
<b>Stabilitet</b>	Stabil
<b>Källa</b>	UR31
<b>Verifierbarhet</b>	Stresstesta servern med flera användare och observera nätverkstrafiken i aktivitetshanterare på servern.



## 21 Verifikationskrav

Krav	SR6.1 Testkod
Beskrivning	För att testa att serverns operationer utförs korrekt.
Motivering	Kårspeket ska kunna förlita sig på programvarans korrekthet.
Behov	Normal.
Prioritet	Standard.
Stabilitet	Stabilt.
Källa	N/A.
Verifierbarhet	Nyx testar manuellt att vår testkod ger tillbaka förväntat svar.

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

*Verification requirements. Plan and methods for verifying and validating the system against the SRD based on user evaluation, testing and (if necessary) formal verification.*

### 21.1 Test av användaracceptans

*Acceptance testing requirements. Plan and methods for verifying that the final system meets the end-user requirements as specified by the URD.*

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	Hög
Prioritet	Låg
Stabilitet	Stabilt
Spår	N/A
Källa	Internt/Nyx
Verifierbarhet	Hitta ifyllda testprotokoll för kund-, säljar-, ekonomi- samt administratörsgränssnittet och verifiera att de överensstämmer med de krav som ställs i URD:n.

<b>Krav</b>	<b>SR7.2 Spridning av testpersonal</b>
<b>Beskrivning</b>	När testpersonal bestäms 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.
<b>Motivering</b>	Testresultaten får inte färgas av en testpersons erfarenhet av ett system de inte bör ha tillgänglighet till. Därför är det viktigt att vi utför test för varje gränssnitt där användaren inte har tidigare erfarenhet av andra gränssnitt än motsvarande roll skulle ha.
<b>Behov</b>	Låg
<b>Prioritet</b>	Låg
<b>Stabilitet</b>	Instabilt
<b>Spår</b>	N/A
<b>Källa</b>	Internt/Nyx
<b>Verifierbarhet</b>	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.

## 22 Dokumentationskrav

*Documentation requirements. A proposal for future system documentation that will be suitable for the job roles and skill levels identified for different end users (actors). (See SRD Section 2.4 and URD Section 2.4)*

### 22.1 Kund

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

Krav	SR 8.1 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	URD #2, UR8.1
Verifierbarhet	Kontrollera att det finns instruktioner för hur bokningen går till på bokningswebbsidan.

### 22.2 Kårspexet

För betrodda medlemmar i kårspexet

Krav	SR 8.2 Tips och tricks
Beskrivning	En sida där dolda hjälpfunktioner och knep finns dokumenterade. Sidan skall vara åtkomlig från webbgränssnittet. Varje syssla i systemet har en unik undersida i "Tips och tricks" som endast är relevant för sysslan. Exempel på knep eller dold hjälpfunktion är tangentbordskommandon.
Behov	Standard
Prioritet	Låg
Stabilitet	Stabil
Källa	
Verifierbarhet	Kontrollera att alla aktuella gränssnitt kan nå sidan. Kontrollera att alla sysslor med dolda funktioner finns dokumenterade.

<b>Krav</b>	<b>SR 8.3 Instruktioner för säljare</b>
<b>Beskrivning</b>	De sysslor som kan utföras av en säljare skall beskrivas var för sig där de utförs i gränssnittet.
<b>Behov</b>	Standard
<b>Prioritet</b>	Medel
<b>Stabilitet</b>	Stabil
<b>Källa</b>	URD #32
<b>Verifierbarhet</b>	Observera att det finns synliga instruktioner för varje syssla i säljargränssnittet.

<b>Krav</b>	<b>SR 8.4 Dokumentation för den ekonomiansvarige</b>
<b>Beskrivning</b>	Instruktioner för hur den ekonomiansvarige kan använda gränssnittets alla funktioner. Utvalda sysslor kan utformas som “gå igenom (walk thru) tutorials”. Instruktionerna skall kunna skrivas ut på behagligt pappersformat och hittas på hemsidan i relation till gränssnittet.
<b>Behov</b>	Plus
<b>Prioritet</b>	Låg
<b>Stabilitet</b>	Stabil
<b>Källa</b>	URD #33
<b>Verifierbarhet</b>	Kontrollera att dokumentationen kan nås från ekonomigränssnittet.

<b>Krav</b>	<b>SR 8.5 Dokumentation för administratören</b>
<b>Beskrivning</b>	Instruktioner för hur administratören kan använda gränssnittets alla funktioner. Utvalda sysslor kan utformas som “gå igenom (walk thru) tutorials”. 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 hemsidan i relation till gränssnittet.
<b>Behov</b>	Plus
<b>Prioritet</b>	Låg
<b>Stabilitet</b>	Stabil
<b>Källa</b>	URD #34, UR8.2
<b>Verifierbarhet</b>	Kontrollera att instruktionerna kan nås från administrationsgränssnittet.

## 22.3 Systemutvecklare

För utvecklare som skall anpassa eller berika produktens funktionalitet.

<b>Krav</b>	<b>SR 8.6 Databasens struktur</b>
<b>Beskrivning</b>	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.
<b>Behov</b>	Standard
<b>Prioritet</b>	Medel
<b>Stabilitet</b>	Stabil
<b>Källa</b>	UR9.1
<b>Verifierbarhet</b>	Kontrollera att dokumentationen ligger där den skall och att dokumentet beskriver den aktuella databasen.

<b>Krav</b>	<b>SR 8.7 Webbplatsens struktur</b>
<b>Beskrivning</b>	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.
<b>Behov</b>	Standard
<b>Prioritet</b>	Medel
<b>Stabilitet</b>	Stabil
<b>Källa</b>	
<b>Verifierbarhet</b>	Kontrollera att dokumentationen ligger där den skall och att dokumentet beskriver den aktuella webbplatsen.

## 23 Säkerhetskrav

*Security requirements. Requirements on data security, access security, security of external system and overall environment. May include e.g. firewall and cryptology techniques, password protection, data encryption, underlying OS security etc.*

<b>Krav</b>	SR9.1 Skydd av lösenord
<b>Beskrivning</b>	Lösenord ska inte sparas i klartext i databasen. De bör sparas som saltade SHA-1 lösenord.
<b>Motivering</b>	Ikke auktoriserade användare ska inte kunna få tillgång till ett användarlösenord
<b>Behov</b>	Standard
<b>Prioritet</b>	Hög
<b>Stabilitet</b>	Stabilt
<b>Källa</b>	URD6.1
<b>Verifierbarhet</b>	Verifiera att lösenorden i databasen sparas som saltade SHA-1 strängar

  

<b>Krav</b>	SR9.2 Användarroller
<b>Beskrivning</b>	Användare ska ges olika roller, som har tillgång till olika delar av systemet
<b>Motivering</b>	Alla användare ska inte ges tillgång till hela systemet, utan endast till den del av systemet som de behöver för att kunna utföra sina uppgifter
<b>Behov</b>	Standard
<b>Prioritet</b>	Hög
<b>Stabilitet</b>	Stabilt
<b>Källa</b>	URD6.1
<b>Verifierbarhet</b>	Verifiera att en de olika användarrollerna endast kommer åt de delar av systemet som de ska komma åt

  

<b>Krav</b>	SR9.3 Datasäkerhet
<b>Beskrivning</b>	Alla frågor till databasen ska använda funktioner som förhindrar SQL-injections.
<b>Motivering</b>	Det ska inte gå att oaktorisera komma åt eller ändra innehåll i databasen utanför det som sidan tillåter
<b>Behov</b>	Standard
<b>Prioritet</b>	Hög
<b>Stabilitet</b>	Stabilt
<b>Källa</b>	URD6.2
<b>Verifierbarhet</b>	Funktionaliteten finns redan i rails, men vi bör verifiera att ev. egenskrivna SQL-frågor inte är sårbara

<b>Krav</b>	SR9.4 HTTPS
<b>Beskrivning</b>	HTTPS ska användas för den inloggade delen av systemet och för ev. kortbetalning
<b>Motivering</b>	HTTPS förhindrar att lösenord till inloggningen och kortbetalningar går okrypterad över internet
<b>Behov</b>	Deluxe
<b>Prioritet</b>	Låg
<b>Stabilitet</b>	Stabilt
<b>Källa</b>	URD6.2
<b>Verifierbarhet</b>	Verifiera att de delar av sidan som använder inloggning eller kortbetalning kräver https

<b>Krav</b>	SR9.5 Förhindra XSS
<b>Beskrivning</b>	För att förhindra XSS bör POST med autentitetsverifiering användas
<b>Motivering</b>	XSS möjliggör för en oauktoriserad tredjepart att modifiera databasen på ett icke tillåtet sätt
<b>Behov</b>	Standard
<b>Prioritet</b>	Hög
<b>Stabilitet</b>	Stabilt
<b>Källa</b>	URD6.2
<b>Verifierbarhet</b>	Funktionaliteten finns i rails. Vi bör dock verifiera att vi inte lägger in några egna svagheter

## 24 Portabilitetskrav

*Portability requirements. Cross platform compatibility.*

Utöver webbrowser/apache/rails/databas/etc



## 25 Kvalitetskrav

*Quality requirements. Includes design quality, software quality, performance quality, report quality, documentation quality, usability quality. A description of plans and methods to impose quality. Standards for performance measurement and reporting.*

<b>Krav</b>	<b>SR11.1 Kontroll av dokumentation</b>
<b>Beskrivning</b>	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.
<b>Motivering</b>	Det är viktigt att den dokumentation som finns i systemet är av hög kvalitet. För att få en oberoende åsikt om den skrivna dokumenteringen behövs en översyn från någon som inte utvecklat koden.
<b>Behov</b>	Standard
<b>Prioritet</b>	Normal
<b>Stabilitet</b>	Stabilt
<b>Ursprung</b>	Internt
<b>Verifierbarhet</b>	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.

<b>Krav</b>	<b>SR11.2 Ansvarsfördelning för testkod</b>
<b>Beskrivning</b>	En policy som tydligt hanterar ansvarsfördelning av testkod för vem som har ansvar för att testkod skrivs och körs samt när denna skall vara skriven och köras.
<b>Motivering</b>	För att undvika att luckor sker i testningen och som en del av kontrollen av mjukvarukvalitén behövs en tydlig policy.
<b>Behov</b>	Standard
<b>Prioritet</b>	Normal
<b>Stabilitet</b>	Instabilt, policyn är ännu inte skriven
<b>Ursprung</b>	Internt
<b>Verifierbarhet</b>	Kontrollera att policy finns att tillgå för alla medlemmar i Nyx.

<b>Krav</b>	<b>SR11.3 Stresstest av systemet</b>
<b>Beskrivning</b>	Testning för den typ av användning systemet skall klara av. Vid felaktigheter analyseras dessa för åtgärd i ett testprotokoll.
<b>Motivering</b>	Systemet måste i så stor utsträckning som möjligt vara stabilt för den användning som är specificerad. Detta för att kunden skall få användning av det.
<b>Behov</b>	Standard
<b>Prioritet</b>	Normalt
<b>Stabilitet</b>	Instabilt, testerna ännu inte designade
<b>Ursprung</b>	Internt
<b>Verifierbarhet</b>	Testprotokoll med tillhörande resultat finns.

<b>Krav</b>	<b>SR11.4 Användartestning av gränssnitt</b>
<b>Beskrivning</b>	
<b>Motivering</b>	För att kontrollera användarvänligheten av systemet behöver systemets gränssnitt testas av personer utanför projektet som faller in under respektive användarkategori. Testarnas åsikter och kommentarer kommer sedan analyseras för att sedan förbättra systemet.
<b>Behov</b>	Standard
<b>Prioritet</b>	Normal
<b>Stabilitet</b>	Instabilt, testerna ännu inte designade.
<b>Ursprung</b>	Internt
<b>Verifierbarhet</b>	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.

<b>Krav</b>	<b>SR11.5 Prestandaövervakning</b>
<b>Beskrivning</b>	Övervakning av prestanda under användartester för att sedan undersöka avvikelser.
<b>Motivering</b>	Vi ska säkerställa att normal användning av systemet klarar de prestandakrav vi ställer.
<b>Behov</b>	
<b>Prioritet</b>	Högt
<b>Stabilitet</b>	Instabilt, tester ännu inte designade.
<b>Ursprung</b>	Internt
<b>Verifierbarhet</b>	Inga avvikande resultat noterade efter användartester i dessa protokoll.

<b>Krav</b>	<b>SR11.6 Granskning av rapporter</b>
<b>Beskrivning</b>	Alla rapportpunkter skall kontrolläsas av minst en person i projektgruppen som inte är ansvarig för punkten. Efter godkännande skrivs namn på den som granskat i kommentarer. Innan slutversion av rapport görs kontrolleras att alla punkter har granskats.
<b>Motivering</b>	För att säkerställa kvalitén på rapporter krävs granskning av en för punkten oberoende person.
<b>Behov</b>	Standard
<b>Prioritet</b>	Högt
<b>Stabilitet</b>	Stabilt
<b>Ursprung</b>	Internt
<b>Verifierbarhet</b>	Namn på granskande person finns i kommentarer till alla rapportpunkter.

<b>Krav</b>	<b>SR11.7 Policy för kontroll av kod</b>
<b>Beskrivning</b>	En policy som säger hur och av vem koden skall granskas samt hur detta skall dokumenteras.
<b>Motivering</b>	Det är viktigt att en policy finns för att säkerställa kvalitén på koden.
<b>Behov</b>	Standard
<b>Prioritet</b>	Normal
<b>Stabilitet</b>	Instabilt, policy ännu inte skriven.
<b>Ursprung</b>	Internt
<b>Verifierbarhet</b>	Den av policyn rekommenderade dokumentationen av granskningen kontrolleras.

## 26 Tillförlitlighetskrav

*Reliability requirements. Includes uptime, mean time to failure, accessibility, loading, average performance, worst-case performance, etc.*

Krav rörande administreringen av servern ligger utanför själva systemet och kommer hanteras av tredje part. De är alltså inte relevanta för detta dokument. Dessa krav inkluderar, men är inte begränsat till: upptid, tillgänglighet och svarstid.

## 27 Underhållskrav

*Maintainability requirements. Levels of code documentation, software commenting, commenting standards needed to maintain, repair and upgrade the code.*

Krav	SR13.1 Dokumentation
<b>Beskrivning</b>	Dokumentation ska finnas (och uppdateras vid ändringar) för skrivna Ruby-klasser (däribland Rails-kontroller, -modeller och -mailers) och dess metoder.
<b>Motivering</b>	Kårspexet vill själva kunna vidareutveckla bokningssystemet, vilket i så fall skulle göras av utvecklare som är obekanta med källkoden. Väldokumenterad kod underlättar i sådana fall mycket.
<b>Behov</b>	Standard
<b>Prioritet</b>	Låg
<b>Stabilitet</b>	Stabilt
<b>Källa</b>	UR8.5
<b>Verifierbarhet</b>	Granska källkoden till bokningssystemet och verifiera att ingen klass eller metod saknar dokumentation.

Krav	SR13.2 Skrivsätt
<b>Beskrivning</b>	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).
<b>Motivering</b>	Sättet som dokumentationen är skriven på bör vara konsekvent och lättöverskådligt.
<b>Behov</b>	Plus
<b>Prioritet</b>	Låg
<b>Stabilitet</b>	Stabilt
<b>Källa</b>	UR8.5
<b>Verifierbarhet</b>	Undersök dokumentationen och kontrollera att inga delar av den är formulerade på felaktigt sätt.

## 28 Trygghetskrav

*Safety requirements. Hazard situations, plans and methods to avoid system failure under hazard. Levels of safety assurance.*

Krav	SR14.1 Databasbackup
Beskrivning	Databasen ska backas upp till en annan disk i en annan lokal så att data aldrig riskerar att gå förlorad
Motivering	Mer än några timmars data ska aldrig försvinna vid en diskkrash eller brand
Behov	Standard
Prioritet	Låg
Stabilitet	Stabilt
Källa	URD6.3
Verifierbarhet	Verifiera att backup av databasen är uppsatt korrekt

## 29 Spårningsmatris mellan användarkrav och mjukvarukrav

*User Requirements vs Software Requirements Traceability matrix. Gives a table cross referencing software functional requirements to user functional requirements. (It is not necessary to trace non-functional requirements here, since it can be difficult to judge performance aspects from a logical model alone.*

Bild (matris/tabell) i

[http://www.nada.kth.se/~karlm/mvk/Project\\_Handbook.pdf](http://www.nada.kth.se/~karlm/mvk/Project_Handbook.pdf)

på sida 15.