



Bokningssystem för Kårspexet

Architectural Design Document

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

Nyx

Version 1.0 – *Final Draft*
March 1, 2011

Abstract

This document describes the architectural design of Nyx' booking system for Kårspexet. It aims to give Nyx' developers a more defined picture of what needs to be delivered during the software development phase. Within, the reader will find a brief overview of the system's architecture, external interfaces, and what methods will be used while implementing the system. Then a full, highly detailed description is given for all the system's components, from its models, and controllers, to its layouts, views, and existing libraries. The project's feasibility is also included, with a likely time plan for the system's development phase. Lastly, a matrix is given that lets the reader trace requirements back to the System Requirements Document.

Ändringslogg

Version	Ändringar
1.0	Färdig korrekturläst version. Mindre formatteringsändringar gjorda.
0.8	Korrekturläsning och rättning, närmar sig färdig version.
0.3	Formatering av tabeller, korrekturläsning och planering skapad.
0.2	Dokumentformatering, tabeller intakta.
0.1	Första sammanställd version av dokumentet.

Dokumentversioner

Dokumentet har genererats från följande deldokument.

ADD/abstract version: 6.

ADD/Ändringslogg version: 9.

Gruppmedlemmar version: 3.

ADD/Introduktion version: 2.

ADD/Introduktion/Syfte version: 5.

ADD/Introduktion/Mjukvarans omfattning version: 3.

ADD/Introduktion/Definitioner akronymer och förkortningar version: 35.

ADD/Introduktion/Källor version: 8.

ADD/Introduktion/Dokumentöversikt version: 10.

ADD/Systemöverblick version: 14.

ADD/Systemkontext version: 16.

ADD/Systemdesign version: 3.

ADD/Systemdesign/Designmetod version: 27.

ADD/Systemdesign/Uppdelningsbeskrivning version: 26.

ADD/Komponentbeskrivning version: 22.

ADD/Komponentbeskrivning/Model version: 85.

ADD/Komponentbeskrivning/Controller version: 27.

ADD/Komponentbeskrivning/View version: 63.

ADD/Komponentbeskrivning/Övriga_komponenter version: 22.

ADD/Genomförbarhet- och resursuppskattning version: 74.

ADD/Spårningsmatris mellan mjukvarukrav och strukturella krav version: 9.

ADD/appendix version: 1.

Grupppmedlemmar

Projektgruppen **Nyx** består av följande medlemmar.

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

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

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

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

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

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

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

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

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

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

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

Innehåll

1	Introduktion	6
1.1	Syfte	6
1.2	Mjukvarans omfattning	6
1.3	Definitioner akronymer och förkortningar	6
1.4	Källor	7
1.5	Dokumentöversikt	8
2	Systemöverblick	9
3	Systemkontext	10
3.1	Ruby on Rails	10
3.2	MySQL	10
3.3	Apache HTTP Server	10
4	Systemdesign	11
4.1	Designmetod	11
4.1.1	Model-View-Controller	11
4.1.2	Komponentbeskrivningar	11
4.2	Uppdelningsbeskrivning	13
4.2.1	Modeller	14
4.2.2	Vyer	14
4.2.3	Kontroller	14
4.2.4	Kopplingar mellan modeller och kontroller	15
5	Komponentbeskrivning	16
5.1	Modeller	16
5.1.1	Bokning	16
5.1.2	Föreställningar	18
5.1.3	Administrativa och övriga modeller	21
5.2	Kontroller	23
5.2.1	Kund	24
5.2.2	Inloggning	24
5.2.3	Admin	25
5.2.4	Ekonomichef	31
5.2.5	Säljare	32
5.3	Vyer	33
5.3.1	Layouter	33
5.3.2	Inloggning	34
5.3.3	Admin	35
5.3.4	Säljare	54
5.3.5	Kund	58
5.4	Övriga komponenter	62
6	Genomförbarhet- och resursuppskattning	66
6.1	Sammanfattning av slutsatser av en genomförbarhetsundersökning av arkitekturmodellen.	66
6.2	Komponentprioritering	66
6.2.1	Modeller	66

6.2.2	Kontroller	67
6.2.3	Vyer	68
6.2.4	Övriga komponenter	69
6.3	Framtida projektuppgifter med tidsuppskattning	70
6.4	Beroenden mellan uppgifter	72
6.5	Uppgiftsplanering	73
6.6	Möjliga risker	74
7	Spårningsmatris	75
7.1	Funktionella krav	75
7.2	Icke-funktionella krav	76

1 Introduktion

1.1 Syfte

Dokumentets syfte är att specificera detaljer kring produktens arkitektur. Den är speciellt skriven för att underlätta arbetet inom Nyx, men skall också kunna läsas av de som skall underhålla produkten åt kunden Kårspexet. Dokumentet redogör för vilka komponenter som finns, systemdesignen samt en uppskattning av Nyx resurser och genomförbarhet.

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

Action (Rails action) *En metod/funktion tillhörande en kontroll i MVC principen.*

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

AJAX (Asynchronous JavaScript and XML) *En teknik för att göra webbsidor mer interaktiva. AJAX tillåter webbläsaren att hämta information från webbservern utan att webbsidan behöver laddas om.*

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

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

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

Gem *Term för Ruby on Rails-bibliotek som ger utökad funktionalitet.*

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*

Kontroll (Railscontroller) *En komponent i MVC principen som tar emot händelser/inmatningar från användaren. Kontrollen kan manipulera modeller och väljer vilken vy som ska visas.*

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

Modell (Railsmodel) *En datastruktur i MVC principen som laddar, sparar och hanterar data genom att vanligtvis arbete mot en databas. Modellen kan innehålla grundläggande logik för att operera på dadan.*

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

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

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

Partial (Railspartial) *En komponent som tillåter en vy i MVC principen att dela upp en renderingsprocess i flera filer. En vy kan bestå av flera partiella vyer som var och en har en avgränsad uppgift.*

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

Router (Railsrouter) *En komponent som identifierar URLer och anropar rätt action på en kontroller. Routern kan generera URLer för länkar och knappar i vyer så hårdkodade URLer kan undvikas.*

Ruby *Ett objektorienterat programmeringsspråk.*

Rails (Ruby On Rails) *Ett abstrakt mjukvarubibliotek med öppen källkod för utveckling av web-bapplikationer.*

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

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.*

Vy (Railsview) *En komponent i MVC principen som renderar innehållet från en modell till ett interaktivt användargränssnitt. Flera vyer är ofta kopplade till samma modell fast med olika syften.*

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.*

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 [4.2.1] den första (1) referensen för rubriken “Uppdelningsbeskrivning” (2) under sektion “Systemdesign” (4).

Apache HTTP Server

[2011-02-28] <http://httpd.apache.org/>

Hänvisning till källan görs från referenserna: [1.3.1].

Ruby on Rails Guides: Getting Started with Rails

[2011-02-28] http://guides.rubyonrails.org/getting_started.html

Hänvisning till källan görs från referenserna: [4.2.1].

1.5 Dokumentöversikt

Detta dokument inleds med en kort introduktion. Efter denna följer sektion 2 som ger en överblick av hela systemet. Detta bör vara utgångspunkten för läsare som inte redan är bekanta med Nyx system.

I sektion 3 beskrivs systemets gränssnitt mot omgivningen. Sektion 4 beskriver systemets interna design, som sedan går igenom i detalj i sektion 5 där alla delkomponenter beskrivs.

I sektion 6 görs en bedömning av projektets genomförbarhet utifrån uppskattningar av behov och tillgänglighet av resurser, främst arbetstid för t ex programmering och testning. Detta inkluderar en riskbedömning.

Sektion 7 kopplar krav från SRD:n till ADD:n. För varje krav listas de komponenter vars direkta syfte är att uppfylla kravet.

2 Systemöverblick

Nyx biljettbokningssystem kommer att ersätta det biljettbokningssystemet som för närvarande används av Kårspexet. Det innebär att vårt system kommer användas på Kårspexets webbplats och då vi endast utvecklar biljettbokningssystemet kommer vårt system behöva passa in på den befintliga webbplatsen. Dock ska Nyx systemet inte interagera med det befintliga systemet, annat än med länkar och därmed behöver vi bara anpassa vårt system visuellt.

Ruby on Rails är det ramverk vi kommer använda och kan därmed anses som ett extern system. Övriga externa system som kommer användas av vårt system är MySQL, för att få tillgång till en databas, samt Apache HTTP Server, för att hantera HTTP-kommunikationen med användarens webbläsare.

Nyx valde att utveckla applikationen i dessa externa system eftersom vi hade mycket positiv erfarenhet av dessa inom gruppen. Dessutom så arbetar Ruby on Rails väldigt väl mot MySQL så vi får därmed mycket funktionalitet på köpet när vi använder dem tillsammans. Anledningen till att Apache valdes var för att det är välanvänt, väldokumenterat samt att det är öppen källkod vilket underlättar för oss.

3 Systemkontext



Systemet är designat för att köras på en webbserver där alla systemets komponenter befinner sig internt på servern, som används via ett webbläsarfönster. Apache anropas som i sin tur kör igång Ruby som i förväg har laddat in Rails biblioteket, applikationen och övriga komponenter. Där behandlas anropet och Rails kommer att kommunicera med MySQL och hämtar relaterad data. Rails förbereder sedan resultatet och skickar tillbaka detta till Apache som vidarebefordrar det till klienten som i sin tur presenterar innehållet i webbläsaren.

3.1 Ruby on Rails

Vi kommer använda MVC ramverket Ruby on Rails som ger oss verktyg för webbutveckling i ruby. Det kommer att utgöra grunden i vårt system.

3.2 MySQL

Applikationen kommer använda sig av en MySQL databas för att hantera data för de olika användargrupperna. Databasen anropas och relevant data läses in och modifieras från Ruby applikationens olika vyer.

3.3 Apache HTTP Server

Servern kommer att köra Apache HTTP Server för att sköta kommunikationen mellan servern och applikationen. När en användare ansluter tar Apache emot anropet och skickar det vidare till applikationen. Apache kommer även ta emot resultat från applikationen och vidarebefordrar det till användaren.

4 Systemdesign

4.1 Designmetod

4.1.1 Model-View-Controller

Nyx har valt att följa designprincipen Model-View-Controller för utveckling av systemet. MVC metoden är en metod för att separera modeller, logik och det visuella interfacet i olika komponenter. Modellerna (*Models*) hanterar och lagrar data som är relevant för systemet. De ser även till att ingen ogiltig data sparas i databasen. Kontroller (*Controllers*) sköter all logik och är ett mellanlager mellan modellerna och interfacet användaren ser. Vyerna (*Views*) i sin tur renderar det interface som användaren ser och sköter postning tillbaka till kontrollerna.

4.1.2 Komponentbeskrivningar

Komponentbeskrivningarna under punkt fem (5) är upplagda enligt följande format.

Alla komponenter har en identifierare på formen 5.Tn, där T är någon av bokstäverna MVCLPX, som står för vilken typ komponenten är, och n är ett tal från ett (1) och uppåt.

I vissa fall används noteringen *@namn* för variabelnamn. Detta indikerar att variabeln är en instansvariabel.

Komponentrubriker

Alla komponenter har en tabell med följande innehåll.

Typ	Kontroll/Modell/Layout/Vy/Partiell vy/Övrig komponent
Syfte	Specificerar vilket syfte komponenten fyller. Referenser till SR-krav.
Funktion	Anger vilka olika funktioner komponenten bidrar med.
Delkomponenter	Eventuellt andra komponenter som är en del av/ingår i denna.
Beroenden	Krav för användandet av denna komponent.
Gränssnitt	Publika metoder som går att anropa. Inkluderar metodnamn och kortare beskrivning av vad metoden gör. Om delar av gränssnittet kräver att användaren är inloggad/admin/accountant/sales ska detta specificeras här.
Resurser	Vilka resurser (modeller/klasser/gems) som komponenten utnyttjar direkt (ej indirekt).
Källor	Referenser till information om eventuellt använda gem och andra externa komponenter.
Process	Lista möjliga arbetsflöden (övergångar).
Data	Tillgängliga instansvariabler för/som används av komponenten (de som är relevanta från ett externt perspektiv).

Komponenttyper

M - Model (Modell)

En modell representerar en viss typ av data och sköter kopplingen till databasen.

Specifiseringar och tillägg till komponentrubrikerna från 4.1.2

Syfte	Beskriver hur ett objekt av typen fungerar samt vilka begränsningar som ställs.
Funktion	En modells funktion är i de flesta fall att representera, skapa och uppdatera en typ av objekt i databasen, samt att validera datan innan insättning och uppdatering. Om ytterligare funktionalitet finns eller om något saknas specificeras det för respektive modell.
Delkomponenter	Vilka relationer till andra modeller som finns (motsvarande Rails has_many- och belongs_to-nyckelord).
Beroenden	Krav för att objekt av denna modell ska vara giltig (valideringskrav), inklusive relationer.
Gränssnitt	I de flesta fall utgörs gränssnittet av de metoder som ärvs av ActiveRecord::Base.
Resurser	ActiveRecord::Base samt de eventuella resurser (modeller/gem) som modellen utnyttjar direkt (ej indirekt).
Källor	Referenser till information om eventuellt använda gem, externa bibliotek, etc.
Process	Eventuella valideringssteg eller andra processer för manipulation av objektet (inkluderar ej skapande/sparande/borttagning av objekt).
Data	Lista med namn, typ och eventuellt beskrivning av modellens attribut. Attributnamn följer lowercase_with_underscore-namngivning. Referenser till objekt i andra modeller tas enbart upp i delkomponenter.

C - Controller (Kontroll)

Specifiseringar och tillägg till komponentrubrikerna från 4.1.2

Delkomponenter	I de flesta fall ej applicerbart.
Gränssnitt	Listar och beskriver alla actions. Inkluderar vilka anropsparametrar (params[]) som utnyttjas. Actions som kräver att användaren är inloggad eller innehar specifika roller (admin/accountant/sales) nämner detta här.
Resurser	ApplicationController samt de eventuella resurser (modeller/gem) som kontrollern utnyttjar direkt (ej indirekt).
Källor	Referenser till information om eventuellt använda gem, externa bibliotek, etc.
Process	Listar möjliga arbetsflöden (övergångar) mellan actions (new > create, edit > update, index > show/destroy, etc.).
Data	Instansvariabler som görs tillgängliga för vyerna och som är relevanta från ett externt perspektiv.

V - View (Vy)

Vyerna utgör användargränssnittet. I vårt fall producerar vyerna HTML-kod.

L - Layout

En layout i Rails är en mall för hur en samling av sidor ska se ut. Den innehåller kod som annars skulle behöva finnas i vyerna.

P - Partial (Partiell vy)

En partiell vy innehåller kod som är gemensam för flera vyer. Vyer kan inkludera partiella vyer och därmed minska kodupprepning.

X - Övriga/externa komponenter

Komponenter som beskrivs här kan vara komponenter som inte passar in i MVC-modellen eller existerande komponenter.

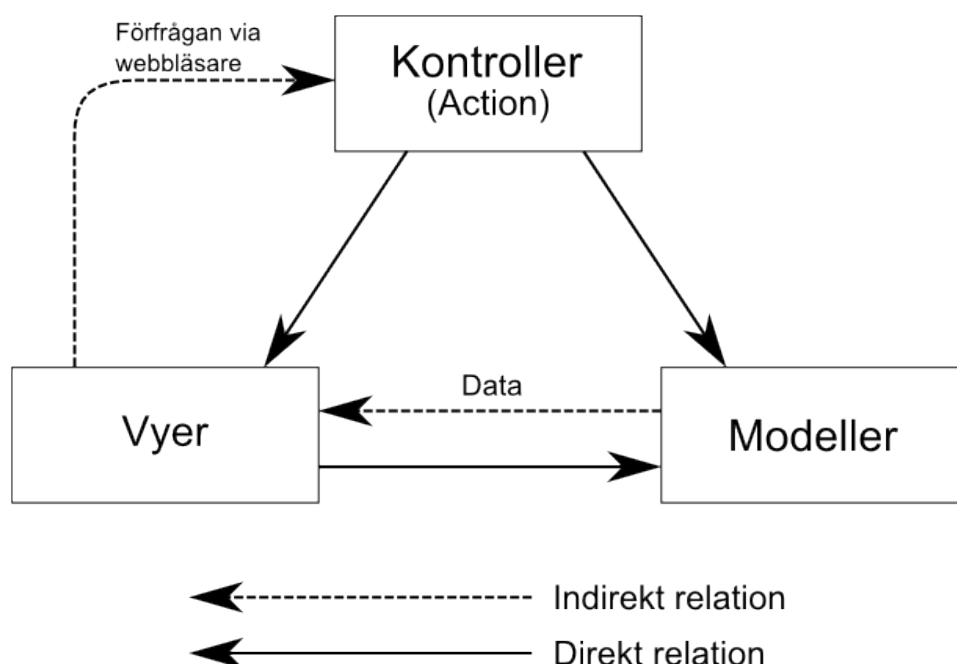
4.2 Uppdelningsbeskrivning

Systemet består av tre komponenttyper enligt MVC-modellen. [4.2.1]

» Modeller

» Vyer

» Kontroller



4.2.1 Modeller

Modeller representerar data som applikationen använder sig av. Den innehåller och består av regler för hur data kan manipuleras. Modellerna används i Nyx fall för att bestämma hur integrationen med tabeller i databasen skall gå till. Den större delen av logiken för applikationen finns i Modellen.

Bokningssystemets modeller kan delas in i följande grupper:

- » Föreställningar - Inkluderar föreställningar, omgångar och teatrar. Sköts av administratören.
- » Bokningar - Representeras av flera modeller, hanteras av både kunder, säljare, administratör.
- » Administrativa modeller - Modeller för användarkonton, rättigheter och mailmallar.

4.2.2 Vyer

Vyer representerar användargränssnittet till applikationen. Detta är olika webbsidor som visar representationer av data som finns i applikationen.

Bokningssystemet har följande grupper av vyer, vilka motsvarar olika användare samt vyer för inloggning:

- » Administration
- » Ekonomi
- » Kund
- » Login
- » Säljare

4.2.3 Kontroller

Kontroller kopplar samman modeller och vyer. I vårt fall är kontrollerna ansvariga för att hantera de inkommande förfrågningarna som kommer från användarens webbläsare. De hämtar och modifierar data från modellerna och gör den tillgänglig för vyerna.

Bokningssystemet har följande typer av kontroller:

- » Administration
- » Ekonomiansvarig
- » Försäljning
- » Kund

4.2.4 Kopplingar mellan modeller och kontroller

Modell	Modifieras av
Theater	TheatersController
Section	SectionsController
Seat	SeatsController
Batch	BatchesController
Show	ShowsController
Pricing	PricingsController
Reservation	BookingController, SalesController, ReservationsController
ReservationCounter	BookingController, SalesController, ReservationsController
Placement	PlacementsController, SalesController
MailTemplate	MailingController
User	UsersController
Ability	
UserSession	SessionController

5 Komponentbeskrivning

En beskrivning av tabellformat och notationer nedan återfinns under punkt 4.1.

5.1 Modeller

5.1.1 Bokning

5.M1 Reservation

Typ	Modell
Syfte	Håller information om en bokning (SR1.2, SR1.3).
Funktion	
Delkomponenter	Tillhör en Show. Har flera Placements. Har flera ReservationCounters.
Gränssnitt	Ärver metoder från ActiveRecord::Base, se ActiveRecord.
Beroenden	Måste referera till existerande Show. Måste ha minst en ReservationCounter. Om inte <code>creator</code> existerar måste <code>customer_reservation</code> vara giltig.
Resurser	grouped_validations för att hantera valideringar
Källor	grouped_validations: http://rubydoc.info/gems/grouped_validations/0.2.2/file/README.rdoc
Process	Valideringsgrupp customer_reservation email måste vara giltig. adress måste existera. post_code måste vara giltig. post_town måste vara giltig. paymentoption måste finnas. deliverymethod måste finnas.
Data	name : string phone : string email : string adress : string post_code : decimal post_town : string comment : text paymentoption : plusgiro, kontant, kort (integer) deliverymethod : brev, uthämtning (integer) paid : boolean paid_sum : decimal placed : boolean retrieved : boolean cost : decimal hash_key : string - identifierar bokningen vid avbokning. creator : integer

5.M2 ReservationCounter

Typ	Modell
Syfte	Håller information om antal platser i en viss sektion, med ett visst pris för en bokning (SR1.2).
Funktion	
Delkomponenter	Tillhör en Reservation. Tillhör en Pricing. Tillhör en Section.
Gränssnitt	Ärver metoder från ActiveRecord::Base, se ActiveRecord.
Beroenden	Måste ha existerande Reservation, Pricing och Section, samt antal > 0.
Resurser	Inga
Källor	Inga
Process	Ej applicerbart
Data	antal : decimal

5.M3 Placement

Typ	Modell
Syfte	En boknings placeringar (sittplatser) (SR1.5, SR1.11).
Funktion	
Delkomponenter	Tillhör en Reservation. Tillhör en Seat. Tillhör en Show. Tillhör en User.
Gränssnitt	Ärver metoder från ActiveRecord::Base, se ActiveRecord.
Beroenden	Måste ha existerande Reservation, Show och Seat
Resurser	Inga
Källor	Inga
Process	Ej applicerbart
Data	Inga.

5.1.2 Föreställningar

5.M4 Theater

Typ	Modell
Syfte	Modell för teater (SR1.9, SR1.22).
Funktion	
Delkomponenter	Har flera Sections. Har flera Seats, genom Sections. Har flera Batches. Har flera Shows, genom Batches.
Gränssnitt	Ärver metoder från ActiveRecord::Base, se ActiveRecord.
Beroenden	Måste ha internal_name, public_name och picture.
Resurser	Inga
Källor	Inga
Process	Ej applicerbart
Data	internal_name : string public_name : string description : string comment : string picture : string - referens till bild för att visa sektioner och sittplatser.

5.M5 Section

Typ	Modell
Syfte	Modell för sektion (SR1.9, SR1.22, SR1.23, SR1.24).
Funktion	
Delkomponenter	Tillhör en Theater. Har flera Seats.
Gränssnitt	Ärver metoder från ActiveRecord::Base, se ActiveRecord.
Beroenden	Måste tillhöra en existerande Theater
Resurser	Inga
Källor	Inga
Process	Ej applicerbart
Data	Inga.

5.M6 Seat

Typ	Modell
Syfte	Modell för sittplats i en teaters sektion (SR1.9, SR1.25).
Funktion	
Delkomponenter	Tillhör en Section. Har flera Placements. Har flera Placement_locks.
Gränssnitt	Ärver metoder från ActiveRecord::Base, se ActiveRecord.
Beroenden	Måste referera till en existerande Section
Resurser	Inga
Källor	Inga
Process	Ej applicerbart
Data	<code>number</code> : <code>integer</code> - stolsnummer <code>row</code> : <code>integer</code> (x,y)-koordinat i teaterns bild eller dylikt.

5.M7 Batch

Typ	Modell
Syfte	Håller information om en omgång (SR1.10).
Funktion	
Delkomponenter	Tillhör en Theater. Har flera Shows.
Gränssnitt	Ärver metoder från ActiveRecord::Base, se ActiveRecord.
Beroenden	Måste referera existerande Theater description får inte vara tom. Synlighet och startdatum måste finnas.
Resurser	Inga
Källor	Inga
Process	Ej applicerbart
Data	<code>description</code> : <code>text</code> <code>visible</code> : <code>boolean</code> - Synlighet, huruvida kunden kan se och boka biljetter till omgången. <code>start_date</code> : <code>datetime</code>

5.M8 Show

Typ	Modell
Syfte	Data för en enskild föreställning (SR1.10).
Funktion	
Delkomponenter	Tillhör en Batch. Tillhör en Theater, genom Batch.
Gränssnitt	Ärver metoder från ActiveRecord::Base, se ActiveRecord.
Beroenden	Måste referera till en existerande Batch.
Resurser	Inga
Källor	Inga
Process	Ej applicerbart
Data	<code>date : datetime</code>

5.M9 Pricing

Typ	Modell
Syfte	I varje Batch har varje Section i teatern normalpris och studentpris. Detta sköts av Pricing-modellen (SR1.30).
Funktion	
Delkomponenter	Tillhör en Batch. Tillhör en Section.
Gränssnitt	Ärver metoder från ActiveRecord::Base, se ActiveRecord.
Beroenden	Referenser till Batch och Section måste vara giltiga. <code>normal_price</code> och <code>student_price</code> måste vara icke-negativa heltal.
Resurser	Inga
Källor	Inga
Process	Ej applicerbart
Data	<code>normal_price : integer</code> <code>student_price : integer</code>

5.1.3 Administrativa och övriga modeller

5.M10 User

Typ	Modell
Syfte	Hålla koll på de olika roller användare kan logga in som (SR1.7, SR9.1, SR9.2).
Funktion	
Delkomponenter	Tillhör en Ability.
Gränssnitt	Ärver metoder från ActiveRecord::Base, se ActiveRecord.
Beroenden	username måste vara unikt. password måste finnas.
Resurser	Inga
Källor	Inga
Process	Ej applicerbart
Data	username : string password : string valid_to : datetime - kan vara nil, vilket betyder att kontot gäller tills vidare.

5.M11 MailTemplate

Typ	Modell
Syfte	Hantera och mailmallar för färdigformattrade mail (SR1.27).
Funktion	
Delkomponenter	Inga
Gränssnitt	Ärver metoder från ActiveRecord::Base, se ActiveRecord.
Beroenden	content måste finnas.
Resurser	ActiveRecord::Base
Källor	Inga
Process	Ej applicerbart
Data	description : string title : string content : string

5.M12 UserSession

Typ	Modell
Syfte	(Se 5.X2 AuthLogic, samt dess dokumentation) (SR1.1)
Funktion	Håller koll på de sessioner som finns inloggade
Delkomponenter	Har en User
Gränssnitt	<code>find</code> Letar upp efterfrågad session om den finns. <code>user</code>
Beroenden	Inga
Resurser	AuthLogic::Base::Session
Källor	AuthLogic: https://github.com/binarylogic/authlogic
Process	Ej applicerbart
Data	Inga

5.M13 Ability

Typ	Modell
Syfte	(Se 5.X3 CanCan, samt dokumentationen) (SR1.1, SR9.2)
Funktion	Håller koll på vilka rättigheter varje användare har.
Delkomponenter	Inga
Gränssnitt	<code>can</code> Returnerar om användaren har rättighet att göra den efterfrågade handlingen.
Beroenden	Inga
Resurser	ActiveRecord::Base, CanCan, User
Källor	CanCan: https://github.com/ryanb/cancan
Process	Ej applicerbart
Data	Inga

5.2 Kontroller

5.C1 ApplicationController

Typ	Kontroller
Syfte	Rails-applikationens baskontroller som bland annat ska hantera autentisering (SR1.1).
Funktion	Alla kontroller ärver funktionerna som denna klass definierar, vilket inkluderar autentisering.
Delkomponenter	Inga
Beroenden	Inga
Gränssnitt	ApplicationController tillhandahåller inga actions utan enbart filter och hjälpmetoder åt övriga kontroller. De hjälpmetoder som definieras är: <code>current_user</code> - Returnerar en User-instans för den aktiva användaren (försöker logga in via Authlogic om nödvändigt). <code>current_user_session</code> - Returnerar en UserSession-instans för den aktiva inloggningen (försöker logga in via Authlogic om nödvändigt). <code>require_user</code> - Hjälpmetod som kastar en exception om den anropas och användaren inte är autentiserad.
Resurser	AuthLogic och CanCan.
Källor	Inga
Process	Filter definierade med <code>before_filter</code> resp. <code>after_filter</code> körs före resp. efter den anropade kontrollerns action.
Data	Inga

5.2.1 Kund

5.C2 BookingController

Typ	Kontroller
Syfte	Hanterar bokningsproceduren för kunden (SR1.2, SR1.3).
Funktion	Skapar en ny bokning och sparar den eller avbokar en existerande bokning. Bokningsprocessen delas upp i flera privata metoder som anropas beroende på vilket steg i bokningen man befinner sig i.
Delkomponenter	Inga
Beroenden	Inga
Gränssnitt	new - påbörjar en bokning create - sparar data i användarsessionen allt eftersom och vid sista steget lagrar en bokning i databasen cancel - presenterar en bokning från en hash_key med möjlighet till avbokning via ett formulär destroy - tar bort en bokning.
Resurser	ApplicationController, Reservation, ReservationCounter, Batch och Show.
Källor	Inga
Process	new > (create)+ > create cancel > destroy
Data	@step - steg i bokning @steps - array med vilka steg som finns @reservation - bokningsdata

5.2.2 Inloggning

5.C3 SessionController

Typ	Kontroller
Syfte	Hantera inloggning och sessioner (SR1.1).
Funktion	Skapar ny session vid inloggning, tar bort session vid utloggning.
Delkomponenter	Inga
Beroenden	Inga
Gränssnitt	new - presenterar inloggningsformuläret create - tar emot data från formuläret och skapar en ny UserSession destroy - loggar ut
Resurser	ApplicationController och UserSession.
Källor	Inga
Process	new > create destroy > new
Data	Inga

5.2.3 Admin

5.C4 AdminIndexController

Typ	Kontroller
Syfte	Generera data för översikt och statistik (SR1.16, SR1.17, SR1.18).
Funktion	Presenterar en översikt samt statistiksidorna för administratören och ekonomichefen.
Delkomponenter	Inga
Beroenden	Autentiserad som administratör eller ekonomichef.
Gränssnitt	<code>index</code> - översikt <code>statistics</code> - statistik
Resurser	ApplicationController, Reservation, Batch och Show.
Källor	Inga
Process	Ej applicerbart
Data	<code>@unplaced</code> - antal betalade bokningar att placera <code>@shows</code> - tabell med information om aktuella föreställningar (antal bokade platser mm) <code>@stats</code> - tabell över föreställningar, sektionsgrupper(prisklass), med antal ordinarie, studenter, gratis, platser kvar

5.C5 ReservationsController

Typ	Kontroller
Syfte	Administrering av bokningar och registrering av betalningar (SR1.12, SR1.13).
Funktion	Redigering av existerande bokningar samt skapande och borttagning. Ekonomichef kan bara läsa information och registrera betalningar.
Delkomponenter	Inga
Beroenden	Autentiserad som administratör eller ekonomichef.
Gränssnitt	Ärver resursactions från InheritedResources med undantag för <code>show</code> . <code>index</code> - lista bokningar med ev. filter <code>new</code> , <code>edit</code> - formulär för att redigera alla attributer hos en bokning bortsett från stolsplacering
Resurser	ApplicationController, InheritedResources och Reservation.
Källor	Inga
Process	Standardprocesser för resurser.
Data	Se InheritedResources. <code>@filters</code> - array med möjliga filter samt indikationer för vilka som är aktiva (<code>index</code>)

5.C6 PlacementsController

Typ	Kontroller
Syfte	Tilldelning av stolar till webbokningar (SR1.11).
Funktion	Hanterar stolsplaceringar för en given bokning med stöd för AJAX-anrop.
Delkomponenter	Inga
Beroenden	Autentiserad som administratör.
Gränssnitt	<p><code>new</code>, <code>edit</code> - presenterar aktuella placeringar och status för alla stolar med möjlighet att placera</p> <p><code>test_multiple</code> [AJAX] - validerar placeringar, skapar temporära läsningar och returnerar dess IDn</p> <p><code>update_multiple</code> - skapar och uppdaterar nya/existerande placeringar</p> <p><code>destroy</code> - tar bort existerande placeringar för given bokning</p>
Resurser	ApplicationController och Placement.
Källor	Inga
Process	<p><code>new</code> > [<code>test_multiple</code>]+ > <code>update_multiple</code></p> <p><code>edit</code> > [<code>test_multiple</code>]+ > <code>update_multiple</code></p> <p><code>edit</code> > <code>destroy</code></p>
Data	<p><code>@reservation</code> - den aktuella bokningsinstansen</p> <p><code>@placements</code> - placeringar</p> <p><code>@theater</code> - aktuell teater (bild med mera)</p> <p><code>@show</code> - information om föreställningen (datum med mera)</p> <p><code>@seats</code> - stolar med positioner och placeringsstatus (upptagen, tillfälligt låst, ledig, vald för denna bokning)</p>

5.C7 TheatersController

Typ	Kontroller
Syfte	Hantera teatrar (SR1.9).
Funktion	Hantera teatrar, inklusive möjligheten att kopiera existerande. Plus: administrera sektioner och platser.
Delkomponenter	SectionsController och SeatsController.
Beroenden	Autentiserad som administratör.
Gränssnitt	Ärver resursactions från InheritedResources. new - formulär för att skapa ny teater, inkluderar val att kopiera en existerande teater.
Resurser	ApplicationController, InheritedResources och Theater.
Källor	Inga
Process	Standardprocesser för resurser.
Data	Se InheritedResources. @theaters - existerande teatrars attributer (new)

5.C8 SectionsController

Typ	Kontroller
Syfte	Hantera sektioner (SR1.9).
Funktion	Visa och redigera sektionsindelningar av en existerande teater.
Delkomponenter	Inga
Beroenden	Autentiserad som administratör, nästlad via TheaterController.
Gränssnitt	Ärver resursactions från InheritedResources med undantag för show.index - visar sektioner för aktuell teater.
Resurser	ApplicationController, InheritedResources, Theater och Section.
Källor	Inga
Process	Standardprocesser för resurser.
Data	Se InheritedResources. @theater - aktuell teater

5.C9 SeatsController

Typ	Kontroller
Syfte	Hantera stolarnas positioner och sektionstillhörighet (SR1.9).
Funktion	Visa och redigera en given teaters stolar och deras positioner på salongsskissen. AJAX används för att hantera stolar utan att ladda om sidan.
Delkomponenter	Inga
Beroenden	Autentiserad som administratör, nästlad via TheaterController.
Gränssnitt	<code>index</code> - presenterar gränssnittet för att lägga till och redigera teaterns stolar <code>create</code> [AJAX] - sparar en ny stol <code>update</code> [AJAX] - uppdaterar en existerande stol <code>destroy</code> [AJAX] - tar bort en existerande stol
Resurser	ApplicationController, Theater och Seat.
Källor	Inga
Process	<code>index</code> > [<code>create</code> / <code>update</code>]+
Data	<code>@seats</code> - existerande stolar <code>@theater</code> - aktuell teater

5.C10 BatchesController

Typ	Kontroller
Syfte	Hantera omgångar (SR1.10).
Funktion	Visa och redigera omgångar.
Delkomponenter	ShowsController och PricingsController.
Beroenden	Autentiserad som administratör.
Gränssnitt	Ärver resursactions från InheritedResources med undantag för <code>show</code> .
Resurser	ApplicationController, InheritedResources och Batch.
Källor	Inga
Process	Standardprocesser för resurser.
Data	Se InheritedResources.

5.C11 ShowsController

Typ	Kontroller
Syfte	Hantera föreställningar (SR1.10).
Funktion	Hanterar föreställningar som hör till en given omgång.
Delkomponenter	Inga
Beroenden	Autentiserad som administratör, nästlad via BatchesController.
Gränssnitt	Ärver resursactions från InheritedResources med undantag för <code>show</code> . <code>index</code> - listar alla föreställningar för given omgång
Resurser	ApplicationController, InheritedResources och Show.
Källor	Inga
Process	Standardprocesser för resurser.
Data	Se InheritedResources. <code>@batch</code> - aktuell omgång <code>@batches</code> - existerande omgångar

5.C12 PricingsController

Typ	Kontroller
Syfte	Hantering av prisklasser (SR1.30).
Funktion	Administrering av priser på omgångsnivå. Varje sektion har ett pris för varje rabattklass. Alla pris-/sektions-kombinationer för given omgång administreras på en gång, oavsett om de redan finns eller inte.
Delkomponenter	Inga
Beroenden	Autentiserad som administratör, nästlad via BatchesController.
Gränssnitt	<code>index</code> - visar formulär för att välja priser för alla sektioner för den aktuella omgången <code>update_multiple</code> - lagrar informationen med datan från <code>index</code>
Resurser	ApplicationController, Batch, Section och Pricing.
Källor	Inga
Process	<code>index > update_multiple > index</code>
Data	<code>@batch</code> - aktuell omgång <code>@sections</code> - sektioner i omgångens teater <code>@pricings</code> - priser för varje sektion- och rabattklasskombination

5.C13 UsersController

Typ	Kontroller
Syfte	Hantera användare och deras lösenord (SR1.7, SR1.8).
Funktion	Administrering av bokningssystemets användare, inklusive lösenord-sändringar och giltighetstid för säljare.
Delkomponenter	Inga
Beroenden	Autentiserad som administratör.
Gränssnitt	Ärver resursactions från InheritedResources.
Resurser	ApplicationController, InheritedResources och User.
Källor	Inga
Process	Standardprocesser för resurser.
Data	Se InheritedResources.

5.C14 MailingController

Typ	Kontroller
Syfte	Hantera manuella och automatiska utskick av mail (SR1.14, SR1.15, SR1.27).
Funktion	Hantera mallar för mail och utskick av mail till förvalda bokningars kontaktpersoner. Nyckelord i mallarna byts ut mot motsvarande bokningsdata vid utskick. Plus: val av mall och editering vid utskick i samma steg med hjälp av AJAX (index).
Delkomponenter	Inga
Beroenden	Autentiserad som administratör.
Gränssnitt	Ärver resursactions från InheritedResources prepare - formulär för att skriva och skicka mail utan mall preview - förhandsgranska mail innan utskick send - skickar mail
Resurser	ApplicationController, InheritedResources, Reservation och Mail.
Källor	Inga
Process	Standardprocesser för resurser. ReservationsController#index > index > preview > send - Utskick av existerande mall till valda mottagare. ReservationsController#index > index > new > create > index > preview > send - Utskick av ny mall till valda mottagare. ReservationsController#index > index > prepare > send - Utskick av mail utan mall till valda mottagare.
Data	Se InheritedResources. @reciepts = session[:reciepts] - mailmottagare (semipermanent) @mail = session[:mail] - den aktuella mailmallen, kan vara temporär för att möjliggöra utskick utan mall

5.2.4 Ekonomichef

Ekonomichefen har åtkomst till AdminIndexController (för översikt och statistik) respektive ReservationsController (för att redigera existerande bokningars betalningsstatus) per SR1.19.

5.2.5 Säljare

5.C15 SalesController

Typ	Kontroller
Syfte	Hantering av säljarens gränssnitt och funktioner (SR1.5, SR1.6, SR1.25).
Funktion	Möjlighet att genomföra kontantköp (med nya bokningar) samt ta betalt för, och lämna ut existerande bokningars biljetter. new delas upp i flera privata metoder som anropas beroende på vilket steg i bokningen man befinner sig i.
Delkomponenter	Inga
Beroenden	Autentiserad som säljare.
Gränssnitt	index - startsida med länkar till ny respektive sök bokning new - påbörjar ett nytt kontantköp create - sparar data i användarsessionen allt eftersom och vid sista steget lagrar en bokning i databasen find - sök efter existerande bokning place - stolsplacering av existerande bokning, görs via Placements-Controller retrieve - presentera vilka biljetter som ska hämtas payment - presenterar hur mycket som ska tas betalt finalize - uppdaterar bokningens status till betald och utlämnad samt instruerar säljaren om att lämna ut biljetten, går ej att ångra
Resurser	ApplicationController, Reservation, ReservationCounter och Placement.
Källor	Inga
Process	index > new > [place]+ > create > retrieve > payment > finalize > index - nytt kontantköp index > find > retrieve > finalize > index - lämna ut betald biljett index > find > retrieve > payment > finalize > index - lämna ut obetald biljett index > find > place > retrieve > payment > finalize > index - lämna ut placerad biljett
Data	@step - steg i bokning @steps - array med vilka steg som finns @reservation = session[:reservation] - bokningsdata

5.3 Vyer

5.3.1 Layouter

5.L1 admin_layout.html

Typ	Layout
Syfte	Alla administratörs- och ekonomichefssidor skall ha en gemensam layout (SR1.19, SR1.28)
Funktion	Renderar en layout för administratörens och ekonomichefens sidor, inklusive meny. De länkar i menyn som visas är beroende på vem som är inloggad, ty ekonomichefen ska inte ha tillgång till alla verktyg som administratören har.
Delkomponenter	Inga
Beroenden	Inga
Gränssnitt	Ej applicerbart
Resurser	Inga
Källor	Inga
Process	Ej applicerbart
Data	menu – Array med alla menylänkar

5.L2 sales_layout.html

Typ	Layout
Syfte	Säljaren ska ha ett enkelt gränssnitt (SR1.5, SR1.5, SR1.25, SR1.28)
Funktion	Alla sidor som säljaren använder skall ha samma layout.
Delkomponenter	Inga
Beroenden	Inga
Gränssnitt	Ej applicerbart
Resurser	Inga
Källor	Inga
Process	Ej applicerbart
Data	Inga

5.L3 simple_layout.html

Typ	Layout
Syfte	Kårspexets personal behöver en layout till sin inloggning (SR1.1, SR9.2, SR1.28)
Funktion	Inloggninssidan ska ha en simpel layout med en ruta där användaren kan logga in.
Delkomponenter	Inga
Beroenden	Inga
Gränssnitt	Ej applicerbart
Resurser	Inga
Källor	Inga
Process	Ej applicerbart
Data	Inga

5.L4 customer_layout.html

Typ	Layout
Syfte	Alla bokningsprocessens steg ska ha samma layout (SR1.2)
Funktion	Layout ska matcha Kårspexets nuvarande webbsida och vara användarvänlig för kunden.
Delkomponenter	Inga
Beroenden	Inga
Gränssnitt	Ej applicerbart
Resurser	Inga
Källor	Inga
Process	Ej applicerbart
Data	menu – Array med alla menylänkar

5.3.2 Inloggning**5.V1 SessionController#new.html**

Typ	Vy
Syfte	Användaren ska kunna logga in på systemet (SR1.1, SR9.2)
Funktion	Renderar inloggningsformuläret.
Delkomponenter	Inga
Beroenden	SessionController
Gränssnitt	Ej applicerbart
Resurser	5.L3 simple_layout.html
Källor	Inga
Process	Ej applicerbart
Data	Inga

5.3.3 Admin

5.V2 AdminIndexController#index.html

Typ	Vy
Syfte	Administratören behöver en snabb, första översiktssida (SR1.19)
Funktion	Ger en översikt direkt efter inloggning.
Delkomponenter	Inga
Beroenden	AdminIndexController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@unplaced - antal betalade bokningar att placera @shows - tabell med information om aktuella föreställningar (antal bokade platser mm)

5.V3 AdminIndexController#stat_standard.html

Typ	Vy
Syfte	Administratören och ekonomichefen ska ha möjlighet att se statistik (SR1.16)
Funktion	Visar statistik i form av rådata i en tabell.
Delkomponenter	Inga
Beroenden	AdminIndexController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@stats - tabell över föreställningar, sektioner(grupper(prisklass)), med antal ordinarie, studenter, gratis, platser kvar

5.V4 AdminIndexController#stat_plus.html

Typ	Vy
Syfte	Administratören och ekonomichefen ska ha möjlighet att se organiserad statistik (SR1.17)
Funktion	Visar statistik i form av ett flertal tabeller under olika rubriker.
Delkomponenter	Inga
Beroenden	AdminIndexController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@stats - tabell över föreställningar, sektionsgrupper(prisklass), med antal ordinarie, studenter, gratis, platser kvar

5.V5 AdminIndexController#stat_deluxe.html

Typ	Vy
Syfte	Administratören och ekonomichefen ska ha möjlighet att se organiserad, grafisk statistik (SR1.18)
Funktion	Visar statistik i form av en interaktiv graf.
Delkomponenter	Inga
Beroenden	AdminIndexController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@stats - tabell över föreställningar, sektionsgrupper(prisklass), med antal ordinarie, studenter, gratis och platser kvar

5.V6 ReservationsController#index.html

Typ	Vy
Syfte	Administratören ska kunna hantera bokningar och ekonomichefen ska kunna registrera betalningar (SR1.12, SR1.13)
Funktion	Administratören (ekonomichefen) får en lista med alla bokningar. Bokningarna ska kunna sorteras och filtreras. En mängd bokningar ska kunna väljas ut och registreras som betalda.
Delkomponenter	Inga
Beroenden	ReservationsController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@reservations @filters - array med möjliga filter samt indikationer för vilka som är aktiva (index)

5.V7 ReservationsController#new.html

Typ	Vy
Syfte	Administratören ska kunna hantera bokningar (SR1.13)
Funktion	Administratören ska kunna skapa en ny bokning från sitt gränssnitt.
Delkomponenter	AR5.P1
Beroenden	ReservationsController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@reservation - den aktuella bokningsinstansen

5.V8 ReservationsController#show.html

Typ	Vy
Syfte	Administratören ska kunna hantera bokningar (SR1.13)
Funktion	Visar en bokning. (vilken data?).
Delkomponenter	Inga
Beroenden	ReservationsController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@reservation - den aktuella bokningsinstansen

5.V9 ReservationsController#edit.html

Typ	Vy
Syfte	Administratören ska kunna hantera bokningar, inklusive registrera betalningar (SR1.12, SR1.13)
Funktion	Administratören ska kunna göra ändringar i en bokning, till exempel ändra antalet biljetter, ändra betalstatus, placeringsstatus och så vidare.
Delkomponenter	AR5.P1
Beroenden	ReservationsController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@reservation - den aktuella bokningsinstansen

5.P1 ReservationsController#_form.html

Typ	Partiell vy
Syfte	Se AR.V9
Funktion	Renderar ett formulär för att redigera Reservations-objekt.
Delkomponenter	Inga
Beroenden	ReservationsController
Gränssnitt	Ej applicerbart
Resurser	Reservation
Källor	Inga
Process	Ej applicerbart
Data	@reservation - den aktuella bokningsinstansen

5.V10 PlacementsController#new.html

Typ	Vy
Syfte	Administratören ska kunna placera ut bokningar (SR1.11, SR1.25)
Funktion	Administratören ska kunna placera ut platser i den korrekta sektionen som kunden har bokat.
Delkomponenter	Inga
Beroenden	PlacementsController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@reservation - den aktuella bokningsinstansen @placements - placeringar @theater - aktuell teater (bild mm) @show - information om föreställningen (datum mm) @seats - stolar med positioner och placeringsstatus (upptagen, tillfälligt låst, ledig, vald för denna bokning)

5.V11 PlacementsController#edit.html

Typ	Vy
Syfte	Administratören ska kunna placera ut bokningar (SR1.11, SR1.25)
Funktion	Administratören ska kunna ändra placeringen av en bokning, så länge som den inte har hämtats ut av kunden.
Delkomponenter	Inga
Beroenden	PlacementsController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@reservation - den aktuella bokningsinstansen @placements - placeringar @theater - aktuell teater (bild mm) @show - information om föreställningen (datum mm) @seats - stolar med positioner och placeringsstatus (upptagen, tillfälligt låst, ledig, vald för denna bokning)

5.V12 TheatersController#index.html

Typ	Vy
Syfte	Administratören ska kunna hantera teatrar (SR1.9)
Funktion	Administratören får en överblick över de teatrar som finns i databasen och vad han kan göra med dem.
Delkomponenter	Inga
Beroenden	TheatersController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@theaters - existerande teatrars attribut

5.V13 TheatersController#new.html

Typ	Vy
Syfte	Administratören ska kunna hantera teatrar (SR1.9)
Funktion	Administratören ska genom ett speciellt gränssnitt kunna skapa en ny teater.
Delkomponenter	AR5.P2
Beroenden	TheatersController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@theater - aktuell teater

5.V14 TheatersController#show.html

Typ	Vy
Syfte	Administratören ska kunna hantera teatrar (SR1.9)
Funktion	Administratören ska kunna se vilka teatrar som finns.
Delkomponenter	Inga
Beroenden	TheatersController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@theater - aktuell teater

5.V15 TheatersController#edit.html

Typ	Vy
Syfte	Administratören ska kunna hantera teatrar (SR1.9)
Funktion	Administratören ska kunna göra ändringar till en teater, till exempel om en sektion är under reparation så ska kunder inte kunna boka platser i den.
Delkomponenter	AR5.P2
Beroenden	TheatersController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@theater - aktuell teater

5.P2 TheatersController#_form.html

Typ	Partiell vy
Syfte	Administratören ska kunna hantera teatrar (SR1.9)
Funktion	Renderar ett formulär för att redigera Theaters-objekt.
Delkomponenter	Inga
Beroenden	TheatersController
Gränssnitt	Ej applicerbart
Resurser	Theater
Källor	Inga
Process	Ej applicerbart
Data	@theater - aktuell teater

5.V16 SectionsController#index.html

Typ	Vy
Syfte	Administratören och säljaren ska kunna välja sektionsplaceringen (SR1.22, SR1.23)
Funktion	Administratören och säljaren ska få en överblick över de sektioner som platser kan placeras ut i och hur detta kan göras.
Delkomponenter	Inga
Beroenden	SectionsController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@sections @theater - aktuell teater

5.V17 SectionsController#new.html

Typ	Vy
Syfte	Administratören och säljaren ska kunna välja sektionsplaceringen (SR1.22)
Funktion	Administratören och säljaren ska kunna välja en sektion att placera en boknings platser i.
Delkomponenter	AR5.P3
Beroenden	SectionsController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@section @theater - aktuell teater

5.V18 SectionsController#show.html

Typ	Vy
Syfte	Administratören och säljaren ska kunna välja sektionsplaceringen (SR1.22, SR1.23)
Funktion	Administratören och säljaren ska kunna se alla sektioner som är valbara.
Delkomponenter	Inga
Beroenden	SectionsController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@theater - aktuell teater

5.V19 SectionsController#edit.html

Typ	Vy
Syfte	Administratören och säljaren ska kunna välja sektionsplaceringen (SR1.22)
Funktion	Administratören ska kunna ändra vilka sektioner som är valbara.
Delkomponenter	AR5.P3
Beroenden	SectionsController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@section @theater - aktuell teater

5.P3 SectionsController#_form.html

Typ	Partiell vy
Syfte	Administratören och säljaren ska kunna välja sektionsplaceringen (SR1.22)
Funktion	Renderar ett formulär för att redigera Sections-objekt.
Delkomponenter	Inga
Beroenden	SectionsController
Gränssnitt	Ej applicerbart
Resurser	Section
Källor	Inga
Process	Ej applicerbart
Data	@section @theater - aktuell teater

5.V20 SeatsController#index.html

Typ	Vy
Syfte	Administratören ska kunna hantera stolsplacering (SR1.11, SR1.25)
Funktion	Administratören ska kunna identifiera och placera ut stolar för en given teater.
Delkomponenter	Inga
Beroenden	SeatsController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@seats @theater - aktuell teater

5.V21 BatchesController#index.html

Typ	Vy
Syfte	Administratören skall kunna lägga till omgångar (SR1.10)
Funktion	Ge en översikt på alla omgångar.
Delkomponenter	Inga
Beroenden	BatchesController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@batches - existerande omgångar

5.V22 BatchesController#new.html

Typ	Vy
Syfte	Administratören skall kunna lägga till omgångar (SR1.10)
Funktion	Skapa en ny omgång.
Delkomponenter	AR5.P4
Beroenden	BatchesController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@batch - aktuell omgång

5.V23 BatchesController#show.html

Typ	Vy
Syfte	Administratören skall kunna lägga till omgångar (SR1.10)
Funktion	Visa information om en omgång, till exempel alla föreställningar som hör till omgången.
Delkomponenter	Inga
Beroenden	BatchesController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@batch - aktuell omgång

5.V24 BatchesController#edit.html

Typ	Vy
Syfte	Administratören skall kunna lägga till omgångar (SR1.10)
Funktion	Ändra omgången och vilka föreställningar som ingår i den.
Delkomponenter	AR5.p4
Beroenden	BatchesController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@batch - aktuell omgång

5.P4 BatchesController#_form.html

Typ	Partiell vy
Syfte	Administratören skall kunna lägga till omgångar (SR1.10)
Funktion	Renderar ett formulär för att redigera Batches-objekt.
Delkomponenter	Inga
Beroenden	BatchesController
Gränssnitt	Ej applicerbart
Resurser	Batch
Källor	Inga
Process	Ej applicerbart
Data	@batch - aktuell omgång

5.V25 ShowsController#index.html

Typ	Vy
Syfte	Administratören skall kunna hantera föreställningar (SR1.10, SR1.20)
Funktion	Ger översikt på alla föreställningar som hör till en vald omgång.
Delkomponenter	Inga
Beroenden	ShowsController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@shows @batch - aktuell omgång @batches - existerande omgångar

5.V26 ShowsController#new.html

Typ	Vy
Syfte	Administratören skall kunna hantera föreställningar (SR1.10)
Funktion	Lägga till nya föreställningar till en omgång.
Delkomponenter	AR5.P5
Beroenden	ShowsController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@show @batch - aktuell omgång @batches - existerande omgångar

5.V27 ShowsController#show.html

Typ	Vy
Syfte	Administratören skall kunna hantera föreställningar (SR1.10)
Funktion	Visa information om en vald föreställning, till exempel antal lediga platser.
Delkomponenter	Inga
Beroenden	ShowsController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@batch - aktuell omgång @batches - existerande omgångar

5.V28 ShowsController#edit.html

Typ	Vy
Syfte	Administratören skall kunna hantera föreställningar (SR1.10)
Funktion	Ändra information på en vald föreställning, till exempel datum.
Delkomponenter	AR5.P5
Beroenden	ShowsController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@show @batch - aktuell omgång @batches - existerande omgångar

5.P5 ShowsController#_form.html

Typ	Partiell vy
Syfte	Administratören skall kunna hantera föreställningar (SR1.10)
Funktion	Renderar ett formulär för att redigera Shows-objekt.
Delkomponenter	Inga
Beroenden	ShowsController
Gränssnitt	Ej applicerbart
Resurser	Show
Källor	Inga
Process	Ej applicerbart
Data	@show @batch - aktuell omgång @batches - existerande omgångar

5.V29 PricingsController#index.html

Typ	Vy
Syfte	Låter administratören hantera priserna för en omgång (SR1.30).
Funktion	Visar ett formulär där man kan välja priser för alla sektioner (för den aktuella omgången).
Delkomponenter	Inga
Beroenden	PricingsController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@batch - aktuell omgång @sections - sektioner i omgångens teater @pricings - priser för varje sektion- och rabattklasskombination

5.V30 UsersController#index.html

Typ	Vy
Syfte	Administratören ska kunna hantera systemets användarkonton (SR1.7, SR1.8)
Funktion	Administratören får en överblick över de användare som finns i systemet och deras rättigheter.
Delkomponenter	Inga
Beroenden	UserController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@users

5.V31 UsersController#new.html

Typ	Vy
Syfte	Administratören ska kunna hantera systemets användarkonton (SR1.7, SR1.8)
Funktion	Administratören ska kunna skapa en ny användare och sätta dennes rättigheter i systemet.
Delkomponenter	AR5.P6
Beroenden	UserController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@user

5.V32 UsersController#show.html

Typ	Vy
Syfte	Administratören ska kunna hantera systemets användarkonton (SR1.7, SR1.8)
Funktion	Administratören ska kunna granska en enskild användare och dennes rättigheter.
Delkomponenter	Inga
Beroenden	UserController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@user

5.V33 UsersController#edit.html

Typ	Vy
Syfte	Administratören ska kunna hantera systemets användarkonton (SR1.7, SR1.8)
Funktion	Administratören ska kunna ändra en användares rättigheter eller lösenord, till exempel säljare som inte ska säljare längre.
Delkomponenter	AR5.P6
Beroenden	UserController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@user

5.P6 UsersController#_form.html

Typ	Partiell vy
Syfte	Administratören ska kunna hantera systemets användarkonton (SR1.7, SR1.8)
Funktion	Renderar ett formulär för att redigera User-objekt.
Delkomponenter	Inga
Beroenden	UserController
Gränssnitt	Ej applicerbart
Resurser	User
Källor	Inga
Process	Ej applicerbart
Data	@user

5.V34 MailingController#index.html

Typ	Vy
Syfte	Administratören ska kunna hantera mailutskick (SR1.14)
Funktion	Ge en översikt på bland annat sparade mailmallar.
Delkomponenter	Inga
Beroenden	MailingController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@mailing @reciepts = session[:reciepts] - mailmottagare (semipermanent) @mail = session[:mail] - den aktuella mailmallen, kan vara temporär för att möjliggöra utskick utan mall

5.V35 MailingController#new.html

Typ	Vy
Syfte	Administratören ska kunna hantera mailutskick (SR1.14)
Funktion	Skapa en ny mailmall för att skicka till en eller flera mottagare.
Delkomponenter	AR5.P7
Beroenden	MailingController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@mail @reciepts = session[:reciepts] - mailmottagare (semipermanent) @mail = session[:mail] - den aktuella mailmallen, kan vara temporär för att möjliggöra utskick utan mall

5.V36 MailingController#show.html

Typ	Vy
Syfte	Administratören ska kunna hantera mailutskick (SR1.14, SR1.15)
Funktion	Visa en mailmall som sparats.
Delkomponenter	Inga
Beroenden	MailingController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@mail @reciepts = session[:reciepts] - mailmottagare (semipermanent) @mail = session[:mail] - den aktuella mailmallen, kan vara temporär för att möjliggöra utskick utan mall

5.V37 MailingController#edit.html

Typ	Vy
Syfte	Administratören ska kunna hantera mailutskick (SR1.14)
Funktion	Innehåll i en befintlig mailmall.
Delkomponenter	AR5.P7
Beroenden	MailingController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@mail @reciepts = session[:reciepts] - mailmottagare (semipermanent) @mail = session[:mail] - den aktuella mailmallen, kan vara temporär för att möjliggöra utskick utan mall

5.V38 MailingController#preview.html

Typ	Vy
Syfte	Administratören ska kunna hantera mailutskick (SR1.14)
Funktion	Förhandsgranska de mail som ska skickas innan utskick.
Delkomponenter	Inga
Beroenden	MailingController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@mail @reciepts = session[:reciepts] - mailmottagare (semipermanent) @mail = session[:mail] - den aktuella mailmallen, kan vara temporär för att möjliggöra utskick utan mall

5.P7 MailingController#_form.html

Typ	Partiell vy
Syfte	Administratören ska kunna hantera mailutskick (SR1.14)
Funktion	Renderar ett formulär för att redigera Mailing-objekt.
Delkomponenter	Inga
Beroenden	MailingController
Gränssnitt	Ej applicerbart
Resurser	Mail
Källor	Inga
Process	Ej applicerbart
Data	@mail @reciepts = session[:reciepts] - mailmottagare (semipermanent) @mail = session[:mail] - den aktuella mailmallen, kan vara temporär för att möjliggöra utskick utan mall

5.3.4 Säljare**5.V39 SalesController#index.html**

Typ	Vy
Syfte	En säljare ska kunna genomföra ett kontantköp och lämna ut biljetter (SR1.5, SR1.6, SR1.25)
Funktion	Visa en översikt på säljarens möjliga sysslor, kontantköp och utlämning av bokade biljetter.
Delkomponenter	Inga
Beroenden	SalesController
Gränssnitt	Ej applicerbart
Resurser	5.L2 sales_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@step - steg i bokning @steps - array med vilka steg som finns @reservation = session[:reservation] - bokningsdata

5.V40 SalesController#show_choice.html

Typ	Vy
Syfte	En säljare ska kunna genomföra ett kontantköp och lämna ut biljetter (SR1.5, SR1.6, SR1.25, SR1.21)
Funktion	Välja vilken föreställning som ärendet gäller.
Delkomponenter	Inga
Beroenden	SalesController
Gränssnitt	Ej applicerbart
Resurser	5.L2 sales_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@step - steg i bokning @steps - array med vilka steg som finns @reservation = session[:reservation] - bokningsdata

5.V41 SalesController#seats.html

Typ	Vy
Syfte	En säljare ska kunna genomföra ett kontantköp och lämna ut biljetter (SR1.5, SR1.24, SR1.25)
Funktion	Välja vilken plats som ärendet gäller.
Delkomponenter	Inga
Beroenden	Om administratören inte har placerat ut den redan bokade platsen så gör säljaren det, men sektionsvalet är redan låst. Om det är ett nytt köp, placerar säljaren också ut platsen. SalesController
Gränssnitt	Ej applicerbart
Resurser	5.L2 sales_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@step - steg i bokning @steps - array med vilka steg som finns @reservation = session[:reservation] - bokningsdata

5.V42 SalesController#confirm.html

Typ	Vy
Syfte	En säljare ska kunna genomföra ett kontantköp och lämna ut biljetter (SR1.5, SR1.6, SR1.26)
Funktion	Fråga användaren om de angivna uppgifterna är korrekta och skall införas i systemet.
Delkomponenter	Inga
Beroenden	SalesController
Gränssnitt	Ej applicerbart
Resurser	5.L2 sales_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@step - steg i bokning @steps - array med vilka steg som finns @reservation = session[:reservation] - bokningsdata

5.V43 SalesController#retrieve.html

Typ	Vy
Syfte	En säljare ska kunna genomföra ett kontantköp och lämna ut biljetter (SR1.5, SR1.6)
Funktion	Uppmana säljaren att gå och hämta en viss biljett.
Delkomponenter	Inga
Beroenden	SalesController
Gränssnitt	Ej applicerbart
Resurser	5.L2 sales_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@step - steg i bokning @steps - array med vilka steg som finns @reservation = session[:reservation] - bokningsdata

5.V44 SalesController#payment.html

Typ	Vy
Syfte	En säljare ska kunna genomföra ett kontantköp och lämna ut biljetter (SR1.5, SR1.6)
Funktion	Uppmana säljaren att ta betalt för bokningen.
Delkomponenter	Inga
Beroenden	SalesController
Gränssnitt	Ej applicerbart
Resurser	5.L2 sales_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@step - steg i bokning @steps - array med vilka steg som finns @reservation = session[:reservation] - bokningsdata

5.V45 SalesController#hand_out.html

Typ	Vy
Syfte	En säljare ska kunna genomföra ett kontantköp och lämna ut biljetter (SR1.5, SR1.6)
Funktion	Uppmana säljaren att lämna ut biljetten.
Delkomponenter	Inga
Beroenden	SalesController
Gränssnitt	Ej applicerbart
Resurser	5.L2 sales_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@step - steg i bokning @steps - array med vilka steg som finns @reservation = session[:reservation] - bokningsdata

5.V46 SalesController#search.html

Typ	Vy
Syfte	En säljare ska kunna genomföra ett kontantköp och lämna ut biljetter (SR1.5, SR1.6, SR1.25, SR1.20)
Funktion	Låter säljaren söka bland bokningar med exempelvis namn.
Delkomponenter	Inga
Beroenden	SalesController
Gränssnitt	Ej applicerbart
Resurser	5.L2 sales_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@step - steg i bokning @steps - array med vilka steg som finns @reservation = session[:reservation] - bokningsdata

5.3.5 Kund**5.V47 BookingController#show_choice.html**

Typ	Vy
Syfte	Kunden ska kunna boka sin biljett via Kårspexets webbsida (SR1.2, SR1.21)
Funktion	Kunden får ett val om vilken föreställning.
Delkomponenter	Inga
Beroenden	BookingController
Gränssnitt	Ej applicerbart
Resurser	5.L4 customer_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@step - steg i bokning @steps - array med vilka steg som finns @reservation - bokningsdata

5.V48 BookingController#section.html

Typ	Vy
Syfte	Kunden ska kunna boka sin biljett via Kårspexets webbsida (SR1.2, SR1.24)
Funktion	Kunden väljer vilken sektion som denne vill boka på den valda föreställningen.
Delkomponenter	Inga
Beroenden	BookingController
Gränssnitt	Ej applicerbart
Resurser	5.L4 customer_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@step - steg i bokning @steps - array med vilka steg som finns @reservation - bokningsdata

5.V49 BookingController#payment.html

Typ	Vy
Syfte	Kunden ska kunna boka sin biljett via Kårspexets webbsida (SR1.2, SR1.4 i delux)
Funktion	Kunden fyller i sin betalningsinformation och hur kunden vill hämta sin biljett.
Delkomponenter	Inga
Beroenden	BookingController
Gränssnitt	Ej applicerbart
Resurser	5.L4 customer_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@step - steg i bokning @steps - array med vilka steg som finns @reservation - bokningsdata

5.V50 BookingController#confirm.html

Typ	Vy
Syfte	Kunden ska kunna boka sin biljett via Kårspexets webbsida (SR1.2, SR1.26)
Funktion	Kunden bekräftar bokningen och alla betalningsuppgifter.
Delkomponenter	Inga
Beroenden	BookingController
Gränssnitt	Ej applicerbart
Resurser	5.L4 customer_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@step - steg i bokning @steps - array med vilka steg som finns @reservation - bokningsdata

5.V51 BookingController#cancel_booking.html

Typ	Vy
Syfte	Kunden ska kunna avboka sin biljett via Kårspexets webbsida (SR1.2, SR1.3)
Funktion	Kunden avbokar sin bokning via en länk i sitt bekräftelsemail.
Delkomponenter	Inga
Beroenden	BookingController
Gränssnitt	Ej applicerbart
Resurser	Inga
Källor	Inga
Process	Ej applicerbart
Data	@step - steg i bokning @steps - array med vilka steg som finns @reservation - bokningsdata

5.V52 BookingController#destroy.html

Typ	Vy
Syfte	Kunden ska kunna avboka sin biljett via Kårspexets webbsida (SR1.2, SR1.3)
Funktion	Kunden får en bekräftelse att bokningen är avbokad.
Delkomponenter	Inga
Beroenden	BookingController
Gränssnitt	Ej applicerbart
Resurser	5.L4 customer_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@step - steg i bokning @steps - array med vilka steg som finns @reservation - bokningsdata

5.4 Övriga komponenter

5.X1 InheritedResources

Typ	Övrig komponent (Rails-gem)
Syfte	Abstraherar bort hantering av resurser för kontroller som hanterar dessa för att minimera kodupprepning.
Funktion	Tillhandahåller alla de (eller valfria) standardactions för kontroller som hanterar resurser och ärver från denna klass.
Delkomponenter	Inga
Beroenden	Inga
Gränssnitt	<p>Alla kontrollers som ärver från denna klass får standardactions för resurser (<code>index</code>, <code>show</code>, <code>new</code>, <code>create</code>, <code>edit</code>, <code>update</code> samt <code>destroy</code>) fördefinierade om inget annat anges.</p> <p><code>actions</code> - Hjälpmetod för att specificera vilka actions som ska fördefinieras.</p> <p>Hjälpmetoder som görs tillgängliga i vyer:</p> <p><code>resource</code> - Enskild instans av den representerade resursen (i <code>show</code>, <code>new</code>, <code>create</code>, <code>edit</code>, <code>update</code>, <code>destroy</code>).</p> <p><code>collection</code> - En array innehållandes alla objekt av resursen (endast i <code>index</code>).</p> <p><code>resource_class</code> - En referens till den modell (klassdefinitionen) som hanteras som resurs.</p>
Resurser	<code>InheritedResources::Base</code> (ingår i Rails-gemet <code>inherited_resources</code>).
Källor	<code>inherited_resources</code> : https://github.com/josevalim/inherited_resources
Process	Ej applicerbart
Data	Instansvariabler av formatet <code>@modellnamn</code> (ex. <code>@reservation</code> för <code>ReservationsController</code>) respektive <code>@modellnamn_i_plural</code> för <code>index-actions</code> (ex. <code>@reservations</code>).

5.X2 AuthLogic

Typ	Övrig komponent (Rails-gem)
Syfte	Hanterar autentisering och användarsessioner (SR1.1).
Funktion	Tillhandahåller funktioner för att autentisera användare utifrån existerande modeller.
Delkomponenter	UserSession.
Beroenden	Inga
Gränssnitt	Authlogic::Session::Base#find - returnerar aktiv användarsession eller försöker initiera en ny om ingen redan finns ActiveRecord::Base#acts_as_authentic - instruerar Authlogic att den aktuella modellen representerar en typ av autentisering
Resurser	Authlogic (ingår i Rails-gemet authlogic), UserSession och User.
Källor	authlogic: https://github.com/binarylogic/authlogic
Process	Ej applicerbart
Data	Inga

5.X3 CanCan

Typ	Övrig komponent (Rails-gem)
Syfte	Definierar och hanterar användarroller (SR9.2).
Funktion	Tillhandahåller metoder för att bestämma vilka användarroller som finns samt vad dessa har rättigheter till. Definierar även hjälpmetoder för att testa rättigheter mot den inloggade användaren.
Delkomponenter	Ability
Beroenden	En metod <code>ApplicationController#current_user</code> måste definieras så att den returnerar ett objekt.
Gränssnitt	<p><code>can?</code> - returnerar huruvida användaren har rättigheter för att utföra en viss handling (tillgänglig i vyer och kontroller)</p> <p><code>cannot?</code> - inversen av <code>can?</code></p> <p><code>ApplicationController#authorize!</code> - kastar en exception om användaren inte har rättighet att utföra den handling som angivits.</p> <p><code>ApplicationController#load_and_authorize_resource</code> - ladda in och autorisera alla standardactions för resurser automatiskt</p>
Resurser	CanCan (ingår i Rails-gemet <code>cancan</code>) och Ability.
Källor	<code>cancan</code> : https://github.com/ryanb/cancan
Process	<ol style="list-style-type: none"> 1. Rättigheter fördefinieras i Ability-modellen. 2. <code>can?(:action, Object)</code> anropas (:action = någon fördefinierad action, Object är någon modell alternativt en modellinstans). 3. CanCan hämtar aktiv användare via <code>ApplicationController#current_user</code>. 4. Användarens rättigheter undersöks av CanCan genom att skicka denna till Ability-modellen. 5. <code>can?</code>-metoden returnerar huruvida rättigheter finns (eller saknas) för att utföra handlingen.
Data	Inga

5.X4 ActiveRecord

Typ	Övrig komponent (Rails)
Syfte	ActiveRecord är en existerande baskomponent i Rails-ramverket som bland annat fungerar som ett gränssnitt till databasens tabeller (SR1.29).
Funktion	Tillhandahåller gränssnitt för sökningar och manipulation av objekt databasen. Implicit ingår databas och databasschema i varje modell.
Delkomponenter	Inga
Gränssnitt	<p>Objektorienterat gränssnitt enligt active-record-patternen. En modell är en klass som ärver från ActiveRecord::Base och motsvarar en tabell i databasen. En instans av klassen motsvarar en rad i tabellen.</p> <p>Följande är några av de metoder som definieras av ActiveRecord::Base och är tillgängliga för alla modeller.</p> <p>all - hämta alla instanser av modellen (alla rader i databastabellen).</p> <p>find - gör diverse sökningar i databasen och returnerar de instanser som matchar.</p> <p>valid? - kontrollera att objektets data är korrekt. Vilka kontroller som görs definieras av varje modell.</p> <p>save - lagrar/uppdaterar en instans av ett modell.</p> <p>update - uppdaterar attributer för en instans i databastabellen.</p> <p>destroy - raderar objektet i databasen.</p> <p>Dessutom finns accessors för varje attribut hos modellen. Dessa definieras dynamiskt.</p>
Beroenden	Inga
Resurser	Inga
Källor	ActiveRecord: http://ar.rubyonrails.org
Process	<p>Följande typscenariot visar skapande och uppdatering av en instans av modellen M.</p> <pre> m = M.new m.attr1 = 'value' if m.valid? then m.save </pre>
Data	<p>Alla modeller har datafälten created_at respektive updated_at av typen timestamp. Dessa uppdateras automatiskt av Rails vid skapande och uppdatering av objektet.</p> <p>Alla modeller har även attributet id som primärnyckel.</p> <p>För övriga attribut se respektive modell. Enkla attribut listas, relationer till andra modeller beskrivs i delkomponenter.</p>

6 Genomförbarhet- och resursuppskattning

6.1 Sammanfattning av slutsatser av en genomförbarhetsundersökning av arkitekturmodellen.

Nyx kommer att följa den så kallade MVC-principen (Model-View-Controller). Alla komponenter under respektive punkt finns angivna nedan. I designen ser vi endast två delar som kommer att vara mer komplicerade, nämligen teaterdesignen och stolsplaceringen. Dessa är dock fullt genomförbara, eftersom det inte är avancerade tekniker som används, utan komplexiteten härstammar främst från att det är mycket data som ska redigeras samtidigt. Inga större förändringar av arkitekturen förutses för de komplicerade delarna nämnda ovan. Slutsatsen är således att projektet är fullt genomförbart med den här arkitekturen.

6.2 Komponentprioritering

6.2.1 Modeller

Komponent	Prioritet
5.M1 Reservation	Standard
5.M2 ReservationCounter	Standard
5.M3 Placement	Standard
5.M4 Theater	Standard
5.M5 Section	Standard
5.M6 Seat	Standard
5.M7 Batch	Standard
5.M8 Show	Standard
5.M9 Pricing	Standard
5.M10 User	Standard
5.M11 MailTemplate	Standard
5.M12 UserSession	Standard
5.M13 Ability	Standard

6.2.2 Kontroller

Komponent	Prioritet
5.C1 ApplicationController	Standard
5.C2 BookingController	Standard
5.C3 SessionController	Standard
5.C4 AdminIndexController	Standard
5.C5 ReservationsController	Standard
5.C6 PlacementsController	Standard
5.C7 TheatersController	Standard
5.C8 SectionsController	Standard
5.C9 SeatsController	Standard
5.C10 BatchesController	Standard
5.C11 ShowsController	Standard
5.C12 PricingsController	Standard
5.C13 UsersController	Standard
5.C14 MailingController	Standard
5.C15 SalesController	Standard

6.2.3 Vyer

Komponent	Prioritet
5.L1 admin_layout.html	Standard
5.L2 sales_layout.html	Standard
5.L3 simple_layout.html	Standard
5.L4 customer_layout.html	Standard
5.V1 SessionController#new.html	Standard
5.V2 AdminIndexController#index.html	Plus
5.V3 AdminIndexController#stat_standard.html	Standard
5.V4 AdminIndexController#stat_plus.html	Plus
5.V5 AdminIndexController#stat_deluxe.html	Deluxe
5.V6 ReservationsController#index.html	Standard
5.V7 ReservationsController#new.html	Standard
5.V8 ReservationsController#show.html	Standard
5.V9 ReservationsController#edit.html	Standard
5.P1 ReservationsController#_form.html	Standard
5.V10 PlacementsController#new.html	Standard
5.V11 PlacementsController#edit.html	Standard
5.V12 TheatersController#index.html	Standard
5.V13 TheatersController#new.html	Standard
5.V14 TheatersController#show.html	Standard
5.V15 TheatersController#edit.html	Standard
5.P2 TheatersController#_form.html	Standard
5.V16 SectionsController#index.html	Standard
5.V17 SectionsController#new.html	Standard
5.V18 SectionsController#show.html	Standard
5.V19 SectionsController#edit.html	Standard
5.P3 SectionsController#_form.html	Standard
5.V20 SeatsController#index.html	Standard
5.V21 BatchesController#index.html	Standard
5.V22 BatchesController#new.html	Standard
5.V23 BatchesController#show.html	Standard
5.V24 BatchesController#edit.html	Standard
5.P4 BatchesController#_form.html	Standard
5.V25 ShowsController#index.html	Standard
5.V26 ShowsController#new.html	Standard
5.V27 ShowsController#show.html	Standard

Komponent	Prioritet
5.V28 ShowsController#edit.html	Standard
5.P5 ShowsController#_form.html	Standard
5.V29 PricingsController#index.html	Standard
5.V30 UsersController#index.html	Standard
5.V31 UsersController#new.html	Standard
5.V32 UsersController#show.html	Standard
5.V33 UsersController#edit.html	Standard
5.P6 UsersController#_form.html	Standard
5.V34 MailingController#index.html	Standard
5.V35 MailingController#new.html	Standard
5.V36 MailingController#show.html	Standard
5.V37 MailingController#edit.html	Standard
5.V38 MailingController#preview.html	Standard
5.P7 MailingController#_form.html	Standard
5.V39 SalesController#index.html	Standard
5.V40 SalesController#show_choice.html	Standard
5.V41 SalesController#seats.html	Standard
5.V42 SalesController#confirm.html	Standard
5.V43 SalesController#retrieve.html	Standard
5.V44 SalesController#payment.html	Standard
5.V45 SalesController#hand_out.html	Standard
5.V46 SalesController#search.html	Standard
5.V47 BookingController#show_choice.html	Standard
5.V48 BookingController#section.html	Standard
5.V49 BookingController#payment.html	Standard
5.V50 BookingController#confirm.html	Standard
5.V51 BookingController#cancel_booking.html	Standard
5.V52 BookingController#destroy.html	Standard

6.2.4 Övriga komponenter

Komponent	Prioritet
5.X1 InheritedResources	Standard
5.X2 AuthLogic	Standard
5.X3 CanCan	Standard
5.X4 ActiveRecord	Standard

6.3 Framtida projektuppgifter med tidsuppskattning

Nr	Benämning	Beskrivning	Tidsuppskattning
1	Modellskelett	Skelett inklusive relationer och attribut	5h
2	Skelett för kontroller	Definiera tomma actions	5h
3	Routes	Skapa standardroutes för resurser	2h
4	Kontroller-logik	Logik för kontroller som hanterar resurser med InheritedResources	10h
5	Redigeringsvyer	Grundläggande vyer (formulär) för att redigera omgångar/föreställningar	10h
6	Indexvyer	Enkla vyer för index av omgång/föreställning	3h
7	Teatervyer	Grundläggande versioner av vyer för administrering av teatrar (enbart attribut)	3h
8	Sektionsredigering	Vyer för att redigera sektioner (inklusive polygonkoordinater)	8h
9	Platsgränssnitt	Vy och gränssnittslogik för utplacering av platser i teatrar	15h
10	Adminlayout	Designa och implementera administratörslayouten	8h
11	Bokningslistning	Listning och en enkel filtrering av bokningar	10h
12	Bokningsredigering	Administratörsvy för att redigering av bokningar	5h
13	Bokningssteg	Implementera grundläggande stegnavigering med databeständighet	20h
14	Kundbokningslayout	Layouter för kundens bokningsprocess	10h
15	Bokningsvyer	Vyer och gränssnittslogik för bokningsprocedurens olika steg	25h
16	Placeringsgränssnitt	Gränssnitt för placering av bokningar	20h
17	Prishantering	Hantering av priser (formulär och generering)	15h
18	Omgångs/föreställningskrav	Validerings- och övriga krav på omgångar, föreställningar och prissättningar	10h
19	Teaterkrav	Validerings- och integritetskrav för teatrar, sektioner och platser	15h

Nr	Benämning	Beskrivning	Tidsuppskattning
20	Bokningskrav	Krav för bokningar (Reservation, ReservationCounter)	10h
21	Placeringskrav	Integritetskrav och övrig logik för placeringar och dess relationer	10h
22	Bokningsfilter	Avancerat filter av bokningar	10h
23	Mailmallar	Administreringsvy för mailmallar(skapa/redigera/ta bort)	10h
24	MailingController	MailingController - resurshantering	3h
25	Mailgenerering	Generering av mail utifrån bokningar	15h
26	Mailutskick	Processen från bokningslistan till utskicken med data inlagd i vald mall	15h
27	Förhandsgranskning	Förhandsgranskning av mailutskick - vy och logik	10h
28	User-modellen	Krav på attribut och roll-testmetoder	5h
29	Ability-klassen	Ability-klassen med roll-logik och rättighetsdefinitioner (med CanCan som grund)	10h
30	Autentisering	Autentisering i ApplicationController, UserSession-modellen (Authlogic) samt SessionController	10h
31	Simplelayouten	Design och implementation av layouten för bland annat inloggning	2h
32	Inloggningsvy	Inloggningsvy med formulär	3h
33	UsersController	Administrering av användare	5h
34	Användarvyer	Vyer för användaradministrering	6h
35	Lösenordsgenerering	Lösenordsgenerering för användare	2h
36	Användaravstängning	Hindra avstängda användare från att logga in	1h
37	Säljarvyer1	Vyer för placerade, färdiga bokningar (uthämtningsproceduren)	5h
38	SalesController1	Logik för uthämtning/betalning av existerande bokningar	15h
39	Säljarvyer2	Vyer för kontantförsäljning (inklusive platsfördelning) samt utplacering	15h
40	SalesController2	Logik för kontantköp och utplacering av platser	15h
41	AdminIndexController	Översiktssidan (action och vy)	5h
42	Statistik	Logik (förberedande och bearbetande av statistikdata) i AdminIndexController	10h
43	Statistikvyer	Vy för presentation av statistik	5h
44	Gränssnittsbegränsning	Infoga logik i Admin-layouten för att begränsa gränssnittet för ekonomiker	2h

6.4 Beroenden mellan uppgifter

Vad gäller komponenterna och då speciellt vyer kan man se beroenden mellan vy och kontroller på följande sätt; AdminIndexController#new.html har AdminIndexController som beroende.

Uppgift	Beroende av uppgift
1	Inga
2	Inga
3	2
4	1, 2
5	Inga
6	Inga
7	Inga
8	Inga
9	1
10	Inga
11	1
12	4
13	1
14	Inga
15	14
16	9
17	4
18	1
19	1
20	1
21	1
22	11
23	1
24	Inga
25	24
26	25
27	25
28	Inga
29	28
30	29, 28
31	Inga
32	Inga
33	28
34	28
35	33, 28
36	33, 28

Uppgift	Beroende av uppgift
37	Inga
38	Inga
39	Inga
40	Inga
41	Inga
42	41
43	42
44	10, 30

6.5 Uppgiftsplanering

Siffrorna nedan korresponderar med uppgifterna ovan.

Namn	Vecka 10	Vecka 11	Vecka 12	Vecka 13	Vecka 14
Victor H	1,2,3,4	10,11	13	22,28	29,30
Andreas T	5,6	17	15	24,25	26,27
Rasmus G	9	9	15	16	16
Erik E	19	12	17	38	33
Per H	7,8	14	23	31,32	34,37,43
Kalle A	18	19	20,21	41,42	43
André G			Utvecklar gränssnitt	Utvecklar gränssnitt	Utvecklar gränssnitt
Peter B	Utvecklar tester	Utvecklar tester	Testar	Testar	Testar
Johan S	Synkar uppstart	Samordnar, resurs	Resurs	Testar	Resurs
Anna Maria M	Utvecklar tester	Utvecklar tester	Testar	Testar	Testar
Daniel N	Utvecklar tester	Utvecklar tester	Utvecklar gränssnitt	Utvecklar gränssnitt	Testar

Namn	Vecka 15	Vecka 16	Vecka 17	Vecka 18	Vecka 19
Victor H	35,36	44	Buggfixar	Finputsar	Demo
Andreas T	27		Buggfixar	Finputsar	Installerar
Rasmus G	39,40		Testar	Testar	Testar
Erik E	40			Testar	Testar
Per H				Testar	Demo
Kalle A				Testar	Demo
André G	Utvecklar gränssnitt	Testar	Testar	Skriver samman- fattning	Skriver samman- fattning
Peter B	Testar	Testar	Testar	Testar	Testar
Johan S	Testar	Testar	Samordnar, resurs	Testar	Demo
Anna Maria M	Testar	Testar	Testar	Testar	Testar
Daniel N	Testar	Testar	Testar	Testar	Testar

6.6 Möjliga risker

Projektplaneringen visar att tiden kommer att räcka för att slutföra projektet och hittills har samarbetet fungerat väl. Nyx har ingen nyckelkompetens som gör oss beroende av en enstaka projektmedlem. Den största risken ligger i att Kårspexet inte anser kraven uppfyllda eller kommer med ändringar i kraven. Då kommer det vara ont om tid för att korrigera dessa och det ger oss ingen marginal för fel. Vi kommer undvika detta genom ett nära samarbete med Kårspexet under implementationen. De kommer dessutom att ha en viktig roll under testningen som skall inledas i god tid.

I övrigt har vi identifierat följande risker:

» Risk att vi implementerar fel funktionalitet (annat än det efterfrågade) och därmed förlorar tid. Det kommer vi undvika genom kontinuerlig kontakt med Kårspexet, om frågor uppstår så kommer de att diskuteras igenom grundligt, ingen gissning kommer pågå, samt att genomgående under implementation återgå till kraven i URD och SRD.

» Risk att Nyx inte hinner implementera alla funktioner som efterfrågas. För att undvika den här situationen har vi en tidsplanering med god marginal för oförutsedda uppgifter.

» Risk att vi implementerar saker i fel ordning så att vi förlorar tid på att vänta på varandra. Här kommer nyckelkompetens in i beräkningarna. Återigen ligger lösningen i en bra planering och bra kontroll på beroenden mellan kontrollerna.

» Risk att vår utvecklingsserver krånglar. Det som skulle hända är att vi inte har en gemensam git-server att skicka uppdaterat arbete till. Det är i sig inte ett stort problem, alla utvecklare har lokala versioner och kan fortsätta jobba tills servern kommer upp igen.

» Risk att det krånglar när vi ska installera vårt system på Kårspexets server. För att hitta eventuella problem i tid kommer vi tidigt installera en demo av bokningssystemet. På så sätt hinner vi korrigera felen i tid för deadline.

7 Spårningsmatris

Följande tabell listar mjukvarukrav från SRD:n. För en enklare överblick över vilka mjukvarukrav ett specifikt systemkrav härstammar ifrån, se de specifika kraven i sektion 5.

7.1 Funktionella krav

Kravkälla	Krav
SR1.1	AR.M12, AR.M13, AR.C1, AR.C3, AR.L3, AR.V1, AR.X2
SR1.2	AR.M1, AR.M2, AR.C2, AR.L4, AR.V47, AR.V48, AR.V49, AR.V50, AR.V51, AR.V52
SR1.3	AR.M1, AR.C2, AR.V51, AR.V52
SR1.4	AR.V49
SR1.5	AR.M3, AR.C15, AR.L2, AR.V39, AR.V40, AR.V41, AR.V42, AR.V43, AR.V44, AR.V45, AR.V46
SR1.6	AR.C15, AR.V39, AR.V40, AR.V42, AR.V43, AR.V44, AR.V45, AR.V46
SR1.7	AR.M10, AR.C13, AR.V30, AR.V31, AR.V32, AR.V33, AR.P6
SR1.8	AR.C13, AR.V30, AR.V31, AR.V32, AR.V33, AR.P6
SR1.9	AR.M4, AR.M5, AR.M6, AR.C7, AR.C8, AR.C9, AR.V12, AR.V13, AR.V14, AR.V15, AR.P2
SR1.10	AR.M7, AR.M8, AR.C10, AR.C11, AR.V21, AR.V22, AR.V23, AR.V24, AR.V25, AR.V26, AR.V27, AR.V28, AR.P4, AR.P5
SR1.11	AR.M3, AR.C6, AR.V10, AR.V11, AR.V20
SR1.12	AR.C5, AR.V6, AR.V9
SR1.13	AR.C5, AR.V6, AR.V7, AR.V8, AR.V9, AR.P1
SR1.14	AR.C14, AR.V34, AR.V35, AR.V36, AR.V37, AR.V38, AR.P7
SR1.15	AR.C14, AR.V36
SR1.16	AR.C4, AR.V3
SR1.17	AR.C4, AR.V4
SR1.18	AR.C4, AR.V5
SR1.19	AR.L1, AR.V2
SR1.20	AR.V25, AR.V46
SR1.21	AR.V40, AR.V47
SR1.22	AR.M4, AR.M5, AR.V16, AR.V17, AR.V18, AR.V19, AR.P3
SR1.23	AR.M5, AR.V16, AR.V18
SR1.24	AR.M5, AR.V41, AR.V48
SR1.25	AR.M6, AR.C15, AR.L2, AR.V10, AR.V11, AR.V20, AR.V39, AR.V40, AR.V41, AR.V46
SR1.26	AR.V42, AR.V50
SR1.27	AR.M11, AR.C14
SR1.28	AR.L1, AR.L2, AR.L3
SR1.29	AR.X4
SR1.30	AR.M9, AR.C12, AR.V29
SR1.31	Tidsmätning loggas automatiskt av Rails.

7.2 Icke-funktionella krav

Kravkälla	Krav
SR9.1	AR.M10
SR9.2	AR.M10, AR.M13, AR.L3, AR.V1, AR.X3