



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 0.1
February 28, 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. A full, highly detailed description is given for all the system's components, from its models, views, and controllers, to its layouts, partials and external components. 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
0.1	Första sammanställd version av dokumentet.

Dokumentversioner

Dokumentet har genererats från följande deldokument.

ADD/abstract version: 4.

ADD/Ändringslogg version: 3.

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: 13.

ADD/Introduktion/Källor version: 5.

ADD/Introduktion/Dokumentöversikt version: 9.

ADD/Systemöverblick version: 8.

ADD/Systemkontext version: 9.

ADD/Systemdesign version: 2.

ADD/Systemdesign/Designmetod version: 19.

ADD/Systemdesign/Uppdelningsbeskrivning version: 11.

ADD/Komponentbeskrivning version: 19.

ADD/Komponentbeskrivning/Model version: 62.

ADD/Komponentbeskrivning/Controller version: 24.

ADD/Komponentbeskrivning/View version: 47.

ADD/Komponentbeskrivning/Övriga_komponenter version: 15.

ADD/Genomförbarhet- och resursuppskattning version: 22.

ADD/Spårningsmatris mellan mjukvarukrav och strukturella krav version: 2.

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	5
1.1	Syfte	5
1.2	Mjukvarans omfattning	5
1.3	Definitioner akronymer och förkortningar	5
1.4	Källor	8
1.5	Dokumentöversikt	8
2	Systemöverblick	9
3	Systemkontext	10
3.1	MySQL	10
3.2	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	15
5	Komponentbeskrivning	16
5.1	Modeller	17
5.1.1	Bokning	17
5.1.2	Föreställningar	20
5.1.3	Administrativa och övriga modeller	24
5.2	Kontroller	26
5.2.1	Kund	27
5.2.2	Inloggning	28
5.2.3	Admin	29
5.2.4	Ekonomichef	36
5.2.5	Säljare	36
5.3	Vyer	37
5.3.1	Layouter	37
5.3.2	Inloggning	38
5.3.3	Admin	39
5.3.4	Säljare	56
5.3.5	Kund	60
6	Övriga komponenter	64
7	Genomförbarhet- och resursuppskattning	68
7.1	Sammanfattning av slutsatser av en genomförbarhetsundersökning av arkitekturmodellen.	68
7.2	Komponentprioritering	68
7.2.1	Modeller	68
7.2.2	Kontroller	69

7.2.3	Vyer	70
7.2.4	Övriga komponenter	72
7.3	Framtida projektuppgifter	72
7.4	Beroenden mellan uppgifter (?? ska tasks översättas till uppgifter??)	72
7.5	Tidsuppskattning för uppgifter	72
7.6	Uppgiftsplanering	72
7.7	Möjliga risker	73
8	Spårningsmatris mellan mjukvarukrav och strukturella krav	74

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

VAR SNÄLL OCH TA BORT STRÄCKET FRAMFÖR AKRONYMER OM DU ANVÄNDER DIG AV DEM.

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

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

Kontroll (Railscontroller) *todo.*

Partial (Railspartial) *todo.*

Action (Rails action) *todo.*

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

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

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

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

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

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

- **Bit** (Binary Digit) *Den grundläggande enhet som datorer arbetar med. En bit kan anta ett utav två möjliga värden (ofta angivna som 0 eller 1).*
- **Byte** *En vanlig enhet för informationsmängd i datasammanhang. En byte är ett paket bestående av åtta bitar.*
- **CentOS** *Ett operativsystem baserat på Red Hat Enterprise Linux som är gratis att använda. [1.3.2]*

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

- **Firefox** (Mozilla Firefox) *En webbläsare som går att köra på de populäraste operativsystemen.*
- **Foreign key** *Inom relationsdatabaser en begränsning som kräver att information på ett ställe finns definierat tidigare på ett annat ställe (kolumner i tabeller).*

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

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

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

GUI (Graphical User Interface) *Se gränssnitt.*

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

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

- **HTTPS** (HTTP Secure) *En kombination av HTTP och SSL/TLS med syftet att förhindra avlyssning av HTTP-trafiken*
- **Hårdvara** *Även kallat Maskinvara. Ett samlingsnamn för en dators fysiska komponenter.*
- **Internet Explorer** *En webbläsare utvecklad av Microsoft för operativsystemet Windows.*
- **InvalidAuthenticityToken** *Ett fel som kan uppstå i rails om användaren backar på sidor innehållande formulär. Uppstår på grund av Rails skydd mot XSS.*

- **KiB** (kibibyte) 2^{10} byte.

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

- **MiB** (mebibyte) 2^{20} byte.

- **Mib** (mebibit) 2^{20} bit.

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

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

Model-View-Controller *Ett koncept som bygger på att separera data (modeller), logik (kon-*

troller) och användarinterface (vyer).

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

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

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

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

Ruby *Ett objektorienterat programmeringsspråk.*

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

- **Sjöslaget** *Årligen återkommande studentfest på Finlandsfärja.*

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

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

- **SHA-1** *Ett sätt att spara strängar så de inte står i klartext. Det går inte att återskapa strängen efter omkodning.*

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

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

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

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

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

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

Webbserver *Program som körs på en server och distribuerar webbsidor och/eller andra filer som en webbläsare begär via HTTP-protokollet.*

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

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

1.4 Källor

Referenser till de källor som använts i dokumentet är listade här under. En och samma källa kan refereras vid flera ställen i texten. En referens är på formatet [Sektion.Rubrik.Löpnummer]. Exempelvis är [4.2.1] den första (1) referensen för rubriken “Uppdelningsbeskrivning” (2) under sektion “Systemdesign” (4).

Apache HTTP Server

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

Hänvisning till källan görs från referenserna: [1.3.1].

CentOS

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

Hänvisning till källan görs från referenserna: [1.3.2].

Cross-site-scripting

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

Hänvisning till källan görs från referenserna: [1.3.3].

Ruby on Rails Guides: Getting Started with Rails

http://guides.rubyonrails.org/getting_started.html

Hänvisning till källan görs från referenserna: [4.2.1].

1.5 Dokumentöversikt

1.5. Overview of the document. Similar to SRD Section 1.5, but describes the ADD.

Again it need not be assumed that readership on the customer side exists. In practise, this document may again be company confidential to the development team.

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

2 System Overview. Summarises: (i) the system context (how it fits into an existing framework of other packages and systems), and (ii) the system design. More detailed descriptions of (i) and (ii) are given in Sections 3 and 4 below.

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 systemet inte interagera med det befintliga systemet, annat än med länkar och därmed behöver vi bara anpassa vårt system visuellt.

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 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 dessa tillsammans. Anledningen till att Apache valdes var för att det är välanvänt, väldokumenterat samt att det är öppen källkod vilket underlättar det för oss.

3 Systemkontext

3 System Context. Gives a detailed description of the system context, with relevant diagrams. Defines the external interfaces of the product under development to these other systems.

3.n External interface definition. Provides an interface definition to each separate external component type or physical component.

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 MySQL

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

3.2 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. System Design. Provides an overview of the design techniques used, especially any in-house or non-standard methods, project specific methods, or non-standard interpretation of standard languages/methods such as UML.

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 identifieras av 5.Tn där T är någon av bokstäverna MVCLPX, som står för vilken typ komponenten är, och n är ett sekventiellt 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)

Beskriver en modell i MVC-modellen.

Specifiseringar och tillägg till komponentrubrikerna från 4.2.1

Syfte	Beskriver hur ett objekt av typen fungerar samt vilka begränsningar som ställs.
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 giltiga (valideringskrav), inklusive relationer.
Gränssnitt	Av Nyx definierade publika metoder som går att anropa och som antingen returnerar information (utöver standard-getters/setters) eller påverkar objektet, ej standardmetoder från ActiveRecord::Base. Inkluderar metodnamn, eventuella parametrar respektive returdata samt kortare beskrivning av vad metoden gör.
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 beskrivning av modellens samtliga attribut. Attributnamn följer lowercase_with_underscore-namngivning.

C - Controller (Kontroll)

Beskriver en kontroller i MVC-modellen.

Specificeringar och tillägg till komponentrubrikerna från 4.2.1

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)

Beskriver en vy i MVC-modellen.

L - Layout

Beskriver en layout i MVC-modellen.

P - Partial (Partiell vy)

Beskriver en partiell vy i MVC-modellen.

X - Övriga/externa komponenter

Beskriver övriga och externa komponenter.

Komponenter som beskrivs här kan vara logiska komponenter (exempelvis inloggningssystemet) eller externa komponenter (exempelvis Ruby-gem).

4.2 Uppdelningsbeskrivning

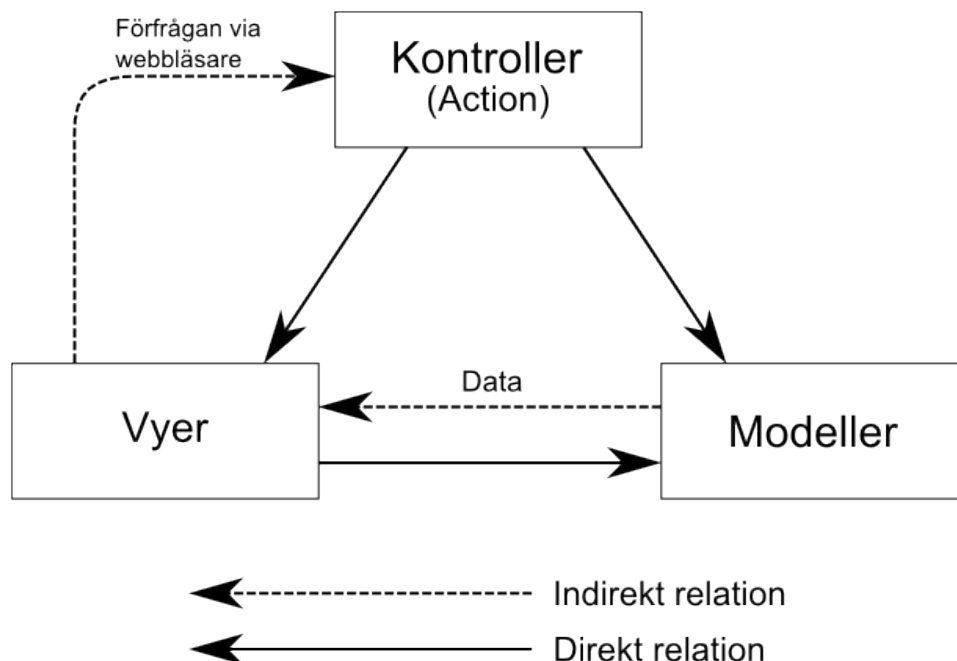
4.2 Decomposition description. Gives the top level view of the systems design, preferably with diagrams. Shows the major components which will be described in detail in Section 5. Identifies control and data flow between components.

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

» Modeller

» Vyer

» Kontroller



misplaced image? how do I make this good? – Peter

4.2.1 Modeller

Modeller representerar den data som applikationen använder sig av och består av och innehåller regler för hur denna data får manipuleras. I Nyx fall används modellerna främst för att bestämma hur tabeller i databasen ska interageras med. Det är här större delen av logiken i applikationen ligger.

Bokningssystemet har följande typer av modeller:

- » Administrativa modeller
- » Bokningar
- » Föreställningar

4.2.2 Vyer

Vyer representerar användargränssnittet till applikationen. Detta är olika webbsidor som visar representationer av data som finns i applikationen. De hanterar även förfrågningar som görs till systemet.

Bokningssystemet har följande typer av vyer:

- » 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 användares webbläsare. De gör förfrågningar vidare till modellerna för data och skickar vidare till vyerna för att kunna presenteras för användaren.

Bokningssystemet har följande typer av kontroller:

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

5 Komponentbeskrivning

5. Component Description.

Gives detailed component information according to a fixed template. Components may be top level components, identified in Section 4.2, or subcomponents of these. Preferably use a component identification scheme which is easy to read/follow/remember.

5.n. [Component identifier] Fill in name here.

5.n.1. Type. Could be a module, an input/output/temporary file, a program, a class, a script, a web page, etc.

5.n.2. Purpose. Describe the purpose of the component, and relate this to a numbered software requirement in the SRD.

5.n.3. Function. Describe the functionality of the component, including its interface properties (call and return types) and logical behaviour.

5.n.4. Subordinates. List the immediate subcomponents of the component, using defined component identifiers.

5.n.5. Dependencies. Describe the logical preconditions for using this component, e.g. files and/or objects that must exist.

5.n.6. Interfaces. Define the control and data flow to and from the object. Gives a detailed picture of its context in the overall system architecture.

5.n.7. Resources. List any resources required by the component, such as external components external subsystems, hardware, etc.

5.n.8. References. Reference any external documents needed to understand the component.

5.n.9. Processing. Describe the control and data flow between immediate subcomponents of this component. If the component has no immediate subcomponents (i.e. it is fully decomposed) then outline the method of processing used by the component to perform its task (e.g. with pseudo-code, state diagrams, etc).

5.n.10. Data. Describe in detail (where possible) the local data values and data structures belonging to (local in scope) this component. Otherwise give an outline description.

Komponenterna är numrerade med en bokstav och en siffra: 5.Xn

Där X är en bokstav som representerar vilken typ av komponent det är och n är ett löpnummer som är unikt för den typen av komponenter. Bokstäverna som används är:

- » M - Model (Modell)
- » C - Controller (Kontroll)
- » V - View (Vy)
- » L - Layout
- » P - Partial (Partiell vy)
- » X - Övriga/externa komponenter

5.1 Modeller

TODO skriv om följande stycke så att varje mening inte börjar med 'modellerna'.

Se även designmetod, det mesta står även där, kanske ska detta stycke tas bort.

Modellerna är datastrukturer som i de flesta fall är kopplade till databasen. Isåfall ärver modellen @ActiveRecord::Base@, se ActiveRecord (5.X4). Modellerna implementerar också grundläggande logik relaterad till datan.

Modellernas funktion är alltså att spara och hämta respektive data från databasen, samt hålla datan temporärt. Funktion har utelämnats i de flesta modeller.

Modellerna är kopplade till relaterade modeller genom ActiveRecord, för varje modell beskrivs relationerna under delkomponenter.

Modellernas gränssnitt och process beskrivs i ActiveRecord.

Modellernas data inkluderar, förutom vad som specificeras för respektive modell, även det som specificeras i ActiveRecord.

Alla modeller med databaskoppling har ActiveRecord som implicit resurs.

SR1.29 (användande av databas) implementeras av alla modeller tillsammans.

Skapande av databasschema och ingår i modellernas implementation.

5.1.1 Bokning

5.M1 Reservation

Typ	Modell
Syfte	Hålla information om en bokning (SR1.2, SR1.3).
Funktion	

|Delkomponenter|Tillhör en Show.

Har flera Placements.

Har flera ReservationCounters.

Gränssnitt

|Beroenden|Måste referera till existerande Show.

Måste ha minst en ReservationCounter.

Om inte @creator@ existerar måste @customer_reservation@ 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	
Beroenden	Måste ha existerande Reservation, Pricing och Section, samt Antal > 0.
Resurser	
Källor	
Process	
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	
Beroenden	Måste ha existerande Reservation, Show och Seat
Resurser	
Källor	
Process	
Data	

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	
Beroenden	Måste ha internalname, publicname och picture.
Resurser	
Källor	
Process	

|Data|@internal_name : string@

@public_name : string@

@description : string@

@comment : string@

@picture : string@ - bild eller referens till bild (?) för att visa sektioner och sittplatser.|

5.M5 Section

Typ	Modell
Syfte	Modell för sektion (SR1.9, SR1.22–SR1.24).
Funktion	

|Delkomponenter|Tillhör en Theater.

Har flera Seats.|

Gränssnitt	
Beroenden	Måste tillhöra en existerande Theater
Resurser	
Källor	
Process	
Data	

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	
Beroenden	Måste referera existerande Section
Resurser	
Källor	
Process	

|Data|@number : integer@ - stolsnummer

@row : integer@ (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	
------------	--

|Beroenden|Måste referera existerande Theater

description får inte vara tom.

Synlighet och startdatum måste finnas.|

Resurser
Källor
Process

|Data|@description : text@

@visible : boolean@ - Synlighet, huruvida kunden kan se och boka biljetter till omgången.

@start_date@|

5.M8 Show

Typ	Modell
Syfte	Data för en enskild föreställning (SR1.10).
Funktion	

|Delkomponenter|Tillhör en Batch.

Tillhör en Theater, genom Batch.|

Gränssnitt	
Beroenden	Måste referera existerande Batch.
Resurser	
Källor	
Process	
Data	@datetime : datetime@

5.M9 Pricing

Typ	Modell
Syfte	I varje Batch har varje Section i teatern normalpris och studentpris. Detta sköts av Pricing-modellen (SR?).
Funktion	

|Delkomponenter|Tillhör en Batch.

Tillhör en Section.|

Gränssnitt

|Beroenden|Referenser till Batch och Section måste vara giltiga.

@normal_price@ och @student_price@ måste vara icke-negativa heltal.|

Resurser

Källor

Process

|Data|@normal_price@

@student_price@|

5.1.3 Administrativa och övriga modeller

5.M10 User

Typ	Modell
Syfte	För att hålla koll på de olika roller användare kan logga in som (SR1.7, SR9.1, SR9.2).
Funktion	
Delkomponenter	Tillhör en Ability.
Gränssnitt	

|Beroenden|@username@ måste vara unikt.

@password@ måste finnas.|

Resurser
Källor
Process

|Data|@username : string@

@password : string@

@valid_to : date@|

5.M11 MailTemplate

Typ	Modell
Syfte	För att hantera och mailmallar för färdigformatteerade mail (SR1.27).
Funktion	
Delkomponenter	
Gränssnitt	
Beroenden	@content@ måste finnas.
Resurser	ActiveRecord::Base
Källor	
Process	

|Data|@description :string@

@title :string@

@content :string@|

5.M12 UserSession

Typ	Modell
Syfte	(Se 5.K2 AuthLogic, samt dess dokumentation) (SR1.1)
Funktion	Håller koll på de sessioner som finns inloggade
Delkomponenter	Har en User

|Gränssnitt|@find@ Letar upp efterfrågad session om den finns.

@user@ |

Beroenden	
Resurser	AuthLogic::Base::Session
Källor	AuthLogic: https://github.com/binarylogic/authlogic
Process	
Data	

5.M13 Ability

Typ	Modell
Syfte	(Se 5.K3 CanCan, samt dokumentationen) (SR1.1, SR9.2)
Funktion	Håller koll på vilka rättigheter varje användare har.
Delkomponenter	Har flera Users (<i>osäker på denna</i>)
Gränssnitt	@can@ Returnerar om användaren har rättighet att göra den efterfrågade handlingen.
Beroenden	
Resurser	ActiveRecord::Base, CanCan, User
Källor	CanCan: https://github.com/ryanb/cancan
Process	
Data	

5.2 Kontroller

5.C1 ApplicationController

Typ	Kontroller
Syfte	Rails-applikationens baskontroller som bland annat ska hantera autentisering (SR1.1).
Funktion	Alla kontrollers 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 kontrollers. De hjälpmetoder som definieras är:

@current_user@ - Returnerar en User-instans för den aktiva användaren (försöker logga in via Authlogic om nödvändigt).

@current_user_session@ - Returnerar en UserSession-instans för den aktiva inloggningen (försöker logga in via Authlogic om nödvändigt).

@require_user@ - Hjälpmetod som kastar en exception om den anropas och användaren inte är autentiserad.

Resurser	AuthLogic och CanCan.
Källor	Inga
Process	Filter definierade med @before_filter@ resp. @after_filter@ 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|@index@ - översikt

@statistics@ - statistik|

Resurser	ApplicationController, Reservation, Batch och Show.
Källor	Inga
Process	Inga

|Data|@@unplaced@ - antal betalade bokningar att placera

@@shows@ - tabell med information om aktuella föreställningar (antal bokade platser mm)

@@stats@ - tabell över föreställningar, sektionsgrupper(prisklass), med antal ordinarie, studenter, gratis, platser kvar|

5.C5 ReservationsController

Typ	Kontroller
Syfte	Administrering av bokningar (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 @show@.

@index@ - lista bokningar med ev. filter

@new@, @edit@ - 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.

@@filters@ - array med möjliga filter samt indikationer för vilka som är aktiva (@index@)|

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|@new@, @edit@ - presenterar aktuella placeringar och status för alla stolar med möjlighet att placera

@test_multiple@ [AJAX] - validerar placeringar, skapar temporära låsningar och returnerar dess IDn

@update_multiple@ - skapar och uppdaterar nya/existerande placeringar

@destroy@ - tar bort existerande placeringar för given bokning|

Resurser	ApplicationController och Placement.
Källor	Inga

|process|new > [test_multiple]+ > update_multiple

edit > [test_multiple]+ > update_multiple

edit > destroy|

|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.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|@index@ - presenterar gränssnittet för att lägga till och redigera teaterns stolar

@create@ [AJAX] - sparar en ny stol

@update@ [AJAX] - uppdaterar en existerande stol

@destroy@ [AJAX] - tar bort en existerande stol|

Resurser	ApplicationController, Theater och Seat.
Källor	Inga
Process	index > [create/update]+

|Data|@@seats@ - existerande stolar

@@theater@ - 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 @show@.
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 @show@.

@index@ - listar alla föreställningar för given omgång|

Resurser	ApplicationController, InheritedResources och Show.
Källor	Inga
Process	Standardprocesser för resurser.

|Data|Se InheritedResources.

@@batch@ - aktuell omgång

@@batches@ - existerande omgångar|

5.C12 PricingsController

Typ	Kontroller
Syfte	Hantering av prisklasser (SR1.30).
Funktion	Administrerang av priser på omgångsnivå. Varje sektion har ett pris för varje rabattklass. Alla pris-/sektionskombinationer för given omgång administreras på en gång, oavsett om de redan finns eller inte.
Delkomponenter	Inga
Beroenden	Autentiserad som administratör, nästlad via BatchesController.

|Gränssnitt|@index@ - visar formulär för att välja priser för alla sektioner för den aktuella omgången

@update_multiple@ - lagrar informationen med datan från @index@|

Resurser	ApplicationController, Batch, Section och Pricing.
Källor	Inga
Process	index > update_multiple > index

|Data|@@batch@ - aktuell omgång

@@sections@ - sektioner i omgångens teater

@@pricings@ - 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 lösenordsändringar 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 genomköra 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 resp. 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 PlacementsController

@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 oplacerad 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	Inga
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	Inga
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	Inga
Data	

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	Inga
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	Inga

|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	Inga
Data	@stats - tabell över föreställningar, sektionsgrupper(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, platser kvar

5.V6 ReservationsController#index.html

Typ	Vy
Syfte	Administratören ska kunna hantera bokningar (SR1.13)
Funktion	Administratören får en överblick över de val han kan göra angående bokningar.
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	AD5.P1
Beroenden	ReservationsController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@reservation

5.V8 ReservationsController#show.html

Typ	Vy
Syfte	Administratören ska kunna hantera bokningar (SR1.13)
Funktion	Administratören ska kunna visa att en bokning ligger i databasen genom att söka efter den med sökfunktionen eller filter.
Delkomponenter	Inga
Beroenden	ReservationsController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@reservation

5.V9 ReservationsController#edit.html

Typ	Vy
Syfte	Administratören ska kunna hantera bokningar (SR1.13)
Funktion	Administratören ska kunna göra ändringar i en bokning, till exempel ändra antalet biljetter, ändra betalstatus, placeringsstatus, och så vidare.
Delkomponenter	AD5.P1
Beroenden	ReservationsController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@reservation

5.P1 ReservationsController#_form.html

Typ	Partiell vy
Syfte	Administratören ska kunna hantera bokningar (SR1.13)
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

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 aktulla 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 attributer

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	AD5.P3
Beroenden	TheatersController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@theater

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

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	AD5.P3
Beroenden	TheatersController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@theater

5.P2 TheatersController#_form

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

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	AD5.P4
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 sektionssplaceringen (SR1.22, SR1.23 (i plus))
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 sektionssplaceringen (SR1.22)
Funktion	Administratören ska kunna ändra vilka sektioner som är valbara.
Delkomponenter	AD5.P4
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

Typ	Partiell vy
Syfte	Administratören och säljaren ska kunna välja sektionspaceringen (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

5.V22 BatchesController#new.html

Typ	Vy
Syfte	Administratören skall kunna lägga till omgångar (SR1.10)
Funktion	Skapa en ny omgång.
Delkomponenter	AD5.3.36
Beroenden	BatchesController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@batch

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

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	AD5.3.36
Beroenden	BatchesController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@batch

5.P4 BatchesController#_form

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

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	AD5.3.41
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	AD5.3.41
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

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	Administratören och ekonomichefen ska kunna registrera betalningar (SR1.12)
Funktion	Administratören och ekonomichefen behöver kunna registrera om en betalning är gjord, så att administratören eller eventuellt säljaren kan placera ut bokningens platser.
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 användarnas konton (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 användarnas konton (SR1.7, SR1.8)
Funktion	Administratören ska kunna skapa en ny användare och sätta denna rättigheter i systemet.
Delkomponenter	AD5.P7
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 användarnas konton (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 användarnas konton (SR1.7, SR1.8)
Funktion	Administratören ska kunna ändra en användares rättigheter eller lösenord, till exempel säljare som inte ska säljare längre.
Delkomponenter	AD5.P7
Beroenden	UserController
Gränssnitt	Ej applicerbart
Resurser	5.L1 admin_layout.html
Källor	Inga
Process	Ej applicerbart
Data	@user

5.P6 UsersController#_form

Typ	Partiell vy
Syfte	Administratören ska kunna hantera systemets användarnas konton (SR1.7, SR1.8)
Funktion	Renderar ett formulär för att redigera User-objekt.
Delkomponenter	Inga
Beroenden	UsersController
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 överiskt 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	AD5.3.56
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 ett 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	Ändra mottagare och innehåll i en befintlig mailmall.
Delkomponenter	AD5.3.56
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

Typ	Partiell vy
Syfte	Administratören ska kunna hantera mailutskick (SR1.14)
Funktion	Renderar ett formulär för att redigera Mailng-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.
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 för en vald omgång som är aktiv.
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 i plus, SR1.25)
Funktion	Välja vilken plats som ärendet gäller.
Delkomponenter	Inga
Beroenden	Om administratören inte har placerat ut den redan boka- de platsen så gör säljaren det, men sektionssvalet 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 i den nuvarande omgång som denne vill gå på.
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 i plus)
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 terminerar bokningen från databasen.
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|

6 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| Alla kontroller som ärver från denna klass får standardactions för resurser (`@index@`, `@show@`, `@new@`, `@create@`, `@edit@`, `@update@` samt `@destroy@`) fördefinierade om inget annat anges.

`@actions@` - Hjälpmetod för att specificera vilka actions som ska fördefinieras.

Hjälpmetoder som görs tillgängliga i vyer:

`@resource@` - Enskild instans av den representerade resursen (i `@show@`, `@new@`, `@create@`, `@edit@`, `@update@`, `@destroy@`).

`@collection@` - En array innehållandes alla objekt av resursen (endast i `@index@`).

`@resource_class@` - En referens till den modell (klassdefinitionen) som hanteras som resurs.

Resurser	InheritedResources::Base (ingår i Rails-gemet inherited_resources).
Källor	inherited_resources: https://github.com/josevalim/inherited_resources
Process	Inga
Data	Instansvariabler av formatet <code>@modellnamn</code> (ex. <code>@reservation</code> för <code>ReservationsController</code>) respektive <code>@modellnamn_i_plural</code> för index-actions (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	Inga
Data	Inga

5.X3 CanCan

Typ	Övrig komponent (Rails-gemet)
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|@can?@` - returnerar huruvida användaren har rättigheter för att utföra en viss handling (tillgänglig i vyer och kontroller)

`@cannot?@` - inversen av `@can?@`

`@ApplicationController#authorize!@` - kastar en exception om användaren inte har rättighet att utföra den handling som angivits

`@ApplicationController#load_and_authorize_resource@` - ladda in och autorisera alla standardactions för resurser automatiskt|

Resurser	CanCan (ingår i Rails-gemet cancan) och Ability.
Källor	cancan: https://github.com/ryanb/cancan

`|Process|`1. Rättigheter fördefinieras i Ability-modellen.

2. `@can?(:action, Object)@` anropas (:action = någon fördefinierad action, Object är någon modell alternativt en modellinstans).

3. CanCan hämtar aktiv användare via `@ApplicationController#current_user@`.

4. Användarens rättigheter undersöks av CanCan genom att skicka denna till Ability-modellen.

5. `@can?@`-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.
Funktion	Tillhandahåller gränssnitt för sökningar och manipulation av objekt i databasen. Implicit ingår databas och databas-schema i varje modell.
Delkomponenter	Inga

|Gränssnitt|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.

Följande är några av de metoder som definieras av @ActiveRecord::Base@ och är tillgängliga för alla modeller.

@all@ - hämta alla instanser av modellen (alla rader i databastabellen).

@find@ - gör diverse sökningar i databasen och returnerar de instanser som matchar.

@valid?@ - kontrollera att objektets data är korrekt. Vilka kontroller som görs definieras av varje modell.

@save@ - lagrar/uppdaterar en instans av ett modell.

@update@ - uppdaterar attributer för en instans i databastabellen.

@destroy@ - raderar objektet i databasen.

Dessutom finns accessors för varje attribut hos modellen. Dessa definieras dynamiskt.

Beroenden	Inga
Resurser	Inga

|Källor|ActiveRecord: <http://ar.rubyonrails.org>

|Process|Följande typscenariot visar skapande och uppdatering av en instans av modellen @M@.

```
<pre>m = M.new
```

```
m.att1 = "value"
```

```
if m.valid?
```

```
then m.save</pre>
```

|Data|Alla modeller har datafälten @created_at@ respektive @updated_at@ av typen @ti-

mestamp@. Dessa uppdateras automatiskt av Rails vid skapande och uppdatering av objektet.

Alla modeller har även attributet @id@ som primärnyckel.

För övriga attribut se respektive modell. Enkla attribut listas, relationer till andra modeller beskrivs i delkomponenter.

7 Genomförbarhet- och resursuppskattning

6. Feasibility and Resource Estimates. Summarise the conclusions of a feasibility study around the architectural model. Prioritise all components with a priority model (e.g. economy, standard, deluxe versions).

Identify and describe all future project tasks. Identify task dependencies in terms of commencement and completion, preferably with a task flow chart. Estimate the effort required for each project task. Produce a task allocation plan and schedule for each project staff member for the remainder of the project. This information should preferably be presented in a table.

Identify possible risks going forward, and for each risk, give a risk management proposal. Estimate the overall schedule for making a detailed design, coding this design, testing and delivering the final product and documentation according to the project deadlines. Identify the critical path in the project, and analyze possible project slippage along this path.

7.1 Sammanfattning av slutsatser av en genomförbarhetsundersökning av arkitekturmodellen.

Arkitekturen Nyx kommer använda kallas Model-View-Controller, MVC. Alla komponenter under respektive punkt finns angivna nedan.

I designen ser vi endast en del som kommer att vara mer komplicerad, nämligen stolsplaceringen. Det är dock fullt genomförbart, eftersom det inte är avancerade tekniker som används, utan att det är mycket data som ska sammankopplas.

Modelleringen är redan gjord och simuleringen borde inte vara några problem, vi kommer nämligen använda rätt arkitektur från början.

Slutsatsen är således att projektet är fullt genomförbart med den här arkitekturmodellen.

7.2 Komponentprioritering

Alla felaktiga numreringar måste fixas till i punkt 5 innan det kan fixas här. Prioritet är inte säkert, saknas många referenser till SRD under syfte i punkt 5. Fixas på måndag!!!

7.2.1 Modeller

Komponent	Prioritet
5.2.1 Reservation	Standard
5.2.2 Reservation-Counter	Standard
5.2.3 Placement	Standard
5.2.4 PlacementLock	Standard
5.5.5 Theater	Standard
5.n Section	Standard
5.n Seat	Standard
5.n.8 Batch	Standard
5.n.9 Show	Standard
5.n.10 Pricing	Standard
5.2.11 User	Standard
5.2.12 MailTemplate	Standard
5.2.13 Mail	Standard
5.2.14 UserSession	Standard
5.2.15 Ability	Standard

7.2.2 Kontroller

Komponent	Prioritet
5.C1 Application-Controller	Standard
5.n BookingController	Standard
5.n SessionController	Standard
5.n AdminIndexController	Standard
5.n Reservations-Controller	Standard
5.n PlacementsController	Standard
5.n TheatersController	Standard
5.n SectionsController	Standard
5.n SeatsController	Standard
5.n BatchesController	Standard
5.n ShowsController	Standard
5.n PricingsController	Standard
5.n UsersController	Standard
5.n MailingController	Standard
5.n SalesController	Standard

7.2.3 Vyer

Komponent	Prioritet	
5.L1 AdminLayout	Standard	
5.L2 SalesLayout	Standard	
5.L3 SimpleLayout	Standard	
5.L4 CustomerLayout	Standard	
5.V1 Session#new.html	Standard	
5.V2 AdminIndexController#index.html	Standard	
5.V3 AdminIndexController#stat_standard.html	Standard	
5.V4 AdminIndexController#stat_plus.html	Plus	
5.V5 AdminIndexController#stat_deluxe.html	Deluxe	
5.V6 ReservationController#index.html	Standard	
5.V7 ReservationController#new.html	Standard	
5.V8 ReservationController#show.html	Standard	
5.V9 ReservationController#edit.html	Standard	
5.P1 ReservationController#_form	Standard	
5.V10 PlacementController#index.html	Standard	
5.V11 PlacementController#new.html	Standard	
5.V12 PlacementController#show.html	Standard	
5.V13 PlacementController#edit.html	Standard	
5.P2 PlacementController#_form	Standard	
5.V14 TheatersController#index.html	Standard	
5.V15 TheatersController#new.html	Standard	Nyx
5.V16 TheatersController#show.html	Standard	

7.2.4 Övriga komponenter

Komponent	Prioritet
5.K1 InheritedRe-sources	Standard
5.K2 AuthLogic	Standard
5.K3 CanCan	Standard

7.3 Framtida projektuppgifter

- » Utveckla hemsidan, det vill säga frontend .(?? Borde översättas??)
- » Utveckla det bakomliggande systemet, backend.
- » Verifiera funktioner genom testning. Dessa tester behöver utvecklas.
- » Förbereda ett demo, produkten ska demonstreras och detta kräver förberedelser.
- » Bestämma designen på hemsidan, den kommer användas av många och bör vara tilltalande samt passa in i Kårspexets egen design.

7.4 Beroenden mellan uppgifter (?? ska tasks översättas till 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.

7.5 Tidsuppskattning för uppgifter

7.6 Uppgiftsplanering

Namn	Vecka 1	Vecka 2	Vecka 3	Vecka 4	Vecka 5	Vecka 6
Projektmedlem 1 <i>mögel</i>	nått	nått				
Projektmedlem 2 <i>tarani</i>	nått	nått				
Projektmedlem 3 <i>perka</i>	nått	nått				
Projektmedlem 4 <i>kalle</i>	nått	nått				
Projektmedlem 5 <i>erik</i>	nått	nått				
Projektmedlem 6 <i>daniel</i>	nått	nått				
Projektmedlem 7 <i>andré</i>	nått	nått				
Projektmedlem 8 <i>lemming</i>	nått	nått				
Projektmedlem 9 <i>johan</i>	nått	nått				
Projektmedlem 10 <i>rasmus</i>	nått	nått				
Projektmedlem 11 <i>mia</i>	nått	nått				

7.7 Möjliga risker

Projektplaneringen visar att tiden kommer att räcka för att slutföra projektet och hittills har samarbetet fungerat väl. Vi 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.

Risk att vi implementerar fel funktionalitet (annat än det efterfrågade) och därmed förlorar tid.

Risk att vi inte hinner implementera alla funktioner som efterfrågas

Risk att vi implementerar saker i fel ordning så att vi förlorar tid på att vänta på varandra. Här kommer nyckelkompetens in i beräkningarna.

Risk att vår utvecklingsserver krånglar.

Risk att det krånglar när vi ska installera vårt system på kårspexets server.

8 Spårningsmatris mellan mjukvarukrav och strukturella krav

7. Software Requirements vs Components Traceability Matrix. Gives a table cross referencing architectural components (based on defined component identifiers) to software requirements numbered in the SRD.