Bokningssystem för Kårspexet

Project Planning Document – PPD

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

Nyx

Version 1.0
November 15, 2010

Abstract

The purpose of this document is to plan out our project; to construct a booking system for the "KTH Kårspex". To start off we go into the detailed specifications from our customer by explaining different user scenarios from the view of their end user, salespersons, administrators, and the financial manager. We reviewed our current skills to see if we needed to obtain knowledge in any area in which we are currently lacking.

To be able to implement a full-functioning and well-performing system for handling ticket reservations, we looked into existing systems to find good ideas that we might be able to use in our implementation of a similar system. For one, we are looking into if there is a commercial ticket reservation system that is suitable for our customer according to their specification. From our research we concluded that SF Bio implements a visually pleasing, and user-friendly ticket-reservation system that we would like to use as a model. There is also a technical aspect to this; we know from our customer's specifications that they want a web-based system. Therefore we looked into a suitable system to run our application. Having decided to go with open-source software, we speculate that a Linux-based system running an apache web server is the best choice.

Conclusively we believe that this project is feasible to follow through with because we have the required knowledge to implement a ticket-reservation system to the customer's specifications.

Innehåll

1 Problemformulering

Kårspexet är en kårförening vid Tekniska Högskolans Studentkår som arrangerar teaterföreställningar. De är i behov av ett nytt bokningssystem. För närvarande använder Kårspexet ett bokningssystem som lånats av Fysikalen, Fysiksektionens motsvarighet till Kårspexet, som de inte är nöjda med. Kårspexet har inte tillgång till bokningssystemets källkod och dess dokumentation är bristfällig. De vill ha ett nytt biljettsystem med bra dokumentation och ett användarvänligt gränssnitt. De vill dessutom ha tillgång till källkoden till systemet så att de har möjlighet till vidareutveckling. Därför har de vänt sig till MVK-kursen och indirekt Nyx för att skapa ett nytt och bättre bokningssystem.

1.1 Detaljerad problemformulering

Kårspexet har definierat fyra typer av användare av bokningssystemet; kund, säljare, ekonomichef och administratör. De olika användarna ska ha olika gränssnitt men varje användare kräver inte personlig inloggning. Exempelvis får säljare tillgång till säljare-kontot och loggar in med det istället för personlig inloggning.

1.1.1 Typscenario

Det typiska scenariot över hur systemet ska användas ser ut som följande:

Kund

1. Tittar via webben efter möjliga föreställningar.

2. Väljer och bokar två biljetter till en föreställning.

3. Tar emot ett mail med betalningsinformation.

4. Betalar omgående.

Ekonomichef

5. Registrerar betalningen.

Administratör

6. Tilldelar stolsnummer till biljetterna.

Kund

7. Får ett mail om att biljetterna finns att hämta ut på kåren.

8. Hämtar ut biljetterna mot bokningsnummer.

Säljare

9. Registrerar bokningen som utlämnad.

Kund

10. Uppvisar biljett på teatern och släpps in.

1.1.2 Funktionalitet

Kunden ska kunna boka sittplats i en sektion, dock inte välja stolsnummer. Priset på biljetterna varierar beroende på sektion, omgång, och eventuella rabatter. Rabatt kan t.ex. gälla medlemmar i Tekniska Högskolans Studentkår. Administatören placerar sedan manuellt ut platserna eller utnyttjar ett automatiskt förslag. Placeringen går inte att ändra på efter att biljetten har hämtats ut av kunden. Vid direktförsäljning kan en säljare placera ut platser direkt.

En bokning kan antingen vara obetald, betald, eller gratis. biljetter kan vara placerade, oplacerade, eller uthämtade av kunden. Kårspexet vill ha möjlighet att ge ut fribiljetter till personer som har hjälpt till eller liknande.

Varje teater har en karta över ett antal platser. Varje omgång av föreställningar har egna priser och egna uppdelningar av platser in i sektioner. Varje föreställning har egna informationsmail och tidpunkter. En föreställning måste publiceras för att synas för allmänheten på hemsidan, men går att administrera innan publicering. Nya teatrar måste även kunna läggas till på ett smidigt sätt. En idé till en enterpriseutgåva vore att ha en biljettgenerator som skriver ut färdiga biljetter.

1.2 Användargränssnitt

1.2.1 Kund

Kunden bokar biljetter över nätet. Till detta gränssnitt har vi fått stor frihet men även ansvar vad gäller design.

Kunden ska kunna se aktuella föreställningar, boka biljett med föreställning, sektion, rabattklass, betalningssätt, leveranssätt och kontaktuppgifter. Dessutom så ska kunden kunna avboka via länk i bekräftelsemailet. Önskemål till eventuell enterpriseutgåva vore att kunden kan betala med kort direkt på sidan.

1.2.2 Säljare

Säljaren har hand om den manuella försäljningen. Det finns många som kommer använda gränssnittet och det måste därmed vara enkelt och felsäkert. Systemet ska även klara av att hantera flera säljare samtidigt.

Säljaren kan placera ut köpta biljetter och lämna ut dem, samt kunna ta emot kontant betalning av en webbokning. Säljaren har en begränsad förmåga att ändra köpta eller bokade biljetter. Säljaren kan endast avboka sina egna bokningar, sina egna misstag. Lediga platser ska uppdateras automatiskt, så att dubbelbokning blir omöjlig.

1.2.3 Ekonomichefen

Ekonomichefen registrerar betalningar, och har tillgång till statistik över sålda biljetter, föreställningar, rabatter och dylikt.

Ekonomichefen ska kunna visa betalda och obetalda bokningar, registrera betalningar genom att ändra bokningsstatus och registrera biljetter som gratisbiljetter. Önskemål är att kunna se omfattande statistik.

1.2.4 Administratör

Administatören har som huvuduppgift att ha inblick i systemet och kunna lösa de problem som uppstår. Löpande sköter administratören uppgifter som att utföra stolsplaceringar och lägga upp nya föreställningar, ändra bokningar med mera. Adminstratörskontot är ej personligt.

Administratören skall kunna ändra alla bokningar, placeringar, status med mera, hantera avbokningar samt kunna ändra informationen i alla utskick; bekräftelsemail, 'hämta biljett'-mail, information om inställd föreställning etc. Det skall vara enkelt att lägga till teatrar, nya föreställningar och omgångar. Det ska även vara enkelt att ändra information om dessa samt prissättning på olika omgångar.

Det finns även önskemål om att se statistik över biljettförsäljningen och att smidigt kunna ändra på platsskisser samt lägga in nya. Önskemål till en eventuell enterpriseutgåva är att kunna se statistik över tid.

1.3 Motivering av projektet

Efter att ha inventerat våra kunskaper inom projektgruppen valde vi Kårspexets projekt eftersom vi ansåg att det passade vår kompetens bra. Vi förde ingående diskussioner om hur mycket kunskap vi skulle behöva införskaffa för att klara av projektet. Detta vägdes emot hur mycket vi skulle behöva lära oss för att skriva ett ännu mer utmanande mjukvaruprojekt. Genom att tänka ur ett företagsperspektiv kom vi fram till att vi hellre skulle välja ett projekt som utnyttjade vår projektgrupps styrkor. Därför kändes det som ett säkrare val att arbeta med ett projekt i ett område som vi har mycket erfarenhet inom. Istället för att införskaffa kunskap kring hur vi t.ex. skriver en mobilapplikation kan vi då lägga vår tid på att dokumentera och utföra projektet väl. Slutligen fastande vi för Kårspexet eftersom vi tyckte att det var ett mycket väl specificerat projekt, vilket gjorde att vi snabbt kände att det var genomförbart och rimligt för vår grupp.

1.4 Mål

Vårat övergripande mål inom gruppen är att lära oss att arbeta väl i grupp och sträva mot samma mål, liknande hur man arbetar på ett vanligt företag. Då vi inte tidigare har arbetat i ett professionellt sammanhang hoppas vi att få ut mycket erfarenheter och kunskaper kring hur det är att arbeta formellt i en större grupp. Ett välanvänt och användarvänligt bokningssystem som används aktivt av Kårspexet tror vi även på som en merit i framtida arbetssökningar.

Dessutom kommer projektet ge flera av oss erfarenheter inom webbprogrammering och databasstrukturering vilket för oss är värdefullt inför framtiden.

1.5 Projektgruppens kompetenser

Inom gruppen har vi flertalet medlemmar med tidigare erfarenhet och kompetens inom webbprogrammering. Vi har även en medlem, Daniel, som är duktig på GUI-design och har läst kurser i Människa- och datorinteraktion. Tre av gruppmedlemmarna – Rasmus, Kalle och Anna Maria – är även duktiga på algoritmer, datastrukturer och logik och är intresserade av att jobba med den logiska delen av projektet.

Det vi kommer behöva lära oss är webbramverket Ruby on Rails samt hur vi ska sköta betalningar via plusgiro. Andreas och Victor kommer ha huvudansvaret för att lära ut Ruby on Rails eftersom de båda har mycket erfarenhet i att använda ramverket sedan tidigare. Hur vi ska lära oss betalningslösningar diskuteras fortfarande. Andreas har dock arbetat lite med kortbetalningar tidigare och har därmed vissa erfarenheter med sig.

2 Problembakgrund

2.1 Kommersiell problembakgrund

2.1.1 Existerande lösningar

Det finns många färdiga lösningar för biljettbokningssystem. Ett exempel är SoftAdmin som utvecklas av MultiSoft Consulting. Systemet kundanpassas av MultiSoft och används av bland annat av Svenska Filminstitutet och Stockholms Konserthus. En utstående funktionalitet hos konserthusets implementationen är att man vid val av plats får se en bild över alla sektioner och kan klicka på den sektion man vill sitta i. Negativt här är dock att man kan klicka även på sektioner där det inte finns några platser kvar.

Det finns även ett open source-projekt, osConcert, som har utvecklats av Cartzone UK. osConcert används av flera teatrar i London, bland annat London Theatre Royal Drury Lane. En funktionalitet som vi tyckte var bra med osConcert är att vid valet av föreställning så användes en kalender med alla föreställningar i den månaden utplacerade.

2.1.2 Liknande system

Förutom de färdiga lösningar vi har hittat finns det även ett antal implementerade biljettbokningssystem. Några exempel är de som används av SF, SJ och Södra Teatern. Vi tycker att SFs system är riktigt bra. Det har få steg för att boka en biljett, det är tydligt vad som ska göras i varje steg och det råder inga tveksamheter om vilken plats man har bokat. En nackdel med systemet är att det använder Flash för platsbokningen, vilket är något som inte fungerar för alla användare. Till skillnad från SF använder Södra Teatern en extern sida för att hantera sina bokningar. Deras system är också relativt bra, men valet av föreställning är oöverskådligt och systemet presenterar mycket onödig information.

Vi anser att SF har det bästa systemet, då det mest liknar det vi vill implementera. Systemet ger en bra överblick av bokningen som tydligt visar vilka steg som är gjorda samt vilka som är kvar. Alla val presenteras i form av en biljett till höger på skärmen så att användaren aldrig behöver vara osäker på vilka val som gjordes i tidigare steg. När det kommer till att välja en plats så möts användaren av en ritning av lokalen i fråga, där användaren får välja exakt vilka platser han eller hon vill boka. De platser som redan är bokade är markerade med rött, och kan inte bokas. Ytterligare en viktig funktion är att användaren kan välja skilda platser i ett sällskap och även dela upp dessa med olika betalmedel. Användaren kan alltid ångra sig i bokningen och backa tillbaka. Användaren kan till sist välja att antingen enbart boka eller betala sin bokning direkt på sidan.

2.1.3 Funktioner som vi kan implementera

Ytterligare funktionalitet som vi gillade var osConcerts kalender för att visa föreställningar. Då vår kund, Kårspexet, endast kommer att ha ett fåtal föreställningar per dag och inte kommer att byta lokal ofta, så ger en kalender bra översikt över Kårspexets föreställningar.

Ett kalenderbaserat bokningssystem förväntas implementeras till standardutgåvan. Att systemet likt SFs visar exakt var i bokningen användaren är förväntas även implementeras i standardutgåvan. En simpel platsbokning per sektion i lokalen hamnar även under standardutgåvan. Betalning direkt på sidan kan komma att implementeras i en eventuell enterpriseutgåva. Platsbokning i form av separata platser som SF använder sig av, kan komma att implementeras i en enterpriseutgåva som inte riktar sig till kunden.

2.2 Teknisk bakgrund

2.2.1 Typ av applikation

Baserat på den kompetens som gruppen innehar har vi valt att implementera alla gränssnitt i form av webbapplikationer. För detta kommer vi först och främst behöva en webbserver samt någon form av databas. Applikationen kommer innefatta en hel del logik som av både praktiska och säkerhetsrelaterade skäl bör exekveras på servern. Vi har valt att uteslutande utnyttja open source-projekt då dessa är fria att använda, väletablerade samt ofta även väldokumenterade.

2.2.2 Plattform

Av ovanstående skäl väljer vi Apache som webbserver och Linux som operativsystem att köra bokningsystemet på. Båda systemen är öppna och används i stor utsträckning. Ett alternativ till detta hade varit Windows med webbservern IIS, men då ingetdera av dessa är fria programvaror väljer vi bort dem.

2.2.3 Programmeringsspråk

Av de programmeringsspråk som är vanliga för webbutveckling har vi valt två kandidater; Ruby on Rails samt PHP. PHP är väletablerat sedan länge, men Rails anses vara agilare (kortare

utvecklingscykel).

I slutändan beror språkvalet (och även övriga komponenter i plattformen) på vilken servermiljö kunden har tillgång till. Det nuvarande bokningssystemet är utvecklat i PHP och det kan mycket väl vara så att ett system skrivet med Ruby on Rails inte är möjligt för dem att köra.

2.2.4 Databas

Bokningssystemet kommer att kräva en databas för att lagra all information. Vi har undersökt databasplattformarna MySQL och PostgreSQL. Båda går att köra under Linux med både PHP och Rails och är även ändamålsenliga för vår typ av applikation. MySQL är populärare, bättre dokumenterat och våra utvecklare är dessutom vana vid det.

3 Slutsats

Vi har gjort bedömningen att våra kompetenser är tillräckliga för att ta oss an projektet, då vi redan har erfarenhet av databaser och webbgränssnitt. Dessa nyckelkompetenser finns alla hos minst två gruppmedlemmar och därmed anser vi att risken är liten att ett avhopp av en enstaka gruppmedlem skulle hindra oss från att slutföra projektet. Daniel har, till skillnad från oss andra i gruppen, fördjupade kunskaper i användargränssnitt. Vi kan dock klara oss utan dessa om vi måste. Även om vi har viss erfarenhet, finns det förstås saker vi måste lära oss. En risk som pekats ut är att vi inte skulle hinna lära oss den plattform vi väljer att använda.

Vi tror att tiden kommer räcka, vi är elva personer och hittills har samarbetet fungerat väl. När det gäller implementationen bedömer vi att vi kommer att hinna med åtminstone en grundläggande implementation av bokningssystemet. Med den utförliga specifikation samt dokumentation som följer med kursen förväntas kodandet gå relativt snabbt gentemot samma projekt utan skrivna dokument.

En risk är att våra specifikationer inte uppfyller kundens krav och att programmet därför struktureras på ett sätt som gör kundens mål svåra att uppfylla. Men då vår kund har tydliga krav och vi har löpande kontakt med dem så bedömer vi risken att det uppstår missförstånd som relativt liten. Från vår sida beror inte kommunikationen med kunden på en enda person, vilket gör att om en av våra ansvariga hoppar av har vi fortfarande en annan person som har bra kontakt med kunden.

En annan risk är att vår kund, Kårspexet, drar sig ur eller ändrar sina krav under projektets gång. Vi bedömer risken att de skulle dra sig ur som mycket osannolik då de under en lång tid har varit mycket måna att få ett eget bokningssystem som fungerar. För att undvika att de ändrar sina krav markant så är löpande kontakt och ömsesidig förståelse av projektets omfattning väldigt viktig.

A Gruppmedlemmar

Andreas Tarandi 890416-0317 taran@kth.se

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 vigge19@gmail.com

Anna Maria Modée 871120-0363 ammodee@kth.se

Daniel Nyberg 900104-4495 dnyb@kth.se

Johan Stjernberg 890315-0533 stjer@kth.se

B Projektets fördelade roller

B.1 Projektledare

Andreas Torandi

Rollen innebär att vara väl insatt i projektet och dess mål, att fastställa beslut, att ha stort ansvar och att gruppmedlemmarnas samarbete leder till rätt mål.

B.2 Projektsekreterare

André Gräsman

Rollen innebär att dokumentera de beslut som fattas under möten, föra protokoll över gruppmedlemmarnas arbetsinsats, samt att dokumentera relevanta idéer inför framtiden.

B.3 Chefsprogrammerare

Victor Hallberg

Rollen omfattar ansvar för program implementationens alla delar, att delegera uppgifter till programmerarna.

B.4 Programmerare

Victor Hallberg, Erik Eklund, Kalle Arvidsson, Per Hagsten

Rollen innebär stort men delat ansvar för implementation av programvara, att tydliggöra tekniska aspekter i projektet.

B.5 Dokumentationshanterare

Peter Boström

Rollen innebär att samla gruppmedlemmarnas dokumentation på ett för alla åtkomligt ställe, samt att ansvara för dokumentationens stil och utformning vid rapportens sammansättning.

B.6 Rapportskrivare

Per Hagsten, André Gräsman, Erik Eklund

Rollen innebär att på uppdrag producera dokument som successivt färdigställer rapporten, samt att hjälpa i gruppen upptagna ansvariga med dokumentation.

B.7 Testare

Peter Boström, Anna Maria Modée, Daniel Nyberg, Johan Stjernberg, Kalle Arvidsson

Rollen innebär att testa implementationens delar efter felaktigheter i programvaran.

B.8 Behovsanalytiker

Rasmus Göransson

Rollen innebär att analysera och förstå kundens krav, att fastställa kundens vaga önskning till något precist, att forska i andra människors önskningar om hur systemet bör fungera, samt ett tätt samarbete med den kundansvariga.

B.9 Arkitekt/Designer

Erik Eklund

Rollen innebär att strukturera hur programvaran skall implementeras, samt att ansvara för problemanalys och lösningar till problem vid implementationen.

B.10 GUI-expert

Daniel Nyberg

Rollen innebär ett ansvar omfattande utformning av användargränssnittets användarvänlighet, samt att testa användarvänligheten på en testgrupp.

B.11 Projektplanerare

Johan Stjernberg

Rollen innebär ansvar för vad som skall göras när, att informera om kommande deadlines, samt att boka mötesrum för schemalagda projektmöten.

B.12 Kundansvarig:

Anna Maria Modée

Rollen innebär att vara kundens enda kontaktperson och gruppens länk till kunden.