1 Presentation

Johnny Pesola heter jag.

Har gjort ett bokningssystem för lokaler, resursbokningar och mat bokningar kommer att implementeras i framtiden.

2 Kunden

* Kunden är en församlingen Västanfors-Västervåla församling som ingår i Svenska Kyrkan. Församligen jobbar med dop, vigslar, begravningar. Driver också en konferrensverksamhet med mat.
* Körverksamhet,

3 Vision

* Församlingen arbet sedan långt tillbaka ett **flertal bokningspärmar**, vilket inte alltid fungerar så bra.
  + **Ibland ligger inte pärmen** **där den ska ligga.**
  + Från avstånd måste man ringa expeditionen för att boka.
* Visionen är kort och gott att lösa detta med ett webbaserat bokningssystem.
* Önskan om ett komplett och rätt komplicerat bokningssystem har funnits länge.

4 Krav och önskemål

* Ett **webbaserat** **system** för **boka lokaler** och **utrustning** och **personal** till dem och sedan **mat** till de som bokar lokalerna.

Eliminera risk för dubbelbokningar.

5 Val av ramverk / mjukvara

* **Angular.js** används som frontend. Ett javascript ramverk backat av **google**
  + Använder **databinding**, vilket gör det möjligt att på ett **enkelt visa** och **samla in** **data** från användaren.
* **Microsoft asp.net. WEB API 2**. Som backend. **Enkelt** att skapa **web-api** med det.
  + Smidig validering med data annotations.
* **Microsoft SQL Server 2008**
  + Funktionsrik relationsdatabas.

Dessa valde jag av olika anledningar

6 Varför MS SQL 2008 R2

* Varför MS SQL?
  + **Constraints** som förhindrar **dubbelbokningar**
  + Många **relationer** i **bokningssystem**. Därför passar **relationsdatabas** bra.
  + Kollade även in **MYSQL** **Problem** med **constraints**.
  + **Gillade det** i **tidigare** kurs
* Varför ASP.NET Web API 2
  + **Starkt typat** språk, **hjälper** till vid **validering**.
  + **Data annotations** ger **smidig** validering.
  + Smidigt och **enkelt** att **skapa** **REST** API:er
  + ADO.NET

7 Varför Angular.js

* Varför Angular.JS?
  + **Databinding** gör det **enkelt** att **presentera** **data** i **realtid** när de **hämtas** ifrån backenden.
  + Bra stöd för **enhetstestning**
  + Rätt så enkelt att göra **formulär validering**.
  + **Återupperbar** kod med **directives.** Egna taggar om man så vill.
  + Snabbt!
  + Ville lära mig mer javascript.
  + Detta är framtiden

9 Målgrupper för gränssnittet

* Väldigt många målgrupper
* Ställer krav på att det är väldigt tydligt och lättanvänt.
* Alla **målgrupper** har sina **egen definition** av vad ett **bokningssystem** är för dem.
* Exempelvis vill **präster** kunna se **sitt eget schema** och **kyrkornas bokningar** på **startsidan.**
* Kökspersonal vill se matrelaterade bokningar.
* En utmaning att uppfylla allas önskemål.

10 Utformningen av gränssnittet

* **Målet** har alltid **varit** att det ska vara **Logiskt**, **Lättanvänt**, **Snabbt**. **Funktionsrikt**, **Överskådligt** och **Mobilanpassat**.
* Men i verkligheten utvecklades gränsnittet utifrån databasen
* En utmaning att uppfylla allas önskemål.

11 Första versionen av Gränsnittet

* **Kalendervyn**, för att få en **inblick** över **dagars bokningar.**
* **Sök-knappen** och **Plus-knappen onödiga** och förvirrande.
* **Undermenyerna** låg för långt till vänster.
* Förväntad **dropdown** på **huvudmenyer** existerade inte.
* **Navigering** gav ett **rörigt** **intryck**.
* Grått och trist

12 Andra versionen av Gränsnittet

* **Dropdown** på huvudmenyer.
* Onödiga knapparna vid huvudmenyn borttagna.
* **Flikar** för snabbare navigering.

13 Skapa lokalbokning

* **Dynamisk validering**
  + Fel **om** **startdatum** är **före** **slutdatum**.
* **Lediga lokaler** visasför valt start och slutdatum.
* **Om värdet fylls** i korrekt försvinner de röda felmeddelanderna direkt.

14 Redigera lokal/plats 1

* **Bra med JS**
  + Vald **bild visas** direkt, men **sparas bara** om **Spara** klickas på.

15 Redigera lokal/plats 2

* **Google maps** förGPS **koordinater**
  + **Går inte** att **spara** **förrän** fälten är **korrekt** ifyllda.

16 Problem på vägen 1

* **Grundfunktionalitet** glömdes bort.
  + **Endast funktioner som ligger långt framtiden.**
  + **För många önskemål** i kravspecifikationen.

17 Problem på vägen 2

* **Gränssnittet** fick **anpassa** sig efter **databasens CRUD** funktionalitet.
* **Går att förbättra i efterhand.**

18 Slutsats Web API 2

* Kan ha **många** **front ends,**
  + Mobilappar
  + Andra **servrar** eller andra **Mashups** appar kan använda API:n.

19 Slutsats Angular.js

* Sidan laddas bara in **en gång**.
  + **Endast** **JSON** **skickas** emellan **Web API:n** och Angular klienten.
  + **Smidigt** klient emot REST apier.
  + **Omedelbar validering** av data.
  + Stöd för **egna taggar** genom **Directives.**

20 Slutsats Angular.js

* Kommer att ha **svårt att** **byta** ifrån javascript frameworks och rest api:er.
  + Gillade dem så pass mycket.
  + Så många fördelar.

Näst sista bilden. Kuden har för lite kunskap för att ge feedback.

Framtiden?

Resursbokningar och Matbokningar.

Fler målgrupper förutom lokalbokningsadministratörer. Präster, pedagoger, kyrkogårdspersonal.

Mobilanpassat gränsnitt.