Entwicklungsdokumentation

**Netzalist Software Plattform**

Version 1.0 vom 13.10.2013

Autor: Tobias Waggoner

# Über dieses Dokument

Dieses Dokument ist die Hauptdokumentation für die zu entwickelnde Softwareplattform der zu gründenden Netzalist GmbH&Co.KG. Es beinhaltet in den einzelnen Sektionen sowohl Architektur und Design als auch Backlog und Historie. Ziel ist im Sinne eines agilen Ansatzes die Komplexität der technischen Dokumentation so gering wie möglich zu halten und erst bei Bedarf via „Refactoring“ weitere Tools (Jira, Wiki, Ticketing, …) einzuführen.

Inhalt

[1 Über dieses Dokument 2](#_Toc371443553)

[2 Historie 4](#_Toc371443554)

[3 Backlog / User Stories 5](#_Toc371443555)

[3.1 Requirements 5](#_Toc371443556)

[3.2 Stories 5](#_Toc371443557)

[3.2.1 NSP – 001 Seite existiert (Erledigt) 5](#_Toc371443558)

[3.2.2 NSP – 002 Layout und Anmeldung 5](#_Toc371443559)

[4 Architektur 7](#_Toc371443560)

[5 Design 8](#_Toc371443561)

[6 Implementation 9](#_Toc371443562)

# Historie

|  |  |  |
| --- | --- | --- |
| Datum | Name | Text |
| 13.10.2013 | TW | Minimal Solution aufgesetzt – kein Backlog, keine Requirements – hatte einfach das Bedürfnis zu starten. Im Kopf steht es und ich glaube an agiles Entwickeln. Test first. Tue das mindeste um etwas zum Laufen zu bringen.  Azure Account angelegt inklusive Website und Mini DB.  <https://manage.windowsazure.com/?whr=live.com#Workspaces/WebsiteExtension/Website/netzalist/quickstart>   * [Tobias.waggoner@ropedevelopment.onmicrosoft.com](mailto:Tobias.waggoner@ropedevelopment.onmicrosoft.com) /defaultpw aber mit „!   Seite: <http://netzalist.azurewebsites.net/>  Solution nach Github deployed und mit Azure verknüpft  <https://github.com/tobiaswaggoner/Netzalist.LeadManager>  Ewig an der Datenbankmigration geschraubt.  Die Kurzform: Visual Studio: Package Manager Console  Jedes Mal, wenn es Änderungen an der DB geben muss:  add-migration <Name der Migration> eingeben.  Die App patched sich selber. Manuell kann man auf der Konsole „update-database“ eintippen.  Das ist schon mal super cool.  Jetzt noch das Ganze mit Unit Tests und einem Build Server verknüpfen und es ist bereit, um mit Inhalten befüllt zu werden.  Außerdem in Github einen Deployment Branch und in Azure einen zweiten Server (bis 40,00€ pro Monat ist für MSDN Kunden kostenlos!)  Dann ist das schon mal extrem professionell. |

# Backlog / User Stories

## Requirements

## Stories

### NSP – 001 Seite existiert (Erledigt)

**Beschreibung:**

Die Solution ist aufgesetzt. Es existiert ein MVC4 Projekt. Dieses kann automatisch per Klick nach Azure deployed werden und ist dann im Internet ansprechbar.

**Akzeptanzkriterien:**

Als Benutzer kann ich eine URL im Internet ansteuern und sehe das Netzalist Logo.

**Protokoll:**

13.10.2013, tw 4 Stunden

* Azure Account und Site angelegt
* Verknüpft mit Github
* Automatisches Deployment inklusive DB Migration (Up/Down)

### NSP – 002 Layout und Anmeldung (Erledigt)

**Beschreibung:**

Das Template MVC Standard Template ist zum Leadmanager umgebaut inklusive Logo und Login Screen. Benutzer können sich anmelden und sehen die leere Anmeldemaske, bzw. können (zunächst noch ohne Rechteverwaltung) die Datenbankupdates / Rollbacks durchführen.

**Akzeptanzkriterien**

Als Benutzer sehe ich die Leadmanager Start Seite und kann mich an- und wieder abmelden. Die Seite wirkt professionell designed.

**Protokoll:**

28.10.2013, tw 3 Stunden

* Eigenes Login (Custom Forms Login) mit Mandant
* Default Login entfernt
* Unnötige Seiten / Menüpunkte entfernt
* Gesichert (Login jetzt erforderlich).

### NSP – 003 Erfassung von Firmen

**Beschreibung:**

Als Benutzer möchte ich Firmen erfassen können. Die Firmendaten werden zunächst in nur einer Tabelle gespeichert. Die Firmendaten umfassen Firmenname, Anschrift mit LKZ, Telefon, Fax, eine zentrale EMail und eine Web Seite.

Außerdem möchte ich Firmen suchen, anzeigen, editieren und löschen können, sowie per Klick auf die URL auf die Homepage navigieren.

**Akzeptanzkriterien:**

Im Menü kann ich zur Firmenliste navigieren, neue Firmen erfassen, editieren und löschen.

**Protokoll:**

05.11.2013, tw 3 Stunden

* Kampf mit Entity Framework
* Gelernt: Foreign Keys besser explizit ins Model
* Zentrale Instanz des DataContexts (sonst Probleme beim Update).

# Architektur

# Design

# Implementation