

Ivica Anic
Halden weg 5
71409 Schwaikheim
Telefon :07195/9859215
Email: delfa.anic@gmail.com



Ausbildung

- Diplom-Ingenieur der Technologie
- Betriebsinformatiker
- SUN Certified Programmer für Java, MCP
- Microsoft Certified Professional .NET
- seit 20 Jahre bilde ich mich regelmässig weiter mit Fachbücher,online Kurse , ...

Personalien

Geburtsdatum: 12.06.1966
Wohnort: Deutschland,Schwaikheim seit 1992
Nationalität: Kroatien

Berufserfahrung

-20 Jahre Berufserfahrung in Deutschland als Programmierer und Softwareentwickler/-Tester

Beruflicher Werdegang

20.07/2020 -bis dato
Softwareentwickler/-tester
Branche:Intralogistik

arbeite als C/C++ Softwareentwickler an Projekten für automatisierten Lagerung-Software (LVS,MFS)-automatisierten Lagerverwaltung
Aufgaben:
Intralogistische Lager-Vorgänge in C++ entwickelten Software abzubilden.
Logische Lager-Abbildung in SQL-Server oder ORACLE Datenbank
Aufgrund des Projekt-Vorgaben, Entwicklung und Test neuer Funktionen,
Entwicklung der C++-Schnittellen :
Kommunikation mit SPS über Telegrammen
SPS-Komuniziert mit Förder-Technik und Roboter.SPS-status-Telegramen in C++-

03/2020 -bis 14.07 Softwareentwickler/-tester

Branche:Energiewirtschaft

**09/2019 -bis 02/2020
Softwareentwickler/-tester**

Firma:regisafe GmbH

Branche:Dokumentmanagement

09/2015 – 08/2019 Softwareentwickler/-tester

Firma: proAvia GmbH

Branche:Luftfahrt

folgende Aufgabe habe selbstständig implementiert

Schnittstellen zu empfangen und weiter zu bearbeiten.Mit SPS-Software-Simulation die Lager-Vorgänge zu testen

Entwicklungs-Tool:Borland C++ CodeGear

Datenbanken:ORACLE und SQL-SERVER

Version-verwaltung:JEDY-Tool

-Angular UI (Microsoft Visual-Code)

-Linux C-Programmierung (vi-Editor) mit ORACLE-DB anbindung.

Version-verwaltung:git

bitte siehe von mir erstellte angular-

Besispiele unter folgende Link

<https://samples-ivica-anic-angular.web.app/>

-C++ Foxit-PDF-Compressor-API in C++-Anwendung einzubinden

-C++ OCR-tesseract-Framework zu untersuchen und in den Projekt anzuwenden

-ePost-WebApi-Schnittstelle in C++

Anwendung anzubinden und zu testen,

-Microsoft-Word-Serien-Briefe mit ePost-WebApi in C++ zu versenden

Entwicklungs-Tool:Microsoft-VisualStudio

Version-verwaltung:svn

-NET core C# Restful-WebApi mit anbindung zur SQLServer (als DB-Mapper wurde Dapper benutzt), automatisch EMail versenden

-.NET core ASP.NET-MVC Anwendung mit zugriff auf SQLServer DB

- ZVT-Kassenschnittstelle testen und erforschen

-AXIS-Kamera-Schnittstelle (cgi)

anzusprechen,der AXIS-Kamera Fahrzeug-

Kennzeichen-Erkennung (LPR) zu erforschen

(AXIS-Kamera-Schnittstelle (cgi) testen und

erforschen) ,anschliessend produktiv einzusetzen

- der HIKVISION-Kamera LPR-Schnittstelle zu erforschen

-Microsoft SQL Server Analyseservice testen und erforschen,anschliessend produktiv einzusetzen (BIG DATA Analyse)

mit Visual Studio Business Intelligence

Analyseservice-Projekte

(tabelarisch,mehrdimensional)

mit zugriff auf SQL-Server-DB zu

erstellen,Analyseservice Projekte auf SQLServer Deployen (CUBE Analyse)

mit WPF-C1-OLAP-Komponente (VB.NET) auf

CUBE zuzugreifen und BIG-DATA-Analyse in OLAP-Format (PIVOT-Tabelle, EXCEL-Format) zu ermöglichen

-Microsoft MachineLearning zu erforschen, aufgrund vorhandene DB-Daten die Firmenumsatz voraussagen

die Aufgaben: mit .NET C# Restful-WebAPI entwickeln und testen

.NET VB.NET RestFul-WebAPI-Schnittstelle mit Zugriff auf der SQL-Server-DB (Entity-Framework), die Firmen-Lieferanten konnten von aussen RestFul-WebAPI-Schnittstelle ansprechen

-Microsoft Signal-R-Framework (Self-Host) testen und erforschen anschliessend produktiv einzusetzen

-.NET core C# über Benutzer-Oberfläche neu SQL-Server-Datenbank mit Benutzer-Berechtigung zu erstellen

iPad App für der Flugzeugen-Leistung-Übersicht (Custoemr Reporting App)

iPad Customer-Reporting-App habe selbstständig in objective-c und swift entwickelt

iPad App Anforderungen:

beim starten der iPad-App wird Login-Screen angezeigt, die/der Benutzer/in meldet sich in der App an. Aufgrund der App-Benutzer-Anmeldung werden zur Benutzer in DB-zugewiesene Flugzeuge (Flugzeug-Bilder) in der iPad-Client-App dynamisch geladen.

Der Flugzeug-Bilder werden mit XCODE Cower-Flow-Effect in iPad-App dargestellt (Flugzeug-Bilder können dynamisch bewegt werden in den App-Screen).

Aufgrund der aktuellen ausgewählter iPad-Client-Flugzeug-Bild, werden entsprechend dem ausgewählten Client-Kachel-Button-Action, der DB-Server-Flugzeug-Daten, über WebServer-json-WebApi Aufruf in den Client-iPad-App geladen. App-Design wurde von UI-Designer mit Photo-Shop als .png-File entwickelt. .png-File habe ich in App-UI eingebunden und dynamische App-UI-Inhalte aufgrund des vorgegebene x und y Koordinate gezeichnet.

Das Projekt habe in App Store deployed

Die iPad -App läuft in 14 verschiedene Ländern bei Verkaufsagenten.

1. Version-Control Tool : GitHub
Client-Seitiges Architektur
2. Entwicklungstool:

3. XCODE als Entwicklungssprache habe objective-c und swift angewendet,
4. das Frontend habe mit MVC Design Pattern implementiert
5. App-UI für iPad-Portrait und iPad-Landscape Orientierung
6. Client-Seitiges json-C#-WebApi aufrufe
7. CocoaPods ,Xcode-Multithreading
8. Client-Seitiges Certificate in json-WebApi Clients-Requests eingebunden
Server-Seitiges Architektur
9. Entwicklungs-Tool Visual Studio 2015
10. Business-Logik :C# WebApi (Rest-json) mit Dapper (micro-ORM mapper)
11. WebServer:InternetInformationService
12. 2014 [DB:SQL-Server](#)

Branche:Luftfahrt

folgendes C#Xamarin Projekt (iPhone-Samsung App) habe selbständig (alleine) (UI) Frontend und (Server) Backend implementiert
die Aufgabe Beschreibung:

der mobile Geräte (iPhone,Samsung) App für der Piloten-Diensteinteilung-Anzeige und der Urlaubsantrag per iPhone bzw. Samsung Geräten

Samsung und iPhone-App Anforderungen:
beim Samsung bzw. iPhone App Start aufgrund User-Anmeldung werden entsprechend dem Angemeldeten Benutzer , der DB-Server-Piloten-Diensteinteilungen-Daten , über WebServer-json-WebApi Aufruf in den Client-iPhone-(Samsung)App geladen.Alle Piloten-Diensteinteilungen bzw. Piloten-Proceessing werden als dynamischen Labels bezogen für entsprechendes Tag dargestellt.

Urlaubs-Antrag:Piloten können Weltweit über iPhone bzw. Samsung App den Urlaubs beantragen,
über App die Piloten können der Status des Urlaubs-Antrags sehen.

Iphone-Samsung App läuft Welt-Weit bei 300 Piloten

1. Entwicklungs-Tool :Visual Studio Community für Mac
2. Version-Control: GitHub
3. iPhone-Samsung-App habe in Schichten Architektur entwickelt
4. Client-Seitiges-Architektur:
5. Xamarin-XAML-Forms

6. MVVM-Pattern
 7. Client-Seitige json-WebApi-Aufrufe mit Client-Certificaten Einbindung
 8. Responsive-XAML-UI-Design für alle iPhone Versionen
 9. iPhone-Droid-App läuft auf alle iPhone bzw. Samsung Versionen, habe Responsive- XAML-Design implementiert, Responsive-XAML-Design habe auf XAMARIN-iOS-Simulatoren und Droid-Simulatoren (Google Emulator) getestet
1. Server-Seitiges-Architektur
 2. Server:RESTful json WebApi mit Dapper (micro-ORM mapper)
 3. WebServer:InternetInformationService
 4. 2014 [DB:SQL-Server](#)

Aufgabe Beschreibung

die iPad App für der Flugverwaltung folgendes XCODE objective c Projekt habe selbständig (alleine) (UI) Frontend und (Server) Backend implementiert

die Anwendung wurde in Schichtenarchitektur entwickelt

Entwicklungstool:

XCODE als Entwicklungssprache habe objective-c angewendet,

das Frontend habe mit MVC Design Pattern implementiert

erste Version der Anwendung habe im Jahr 2014 mit SOAP Schnittstelle implementiert

zweite Version der ipad-Anwendung habe im Jahr

2017 mit RESTful WebApi implementiert

RESTful WebApi habe selbständig in C# mit Anbindung auf SQL-Server DB implementiert

als Entity Framework habe Dapper

(stackoverflow nuget package) eingebunden

folgende Funktionalität habe implementiert:

habe selbständig der Anwendung .ipa File mit

Einbindung von Apple Certificate und

Provisioning Profile erstellt

Aufgabe Beschreibung:

software für Audit-Verwaltung

Folgende Projekt habe selbstständig implementiert

.NET Projekt - Entwicklung und Test der C# WPF

Anwendung in Srcum Team

- die Anwendung wurde in der Schichtenarchitektur entwickelt (MVVM Pattern).
- Business Logik wurde mit der Einbindung Dapper-FrameWork (stackoverflow Nugget package) entwickelt.
- Datenbank:SQL Server 2014 und 2016 – Kommunikation Client-Server über Restful C#-WebApi Services,
- FrontEnd am Client :WPF XAML File mit der Einbindung Microsoft-Xceed Controls.
- printing wurde mit itextsharp nuget package implementiert
- logging wurde mit log4net implementiert
- Entwicklung der Javascript Schnittstelle
- Unit Test für Business Logik Testing
- Microsoft-Tools:VisualStudio (2013,2015,2017),GitHub als Repository
- deploying der WebApi auf Internet Informations Service
- Kunden Support, der Fehler Behebung

[ASP.NET](#) [VB.NET](#) Projekt Kunden Support und weiter Entwicklung

- Die Anwendung wurde mit [ASP.NET](#) in [VB.NET](#) entwickelt.
- Server-Seitige Business Logik wurde mit der Einbindung .NET Entity Framework entwickelt
- Datenbank:SQL Server 2014 und 2016
- UI wurde mit [ASP.NET](#) Controls und DevExpress Controls entwickelt
- javascript wurde in UI eingebunden
- Microsoft-Tools:VisualStudio (2013,2015,2017),GitHub wurde als repository verwendet

| SOFTWARE

MS Word

●●●●●●

 MS Excel


●●●●●●

 MS Powerpoint

●●●●○○

MS Projekt

●●●●○○

 MS One Note

●●●●○○

 Deutsch

●●●●●●

 Englisch

●●●●●●

Französisch

●●●●○○

Großes Latein

●●●●○○