RAUMPLANUNG

Technische Dokumentation

Stand: 31.08.2020, Version: Nr. 0.1

Inhaltsverzeichnis:

Historie

Datum	Bearbeitet von	Beschrei-bung	Bemerkung
31.08.2020	Bamberger	Erstellung	Initiale Erstellung der Dokumentation

1 Allgemein

1.1 Einleitung

Die neue Raumplanung des NJZKO ermöglicht es Räume transparent zu verwalten. Sie wird von der IT-Abteilung des OVG ständig weiterentwickelt. Aktuell ist es noch notwendig zur Buchung von Terminen eine E-Mail zu versenden. Bitte verwenden Sie einen aktuellen Browser wie Google Chrome oder Mozilla Firefox, da der Internet Explorer nicht mehr unterstützt wird. Unter http://rema.ovgvg.jmrlp.de haben Sie die Möglichkeit sich zu informieren ob Räume belegt sind und wie lange. Zusätzlich können Sie die angezeigten Räume filtern und zwischen Monats-, Wochen- und Tagesansicht wechseln. Ihre Startseite ist hierbei die Planungsübersicht. Über die Navigation an der linken Seite können Sie auf andere Ansichten wechseln. Rechts oben sehen Sie den ersten Buchstaben Ihres Namens, diesen Menüpunkt können Sie anklicken um Informationen zu Ihrem Profil im Programm zu erhalten.

1.2 Gesamtsystem

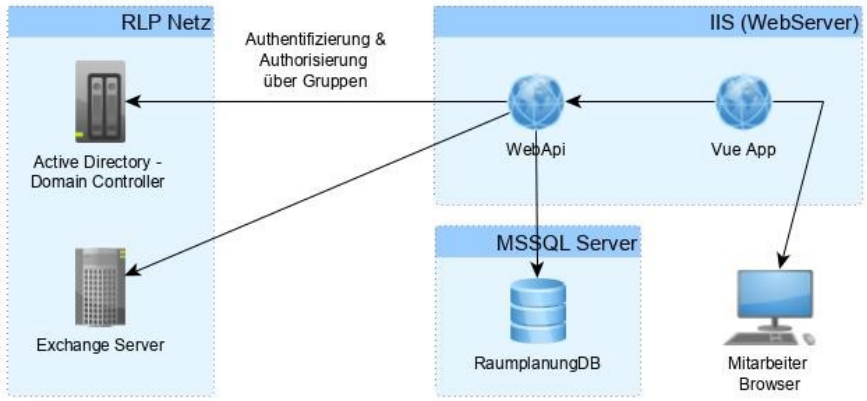

Abbildung 1: Übersicht Gesamtsystem

1.3 Eingesetzte Produkte

1.3.1 Microsoft. NET Framework

Als Framework für die Import-Komponenten kommt .NET Framework[1] zum Einsatz. Dabei handelt es sich um eine Entwicklungsplattform zum Erstellen von Anwendungen vorrangig im Microsoft Umfeld. Es besteht aus der Common Language Runtime (CLR) und der .NET Framework-Klassenbibliothek.
Der Datenzugriff ist mit dem Entity Framework[2] per Code-First Strategie realisiert.

1.3.2 Microsoft .NET Core

Als Webframework zur Realisierung der WebApi kommt Microsoft .NET-Core[3] zum Einsatz. Die WebApi ist grundsätzlich als REST[4] WebService realisiert. Mit Standardfunktionen wird durchgängig auf Dependency-Injektion[5] auf Konstruktur-Level gesetzt. Um komplexe Abhängigkeitsprobleme beim Datenaustausch zu vermeiden wird AutoMapper[6] verwendet, dieser wandelt komplexe Objekte in flache Austauschformate und Umgekehrt.

1.3.3 Vue.js

Als Front-End-Webapplikationsframework kommt Vue 2[7] zum Einsatz. Die UI ist mit Hilfe der Komponenten Bibliothek Vuetify realisiert, der selbst definierte CSS-Code ist somit auf ein Minimum reduziert. Die Web-Anwendung ist in Form eines Fat-Client konzipiert. Bei der Initialisierung werden alle relevanten Daten geladen und per Vuex-Orm in eine Relatione Struktur gebracht, innerhalb einer Flux-Architektur (Single source of

---

[1] https://docs.microsoft.com/de-de/dotnet/framework
[2] https://docs.microsoft.com/de-de/ef/core/
[3] https://docs.microsoft.com/de-de/dotnet/core
[4] https://de.wikipedia.org/wiki/Representational_State_Transfer
[5] https://de.wikipedia.org/wiki/Dependency_Injection
[6] https://automapper.org/
[7] https://vuejs.org/

truth). Vuex-Orm greift hierbei direkt auf die Rest-Schnittstellen zu und hält somit unter Verwendung der CRUD-Operationen den lokalen Datenstand aktuell mit jenem auf dem Server. Bei jeder CRUD-Operation an einem Termin, werden die Termindatensätze aktualisiert.

2 Grafische Benutzerschnittstelle

Zur Verwendung wird der Google Chrome Browser empfohlen aufgrund seiner Verwendung in der Justiz und seines aktuellen und sicheren Funktionsumfangs. Andere moderne Browser funktionieren größtenteils korrekt, allerdings ist mit Funktionseinschränkungen zu rechnen. Der Internet Explorer wird explizit nicht unterstützt. Im Mozilla Firefox funktioniert die „Termin drucken“ Funktion nur über den Umweg eines automatischen Downloads, im MS Edge funktioniert sie gar nicht.

2.1 Authentifizierung

Die Authentifizierung erfolgt passiv über das Active Directory. Es ist keine aktive Handlung des Nutzers erforderlich. Zugriffe sind nur von angemeldeten Nutzern innerhalb der Domain möglich, durch diese Authentifizierung ist auch eine Identifizierung und Authentifizierung der Nutzer realisiert. Beim ersten „kontakt“ der Anwendung mit einem Nutzer, wird für diesen ein Nutzerkonto in der Datenbank erstellt. Ein solcher Kontakt kann zB. durch den Aufruf der Webseite durch den Benutzer entstehen oder auch durch das Auswählen eines Nutzers als Ansprechpartner für einen Termin. Gespeichert wird neben Namen und Organisationseinheit auch die ID des Nutzers im AD. Diese ermöglicht eine Verbindung zwischen den Informationen aus der AD und dem Nutzer Datensatz in der DB.

2.2 Autorisierung

Generell kann jeder Nutzer innerhalb der Domain Termine ansehen, zur Aufteilung der Berechtigungen existieren drei Rollen, diese müssen Gruppen in der AD zugewiesen werden. Die Berechtigung von

3 Installation

Um das Raumplanungssystem zu installieren werden folgende Komponenten vorrausgesetzt:

- Eine MSSQL Datenbank Instanz, auf welcher das Datenbankschema aufgesetzt wird.
- Ein Aktive Directory mit Gruppen für die Rollen:
    - Administrator
    - Bearbeiter
    - Benutzer

  Wobei auch mehrere Rollen einer Gruppe zugeordnet werden können.
- Einen Webserver, bisher IIS, welcher die .Net Core Anwendung ausführen und die Web-App ausliefern kann.
- Einen Email-Server sowie ein Konto welches von diesem E-Mails verschicken kann.

4 Updates

Um Updates möglichst einfach zu ermöglichen gibt es im vue-Projektordner ein script innerhalb der package.json: „npm run build-all“.

Dieses erstellt sowohl die Web-Anwendung als auch die Webserver Software, kopiert alles in einen Ordner und erstellt im Wurzelverzeichnis ein Zip-Archiv mit Namen und Versionsnummer des Programms. Die erforderliche Konfigurationsdatei sind innerhalb diese Archivs umbenannt, sie enthalten das Wort „template“ zwischen Dateinamen und Dateiendung. Um eine neuen Instanz aufzusetzen müssen diese Dateien also umbenannt und mit eigenen Daten gefüllt werden. Falls sich mit einem Update die Inhalte der Konfigurationsdateien ändern können diese umbenannten Konfigurationsdateien als Referenz herangezogen werden.
Der Inhalt des Zip-Ordners kann also zum Updaten einfach in das Wurzelverzeichnis einer Instanz kopiert werden. Falls Konfigurationsänderungen oder Datenbankschemaanpassungen notwendig sind müssen diese noch vorgenommen werden. Das Datenbankschema lässt sich updaten durch Verwendung des aktuellsten Update-Scripts im Ordner „Rema.DbAccess -> Migrations“ des Projekts.