AR-Plattform

Entwicklung einer Augmented Reality Plattform

Eine Präsentation im Rahmen der Ausbildung zum Fachinformatiker Anwendungsentwicklung
Sommer 2019

Quantumfrog

Timo Teschke

Agenda

Vorstellung

Ausgangssituation

Analyse

Entwurf

Implementation

Dokumentation

Fazit und Aussicht

Agenda

Vorstellung

Ausgangssituation

Analyse

Entwurf

Implementation

Dokumentation

Fazit und Aussicht

QUANTUMFROG GMBH

TGO Oldenburg

Seit 2011

Individualsoftware, Webseiten, Applikationen

Quantumfrog

100

Agenda

Vorstellung

Ausgangssituation

Analyse

Entwurf

Implementation

Dokumentation

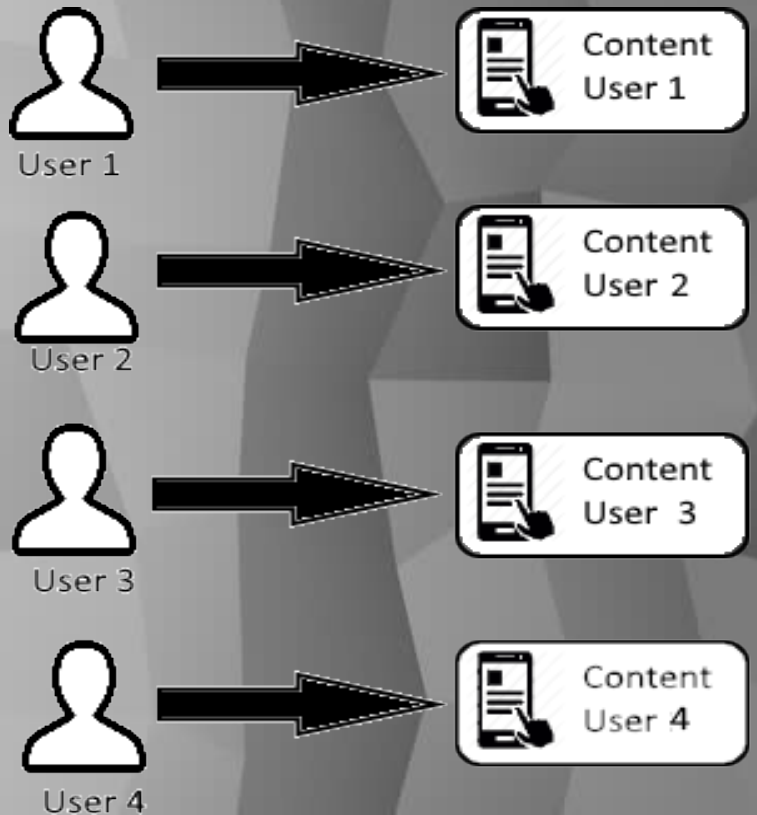
Fazit und Aussicht

ALTES SYSTEM

User benötigen individuelle Applikationen

Grundsätzlich selbe Codebasis

Inhalte statisch hinterlegt



PROBLEME

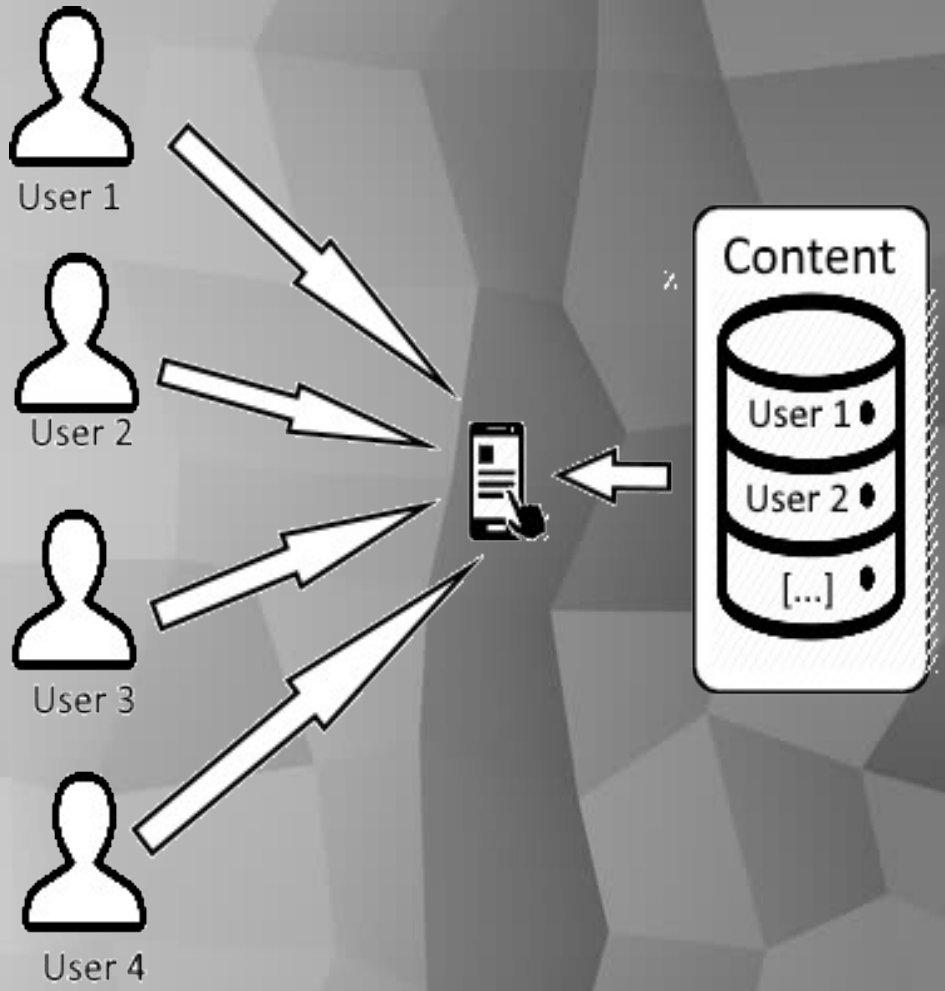
Updates müssen für alle Applikationen übernommen werden

Mehraufwand

Langer Zeitraum für neue Kundenwünsche,
Da Entwickler diese manuell einpflegen

Eine Applikation für die Gesamtheit der Nutzer

Nutzer sollen selbst Inhalte einpflegen
und Verwalten können



Agenda

Vorstellung

Ausgangssituation

Analyse

Entwurf

Implementation

Dokumentation

Fazit und Aussicht

Make OR Buy

??

PROJEKTKOSTEN - STUNDENSATZ

Jahresstunden: 8 Stunden * 220 Tage = 1760 Stunden

Stundensatzberechnung: (Monatsgehalt * 12) / Jahresstunden

	Auszubildender	Mitarbeiter	Geschäftsführer
Monatsgehalt	1000 €	2500 €	6000 €
Jahresgehalt	12.000 €	30.000 €	72.000 €
Stundensatz	6,82 €	27,05 €	40,91 €

Ressourcenkosten: Pauschal 15,00 € / Stunde

PROJEKTKOSTEN

70H	Entwicklung	6,82 € + 15 €	1527,40 €
2H	Abnahme	40,91 € + 15 €	111,82 €
5H	Schulung	17,05 € + 15 €	160,25 €
	Gesamtkosten		1799,47 €

AMORTISIERUNG

Erwartete jährliche Einsparungen	4422,90 €
Projektkosten	1799,47 €

$$Amortisationszeit(t) = \frac{Projektkosten}{Durchschnittlicher\ jährlicher\ Gewinn}$$

$$t = \frac{1799{,}47€}{4422{,}90€}$$

$$t \approx 0{,}41\ Jahre \approx 5\ Monat$$

Agenda

Vorstellung

Ausgangssituation

Analyse

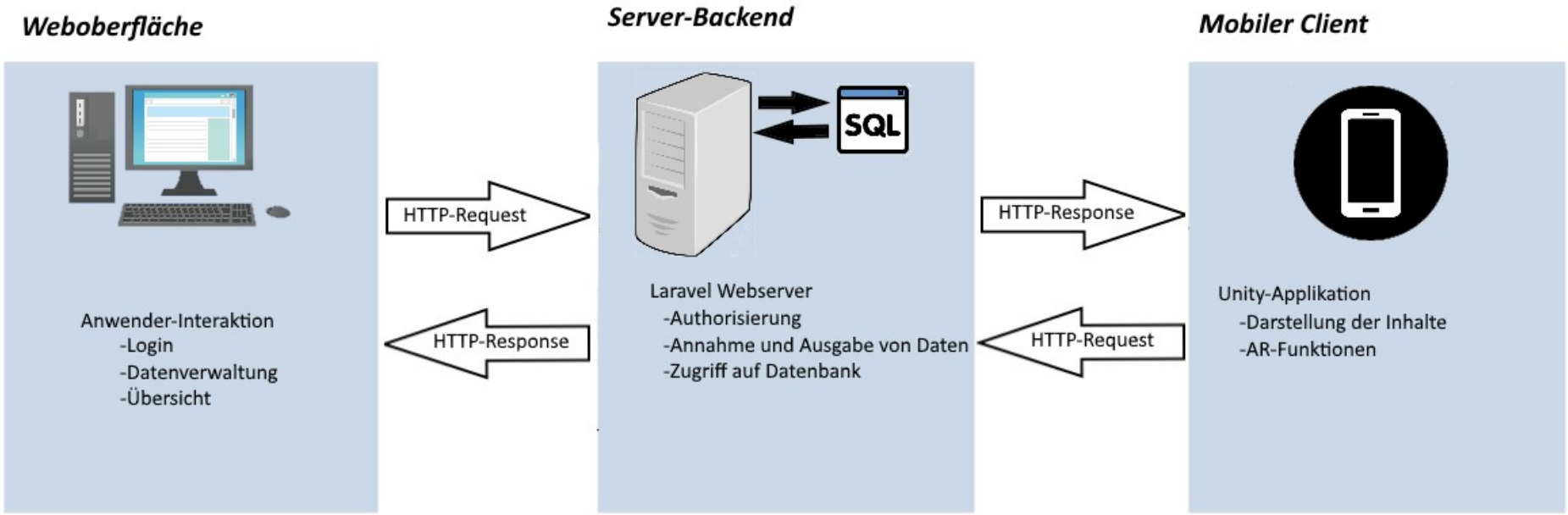
Entwurf

Implementation

Dokumentation

Fazit und Aussicht

KOMMUNIKATION UND KOMPONENTEN

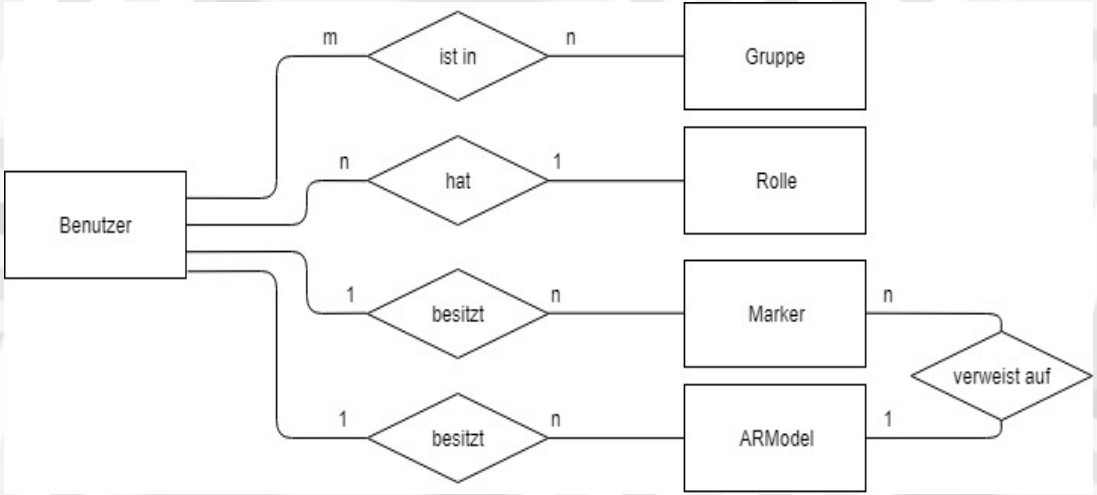


TECHNOLOGIEN

ENTITY-RELATIONSHIP-MODEL

Interpretation der Anforderungen

Übersicht über die zu erstellenden Entitäten

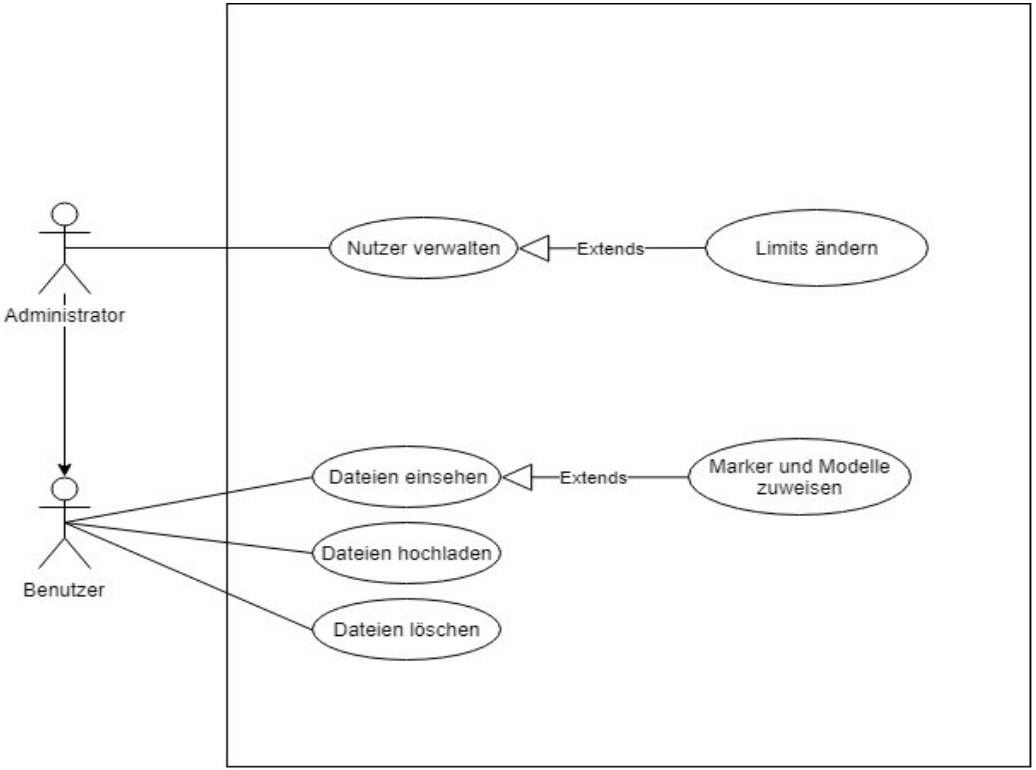


USE CASE DIAGRAMM

Anforderungen aus Lastenheft

Sicht des Anwenders

Aktionen und Nutzerinteraktionen

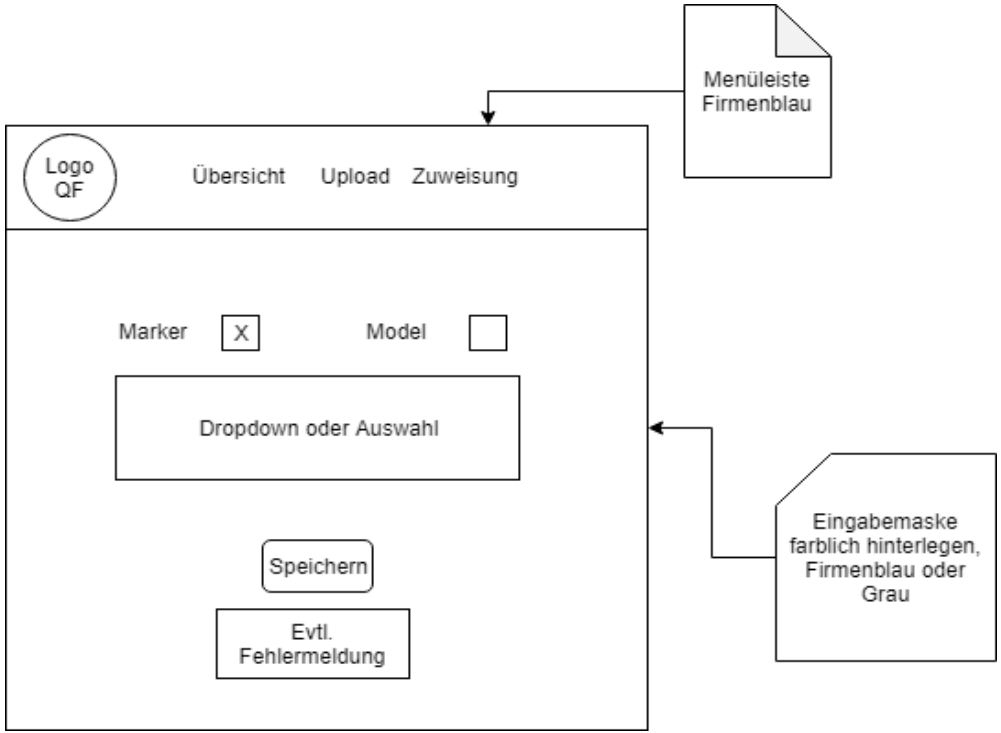


BENUTZEROBERFLÄCHE WEB

Intuitiv gehalten

Mobil nutzbar

Kein direktes „Branding“



PFLICHTENHEFT

Konkrete Beschreibung

Umsetzung der Anforderungen

„Gegenstück“ vom Lastenheft

- Backend-Server
  - Der Server wird Zugang zu einer MySQL-Datenbank besitzen, in der Daten gespeichert, ausgelesen und gelöscht werden.
  - Separat werden Marker und 3d-Objekte als Dateien direkt auf dem Server gespeichert, wobei ein Zugriff von außen nicht ohne Authentifizierung möglich ist
  - Das Framework „Laravel“ wird genutzt, wodurch sich eine „MVC“-Architektur ergibt
  - Der Server muss Schnittstellen zum Bearbeiten, Erstellen, auslesen und löschen der Daten anbieten und diese durch eine Autorisierung schützen.

Agenda

Vorstellung

Ausgangssituation

Analyse

Entwurf

Implementation

Dokumentation

Fazit und Aussicht

IMPLEMENTATION - DATENBANK

ORM „Eloquent“ genutzt

Neue Datenbank anlegen

Zugriffsinformationen im Projekt hinterlegen

Migrationen basierend auf dem Datenbankmodell erstellen

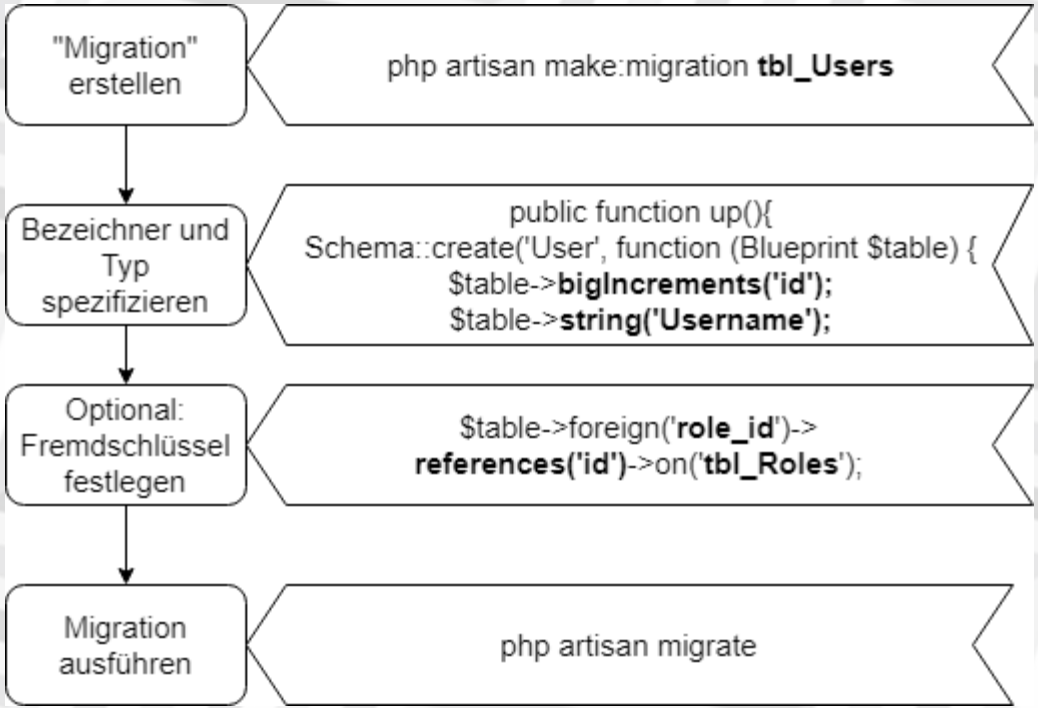
IMPLEMENTATION - MIGRATIONS

Über CLI „Artisan“ Grundgerüst erstellen

Einzelne Felder spezifizieren

Verbindungen angeben

Migration ausführen



IMPLEMENTATION – ROUTING - LARAVEL

Endpunkte definieren

Umgang mit HTTP-Anfragen spezifizieren

Sowohl nach Endpunkten als auch nach Methoden unterscheiden

```
Route::post("/armodel/create","ARModelController@store")->middleware("auth:api");
```

1	2	3	4

IMPLEMENTATION – LOGIK - LARAVEL

Endpunkte mit auszuführenden Anweisungen definieren

Benutzerauthentifizierung einbinden

Dateiverwaltung einrichten, „Controller“ implementieren

IMPLEMENTATION – LOGIK - LARAVEL

User überprüfen

Bei Markern: Limit prüfen

Request validieren

Datei speichern

Model erstellen

```php
/**
 *
 * Store a given ARModel
 *
 * @param       Request  $request HTTP-Requestdata that holds Information about the user and data
 * @return      response
 *
 */
    public function store(Request $request)
    {
        $user = $request->user();
        if ($user == null) {
            return response()->json(['data' => "unauthorized"], 403);
        }

        $request->validate([
            "file" => "required",
            "title" => "required",
            "filetype" => "required",
        ]);

        $file = $request->file->store("armodels");

        $arm = new armodel([
            "ressourcename" => $file,
            "user_id" => $user->id,
            "filetype" => $request->filetype,
            "Title" => $request->title

        ]);
        $arm->save();
    }
```

IMPLEMENTATION – LOGIK - FRONTEND

```
Save(formData){
    axios.post( this.backendUrl + "/"+this.ChoosenType+"/create",
        formData,
        {
            headers: {
                'Content-Type': 'multipart/form-data',
                'Accept': 'application/json, text/plain,text/html, */*',
                'Authorization': 'Bearer ' + window.localStorage.getItem("bearertoken")
            }
        }
    ).then(function(){
        this.fileList ="",
            this.Filename ="";
        this.InfoTextColor="#38FF60";
        this.InfoText = "Erfolgreich hochgeladen"

    }.bind(this))
        .catch(function(e){
            this.InfoText = "Datei konnte nicht hochgeladen werden"
        }.bind(this));
},
```

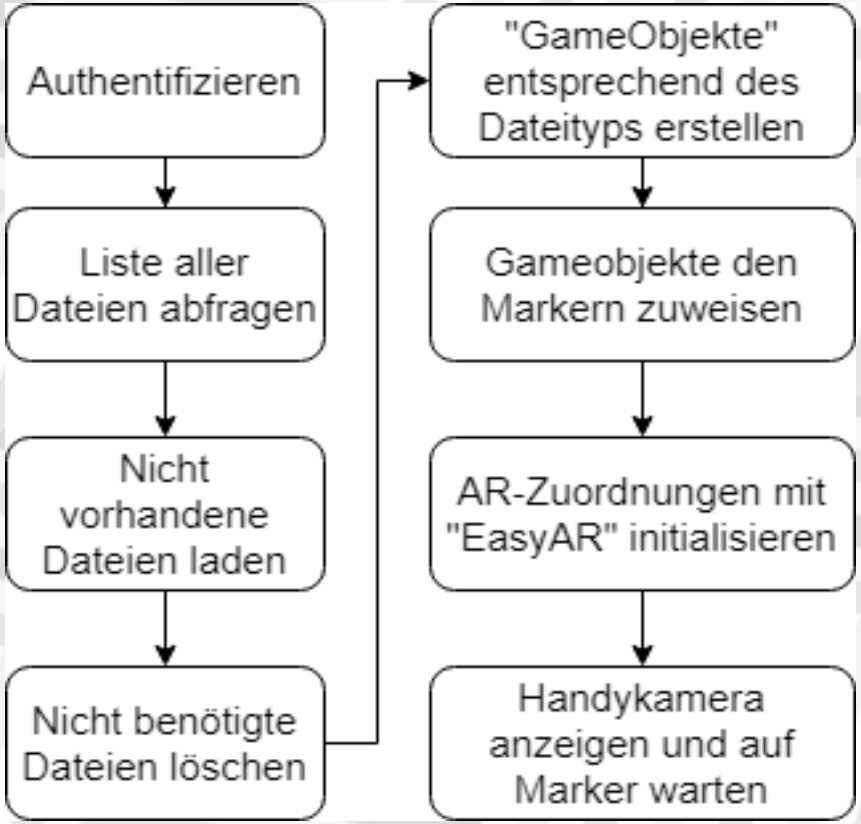
IMPLEMENTATION – LOGIK - UNITY

Anbinden der Authentifizierung des Clients

Laden und abspeichern der Dateien

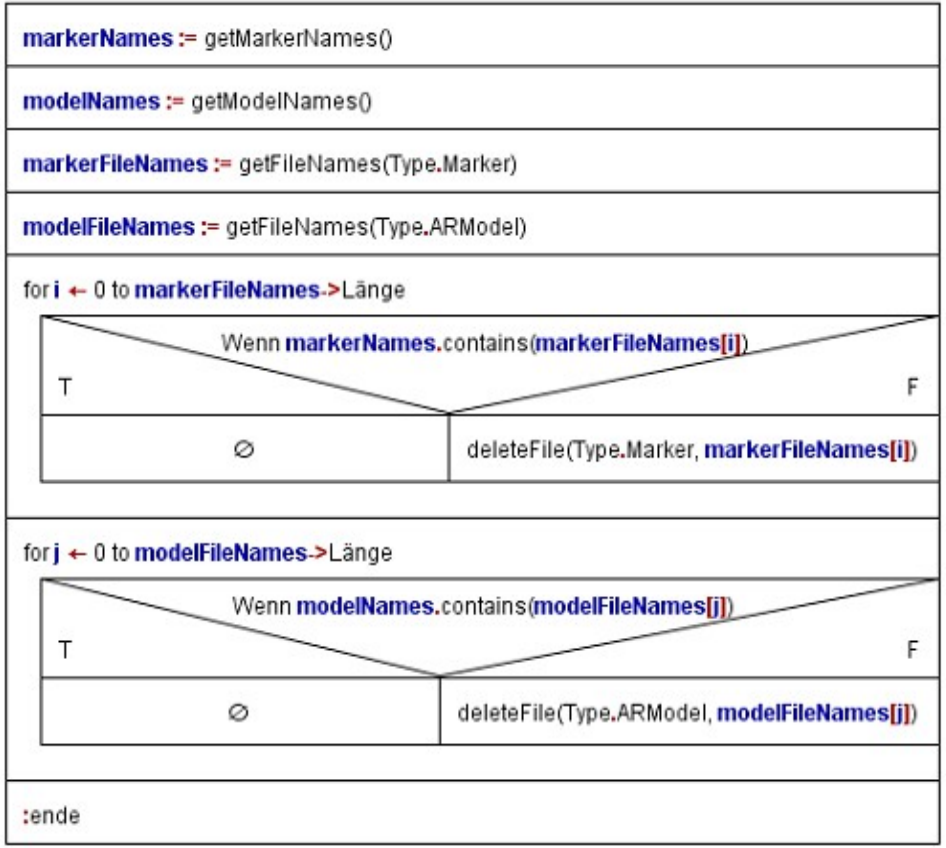
Zuordnen der Inhalte

Einbinden von „EasyAR“ zum anzeigen der Inhalte



STRUKTOGRAMM - UNITY

Löschen von nicht genutztem Content



Agenda

Vorstellung

Ausgangssituation

Analyse

Entwurf

Implementation

Dokumentation

Fazit und Aussicht

DOKUMENTATION

Einfache Benutzerdokumentation

Übersicht über die Funktionen sowie deren Bedienung

Erklärung von Fachbegriffen

TESTS

Implementierung einiger Unittests

Manuelle Tests der internen Qualitätssicherung

Agenda

Vorstellung

Ausgangssituation

Analyse

Entwurf

Implementation

Dokumentation

Fazit und Aussicht

FAZIT

Projekt wurde erfolgreich abgeschlossen

Alle Anforderungen wurden umgesetzt

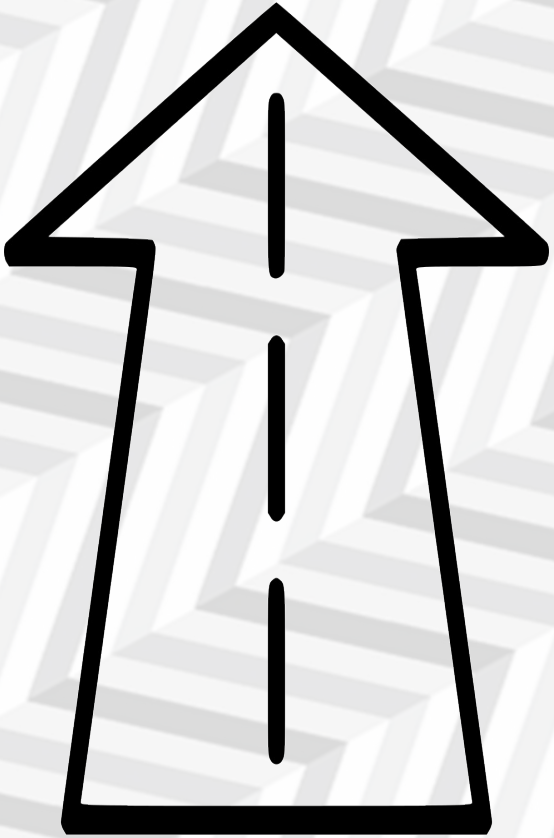
Zeitplanung konnte eingehalten werden

AUSSICHT

Implementierung als Modul in vorhandene Applikationen

Design der mobilen App erstellen

Erweiterung der hochzuladenen Medienarten
Durch beispielsweise Texturen



QUELLEN

https://vuejs.org/

https://laravel.com/

https://unity.com/de

https://www.easyar.com/

https://pixabay.com

https://icons8.de

https://thenounproject.com

https://www.tgo-online.de/

https://slidehood.com

Slidescarnival.com

VIELEN DANK FÜR IHRE AUFMERKSAMKEIT

Eine Präsentation im Rahmen der Ausbildung zum Fachinformatiker Anwendungsentwicklung
Sommer 2019

Timo Teschke