

# **Software Requirements Specification**

**UGW  
Um Gottes Willen**

Version 1.0  
17.10.2020

**Irimus Michael  
Schmitzhofer Stefan**

Software Requirements Specification  
**Inhaltsverzeichnis**

Projektname

<b>Einleitung</b>	<b>1</b>
Zweck	1
Hintergründe und Ziele des Projekts	1
organisatorische Einbettung	1
technische, wirtschaftliche, organisatorische, ergonomische Ziele	1
Produktumfang	1
Muss-Kriterien	1
Soll-Kriterien	1
Kann-Kriterien	1
Abgrenzungs-Kriterien	2
<b>Allgemeine Übersicht</b>	<b>3</b>
Beschreibung der Ausgangssituation (Ist-Zustand)	3
Produkteinsatz	3
Anwendungsbereiche und Zielgruppen	3
Betriebsbedingungen	3
Annahmen und Abhängigkeiten	3
Produktumfeld	4
Systemschnittstelle	4
Benutzerschnittstelle	4
Kommunikationsschnittstelle	4

# 1. Einleitung

## 1.1. Zweck

### 1.1.1. Hintergründe und Ziele des Projekts

Events christlicher Natur finden in Österreich nur schwer den Weg in die Öffentlichkeit. Trotz Vorhandenseins zahlreicher Veranstaltungen unterschiedlicher christlicher Denominationen, sind diese, auch durch Social Media, nur schlecht wiederauffindbar geschweige denn zentralisiert abrufbar. UGW soll diese Lücke füllen und als unterstützendes Mittel für die Suche eines passenden Events mit christlichem Hintergrund dienen.

### 1.1.2. Organisatorische Einbettung

Das Projekt wird im Rahmen des 3. Semesters des Masterstudiengangs “Software Design and Engineering” für die Lehrveranstaltung “Web Engineering” als Fullstack Gruppenprojekt unter Hr. Leon Freudenthaler umgesetzt.

### 1.1.3. Technische, wirtschaftliche, organisatorische, ergonomische Ziele

Das Ziel des Projekts ist es, eine Webapplikation von Konzeption bis zur Umsetzung selbstständig zu entwickeln, welche langfristig zur Koordination von Veranstaltungen genutzt werden kann. Obwohl das Konzept stark darauf ausgelegt ist frei verwendbar zu sein, ist eine limitierte Schaltung von Werbung und somit Kostendeckung oder damit verbundener Gewinn nicht ausgeschlossen. Dennoch liegt das aktuelle Hauptziel in der Bereitstellung der Funktionalität zum Allgemeinwohl und Gewinne vorerst nebensächlich und der Eigennutzen eher sekundär durch die damit verbundene Marktpräsenz und ggfs. Imagesteigerung der Entwickler.

## 1.2. Funktionale Anforderungen

### 1.2.1. Muss-Kriterien (FR)

- Veranstaltungen müssen dynamisch in einer Listenansicht angezeigt werden. (Irimus)
- Veranstaltungen müssen von einer dedizierten Schnittstelle abgerufen werden. (Schmitzhofer)
- Jede Veranstaltung enthält Informationen zu Ort, Zeit und verfügt über eine Bezeichnung. (Irimus & Schmitzhofer)
- Die Applikation wird bei Fertigstellung kostenlos freigegeben. (Irimus & Schmitzhofer)

### 1.2.2. Soll-Kriterien (FR)

- Veranstaltungen und Orte sollen getrennt gespeichert und per Orts-ID miteinander verknüpft werden. (Schmitzhofer)
- Veranstaltungen sollen ausgehend von der aktuellen Zeit, zeitlich sortiert angezeigt werden. (Irimus & Schmitzhofer)
- Veranstaltungen sollen relativ zur aktuellen Benutzer-GPS-Position örtlich gefiltert werden. (Irimus)
- Die vom Benutzer im Browser freigegebene GPS-Position soll für die Distanzermittlung herangezogen werden (Irimus)

### 1.2.3. Kann-Kriterien (FR)

- Veranstaltungen (bzw deren Ort) könnte zusätzlich innerhalb einer Kartenansicht dargestellt werden. (Irimus)

## Software Requirements Specification

Projektname

- Es könnten wiederholbare Ereignisse in der Datenbank hinterlegt werden (zB wöchentliche Ereignisse). (Schmitzhofer)
- Ein Benutzer kann den Radius der aktuell anzuzeigenden Events anpassen/ändern. (Irimus)
- Bei (vom Benutzer) nicht-erlaubter GPS-Freigabe, könnte die IP für Geolocation herangezogen und via (freier/registrierter) API abgerufen werden. (Irimus & Schmitzhofer)

### 1.2.4. Abgrenzungs-Kriterien (FR)

- Es wird vorerst kein Admin-Portal geben. Daten werden direkt in der Datenbank zur Verfügung gestellt.
- Es wird vorerst keine Funktion geben mit der Benutzer selbst Veranstaltungen hinzufügen können.

## 1.3. Nicht-Funktionale Anforderungen

(Zuständigkeiten: Nicht-Funktionale Anforderungen sind implizit von beiden Teammitgliedern einzuhalten.)

### 1.3.1. Muss-Kriterien (NFR)

- Jede Veranstaltung muss zumindest Informationen zu Ort, Datum und über eine Bezeichnung verfügen.
- Die Applikation muss bei Fertigstellung (initial) kostenlos freigegeben werden.

### 1.3.2. Soll-Kriterien (NFR)

- Orte sollen sowohl mit Adresse als auch GPS (Lon/Lat) festgehalten werden können (entweder & oder).
- Backend und Frontend sollen neu entwickelt werden (unabhängig von eingesetzter Technologie).

### 1.3.3. Kann-Kriterien (NFR)

- Das Backend kann mit spezialisierter Sprache (z.B. Rust oder Go) oder auch via moderner Backend Abstraktionstools (z.B. supabase) entwickelt werden.
- Das Frontend kann mit einem JavaScript Framework oder Library (z.B. Svelt oder Vue.js, aufbauend auf einem gängigem Framework wie Node.js oder Deno) entwickelt werden.
- Die Schnittstelle zwischen Front- und Backend soll über eine gängige Abstraktionstechnologie verfügen (z.B. REST, GraphQL, oder gRPC).

### 1.3.4. Abgrenzungs-Kriterien (NFR)

- Es wird im Rahmen dieses Projektes noch nicht möglich sein mit Clients unterschiedlicher Plattformen auf die Funktionen zuzugreifen (nur eine Art von Client wird entwickelt; Web, oder Android, oder iOS)
- Die Applikation verfügt über keine kaufbaren Funktionen ("in-game purchase")

## 2. Allgemeine Übersicht

### 2.1. Beschreibung der Ausgangssituation (Ist-Zustand)

Die christlich-österreichische Community verfügt zum Zeitpunkt der Erstellung dieses Dokuments über keine zentrale Informationsquelle jeglicher christlicher Events unabhängiger Konfession.

### 2.2. Produkteinsatz

#### 2.2.1. Anwendungsbereiche und Zielgruppen

UGW verfügt über zwei verschiedene Anwendungsgebiete. Größter Anwendungsbereich sind Privatpersonen christlichen Glaubens in Österreich. Kleinerer, nicht irrelevanter Bereich schließt christliche Institutionen, Kirchen, Bands und andere Organisationsformen, welche unterschiedliche Events für die christliche Community bereitstellen, mit ein.

Im ersten Fall besteht die Zielgruppe hierbei aus jungen Menschen zwischen 18 und 30 welche ihren Platz in der christlichen Community suchen. Außerdem zielt UGW darauf, das Bewusstsein für die unterschiedlichen christlichen Konfessionen in Österreich zu schärfen und die Einheit unter diesen zu stärken. Im Bezug auf den zweiten Anwendungsbereich strebt UGW eine Vereinfachung sowie Verbesserung des Eventsharings jeglicher christlicher Institutionen an. Es soll als zentrale Anlaufstelle und ein kordinierbares Eventmanagementsystem für die gesamte christlich-österreichische Community dienen.

#### 2.2.2. Betriebsbedingungen

Für eine sinnvolle Verwendung der App durch den Endbenutzer, ist es erforderlich, dass bereits Daten/Events für das verwendete Gebiet eingepflegt wurden.

#### 2.2.3. Annahmen und Abhängigkeiten

UGW könnte mittelfristig als zentrale Eventmanagement-Applikation für Veranstalter und Event-Suchende werden. Es besteht somit eine Abhängigkeit von den Veranstaltern und Personen die Event-Informationen zur Verfügung stellen.

## 2.3. Produktumfeld

### 2.3.1. Systemschnittstelle

Als Systemschnittstelle wird eine Serveranwendung implementiert, die unter anderem den Zugriff zur Datenbank abstrahiert.

### 2.3.2. Benutzerschnittstelle

Im Rahmen dieses Projektes wird eine Client Anwendung konzipiert und implementiert, die es einem Benutzer ermöglicht die zuvor eingetragenen Veranstaltungen abzurufen und gefiltert und sortiert einzusehen.

### 2.3.3. Kommunikationsschnittstelle

Als Kommunikationsschnittstelle wird ein Datenbankabstraktions-Service implementiert (z.B. REST, GraphQL, etc.), welcher eine spezifizierte Schnittstelle zwischen Systemanwendung und Benutzeranwendung(en) zur Verfügung stellt.