

8. Spezifikation der Proof of Concepts

Es müssen mehrere PoCs entwickelt werden, die das Projekt auf ihre Machbarkeit prüft. Die Risiken liefern eine gute Basis für deren Spezifikation.

8.1. Client-/Serverkommunikation

Die Systemarchitektur setzt eine Client/Serverarchitektur voraus. Mittels Node.js soll ein Webserver aufgesetzt und eine Android-App implementiert werden.

Exit: Das Ziel ist erreicht, wenn zwischen Android und dem Webservice eine Verbindung aufgebaut ist. Ausserdem muss der Server in der Lage sein, auf GET und POST Methoden angemessen zu reagieren.

Fail: Das Proof of Concept ist gescheitert, wenn keine Kommunikation zwischen Clients und Server hergestellt werden konnte.

8.2. Überprüfung der Nutzungsrechte von Daten eines Fremdanbieters

Es werden Nutzungsrechte der Daten von Fremdanbietern überprüft, ob diese im System verwendet werden dürfen und manipuliert werden dürfen.

Exit: Wenn in den Nutzungsrechten dokumentiert ist, dass die Verwendung und Manipulation der Daten gestattet ist, dann ist das Ziel erreicht.

Fallback: Eine mögliche Alternative wäre bei bekannten Rezeptenanbietern über eine Kooperation anzufragen. Hier zu sollte genügend Zeit für die Kommunikation zwischen den möglichen Partnern eingeplant werden, da ein solches Vorhaben viel Besprechungsbedarf benötigt. Des Weiteren wäre eine communitybasierte Datennutzung sowie die Nutzung von kostenpflichtigen Daten eine weitere Möglichkeit.

8.3. Benutzung von APIs

Die Applikation soll auf eine 3rd Party-API Rezepte abrufen. Dieses PoC setzt ein positives Ergebnis aus Abschnitt 8.2 voraus.

Exit: Es existiert mindestens eine 3rd-Party-API, die einen Zugriff auf deren Datenbestand ermöglicht. Sie gibt Daten in einem brauchbaren Format, vorzugsweise JSON, zurück.

Fail: Wenn die API nicht ins System implementiert werden kann und somit nicht auf die Daten zugegriffen werden kann, so ist dieser Punkt des Proof of Concept fehlgeschlagen.

Fallback: Als eine Alternative kann das Anlegen einer eigenen Rezeptsammlung in Betracht gezogen werden.

8.4 Matchingalgorithmus

Der Matchingalgorithmus ist ein essentieller Bestandteil des Systems. Die ordnungsgemäße Implementation trägt zum Erfolg bei.

Exit: Das Ermitteln passender Rezepte läuft ohne Probleme. Das System ist in der Lage, mit den gesammelten Informationen zu den Zutaten Anfragen an Rezeptquellen durchzuführen. Nach weiteren Filterungen auf Basis der Benutzerprofile liegen am Ende sinnvolle Rezeptvorschläge vor.

Man nehme an, dass unter den Gästen ein Allergiker ist. Demnach darf das System nach dem Matchingvorgang keine Rezepte vorschlagen, dessen Verzehr allergische Reaktionen bei der besagten Person hervorrufen könnte.

Fail: Der Vorgang schlägt fehl oder liefert unangemessene Ergebnisse, weil das System nicht auf die Benutzerprofile zugreifen konnte oder ein Teilnehmer vergaß, Angaben zu seiner Ernährungsweise zu machen.

Fallback: Unangemessene Rezepte werden trotzdem vorgeschlagen, Benutzer muss selbst darauf achten.

8.5. Empfangen und Senden von Push Benachrichtigung über GCM (Google Cloud Messaging)

//beschreibung

Exit: Das Risiko gilt als beseitigt, wenn die Push-Benachrichtigungen erfolgreich an die Clients gesendet sind.

Fail: Das Senden von Push-Benachrichtigungen schlägt fehl.

Fallback: Auf andere Alternativen zurückgreifen , wie z.b Amazon AWS oder Pushy.

8.6. Fehlende Erfahrung in der App Entwicklung

Im Team sind keine Kenntnisse bzw. Erfahrungen in der Entwicklung einer Applikation für das Betriebssystem Android vorhanden. Da es für eine Client/Server Kommunikation mindestens ein Client implementiert werden muss, muss sich das Team in die Entwicklung von Android Applikationen einarbeiten.

Exit: Das Exit-Kriterium ist erreicht, wenn die Implementation der drei wichtigsten Proof of Concepts am

Fail: Dieses Kriterium ist fehlgeschlagen, wenn bis zum genannten Zeitpunkt keine oder geringe Teile des Systems des Systems implementiert sind.

Fallback: Anstatt einer Android Applikation wird eine Desktop Anwendung entwickelt, damit an dieser Anwendung die Funktionalitäten ersichtlich werden.