

# Dokumentation Veggi-Zöli-App



## Inhaltsverzeichnis

1. Testprotokoll .....	3
2. Reflexion.....	4
3. Vorbereitung zu Veröffentlichung .....	5
4. Installationsanleitung.....	6

# 1. Testprotokoll

Grün: Erfolgreich

Rot: Durchgefallen

Identifikation	Person	Vorbedingungen	Vorgehen	Erwartets Resultat	Ergebnis
«Rezept» Button	Emina Heinz	Man muss auf der Startseite sein.	Man muss den Button anklicken.	Liste aller Rezepte wird ersichtlich.	
«Burger» Button	Emina Heinz	Man muss auf der Rezept Seite sein.	Man muss den Button «Burgern» anklicken.	Man sieht die Zutaten und zusätzlich die Anleitung.	
«Pasta» Button	Emina Heinz	Man muss auf der Rezept Seite sein.	Man muss den Button «Pasta» anklicken.	Man sieht die Zutaten und zusätzlich die Anleitung.	
«Back» Button (Burger)	Emina Heinz	Man muss auf der Burger Seite sein.	Man muss den Button anklicken.	Die Rezepte Seite ist wieder ersichtlich.	
«Back» Button (Pasta)	Emina Heinz	Man muss auf der Pasta Seite sein.	Man muss den Button anklicken.	Die Rezepte Seite ist wieder ersichtlich.	
«Back» Button (Rezept)	Emina Heinz	Man muss auf der Rezept Seite sein.	Man muss den Button anklicken.	Die Startseite ist wieder ersichtlich.	
«News» Button	Emina Heinz	Man muss auf der Startseite sein.	Man muss den Button anklicken.	Liste aller News wird ersichtlich.	
«Back» Button (News)	Emina Heinz	Man muss auf der News Seite sein.	Man muss den Button anklicken.	Die Startseite ist wieder ersichtlich.	
«Restaurant» Button	Emina Heinz	Man muss auf der Startseite sein.	Man muss den Button anklicken.	Liste aller Restaurants wird ersichtlich.	
«Back» Button (Restaurants)	Emina Heinz	Man muss auf der Restaurants Seite sein.	Man muss den Button anklicken.	Die Startseite ist wieder ersichtlich.	
«Fakten» Button	Emina Heinz	Man muss auf der Startseite sein.	Man muss den Button anklicken.	Liste aller Fakten wird ersichtlich.	
«Back» Button (Fakten)	Emina Heinz	Man muss auf der Fakten Seite sein.	Man muss den Button anklicken.	Die Startseite ist wieder ersichtlich.	

## 2. Reflexion

Ich konnte alle meine Funktionalen und nicht Funktionalen Anforderungen erfüllen, auch diese die Optional waren. Ich hatte ganz am Anfang noch drin das ich eine Searchbar machen wollte dies habe ich dann aber ganz weggelassen, dass es keinen Zweck dafür gab und ich es nun ohne fast schöner fand. Dies ist auch der Grund, wieso diese Funktion auf meinem alten Activity Diagramm drauf ist.

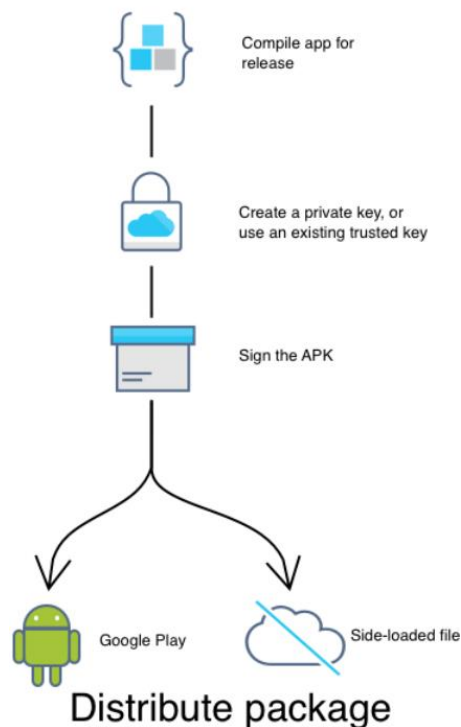
Ich würde mich nächstes Mal definitiv nochmals für maui .net entscheiden. Ich bin mit dieser Technologie sehr gut klargekommen, der grösste Vorteil war das ich mich mit C# schon auskannte und deshalb schon einige Vorkenntnisse hatte. Es gibt sehr viele Dokumentationen über spezielle Themen, die mir auch weitergeholfen haben.

Ich hatte eine simple Idee ausgewählt, aber ich denke das war nicht schlecht so konnte ich alles fertig machen das ich mir vorgenommen habe. Die App passt auch sehr gut zu mir und ich musste für deren Inhalt fast nicht Googlen da ich diese Infos schon alle im Voraus kannte.

Falls ich die App wirklich veröffentlichen würde, dann würde ich die Daten natürlich nicht aus einem json File auslesen, sondern aus einer Datenbank.

### 3. Vorbereitung zu Veröffentlichung

#### Prepare for deployment



Um eine .NET Maui-Android-App (.NET Multi-Platform App UI) zu verteilen, müssen Sie sie vor dem Hochladen in Google Play mit einem Schlüssel aus Ihrem Keystore signieren. Keystores sind Binärdateien, die als Repositories von Zertifikaten und privaten Schlüsseln dienen.

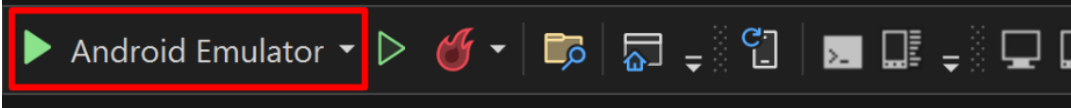
Google Play erfordert, dass Sie Ihre App als Android App Bundle (AAB) übermitteln. Google Play verwendet Ihr App-Bündel, um optimierte Android-Pakete (APK) für jede Gerätekonfiguration zu generieren und bereitzustellen, sodass nur der Code und die Ressourcen heruntergeladen werden, die für ein bestimmtes Gerät zum Ausführen Ihrer App benötigt werden.

Der Prozess zum Verteilen einer .NET MAUI Android-App über Google Play sieht wie folgt aus:

1. Erstellen Sie ein Google Play-Entwicklerkonto.
2. Erstellen Sie Ihre App in der Google Play-Konsole.
3. Richten Sie Ihre App in der Google Play-Konsole ein.
4. Stellen Sie sicher, dass Ihre App das richtige Paketformat verwendet
5. Erstellen und signieren Sie Ihre App in Visual Studio, und verteilen Sie sie dann über die Google Play Console.
6. Anschließend können nachfolgende Versionen Ihrer App über Visual Studio veröffentlicht werden.

Weitere Details zu jedem Punkt findet man unter: <https://learn.microsoft.com/de-de/dotnet/maui/android/deployment/publish-google-play?source=recommendations#create-a-google-play-developer-account>

## 4. Installationsanleitung

1. Visual Studio Installieren: <https://visualstudio.microsoft.com/de/downloads/>
2. Nach Installation Visual Studio Installer öffnen:
  - a. «Ändern» klicken
  - b. Desktop- und Mobilgeräte
  - c. «.NET Multi-Plattform App UI-Entwicklung» auswählen
  - d. Unten rechts «Ändern» klicken
3. Neuer Ordner erstellen in einem Verzeichnis in dem **NUR ASCII ZEICHEN** sind heisst keine Umlaute oder anderes.
4. Terminal öffnen und Befehl ausführen:
  - a. `git clone https://github.com/eminaheinz/m335.git`
5. In Verzeichnis «. /m335/src/MobileApp/» bewegen
6. Doppelklick auf MobileApp.sln
7. «Windows Machine» drop-down menu öffnen
8. Android Emulators -> Android Emulator auswählen
9. 
10. Nun kommt die Android SDK – License Agreement diese müssen alles akzeptiert werden
11. Nun Punkt 9. Wiederholen
12. Yes -> Create
13. Nun muss unter Einstellungen -> Datenschutz und Sicherheit -> Für Entwickler -> Entwicklermodus Ein
14. Nun in der Tool Bar Projekt -> NuGet-Packete verwalten
15. Newtonsoft.Json installieren
16. Maui Controls installieren
17. Wählen sie nun das erstellte Handy aus, um die Applikation zu starten.