

# **Dokumentation der Android App EasyTalkNew**

erstellt von

Michel Weidemann

Matrikelnummer: 10217061

## 1. Ziel der App

Ziel der Android App war es eine unabhängig Plattform zum Unterhalten mittels Textnachrichten zu erstellen. Unabhängig bedeutet das keine externen Funktionen von Drittanbietern genutzt werden soll. Dazu zählen die von Google zur Verfügung gestellten Dienste wie zum Beispiel Firebase. Auch sollte die App keine Pushnachrichten an das mobile Endgerät senden, wie es bei dem Google Cloud Messaging(GCM) der Fall ist.

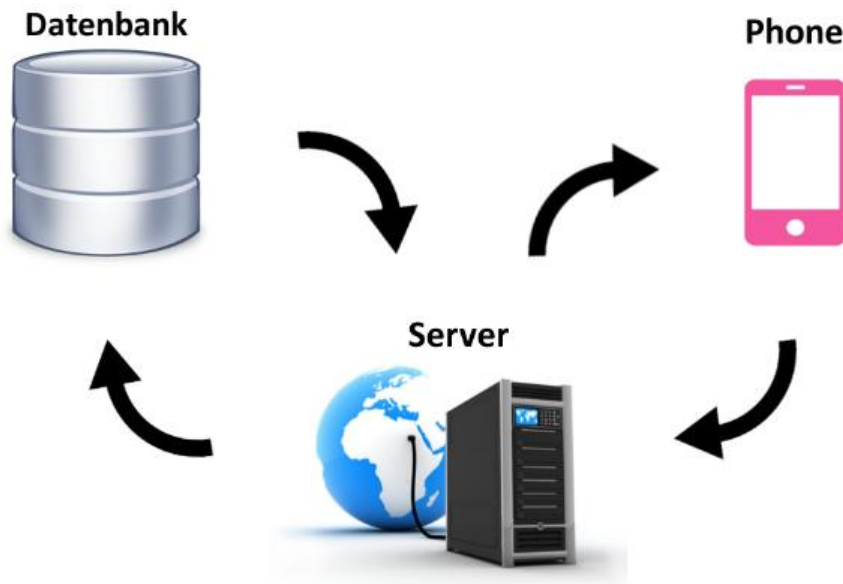
Des Weiteren sollten auch Geräte ohne Sim-Karte bzw. Telefonnummer funktionieren, dazu wurde eine Registration und ein Login mittels Web-Server über eine Datenbank verwendet. Der Nachrichtenaustausch erfolgt auch über den Web-Server und der Datenbank.

## 2. Funktionsweise der einzelnen Appfunktionen

In diesen Abschnitt der Dokumentation wird grob auf die einzelnen Elemente der App eingegangen.

Bei der ersten Funktion handelt es sich um das Registrieren eines neuen Nutzers. Dabei muss der Anwender die Registrationsaktivität aufrufen mittels des Listeners auf dem Element("Register Here"). Der Anwender wird daraufhin auf die neue Aktivität geleitet. Dort werden die Daten mittels Eingabefelder erfasst. Nachdem der Anwender auf den Register-Button drückt sendet die App einen Request an den Web-Server. Der Request beinhaltet die URL, welche zum Aufrufen des richtigen PHP-Scripts dient und zusätzlich noch den Inhalt der zwei Variablen, welche per POST an den Script übergeben werden. Der Web-Server führt dann den aufgerufenen Script aus und sendet eine Anfrage an die Datenbank, außerdem wird vorher geprüft ob der Nutzer vorhanden ist. Zusätzlich wird das Passwort mittels eines vorgefertigten PHP-Scripts verschlüsselt. Wenn der Nutzer schon vorhanden ist erhält die App ein JSON-Objekt mit einer Negativantwort, welche dann mit einer Fehlermeldung dem Anwender angezeigt wird. Sobald der Nutzer noch nicht

vorhanden ist schreibt der PHP-Script die Daten in die Datenbank. Der Web-Server sendet eine positive Antwort zu der App zurück und der Anwender wird wieder an die Login-Aktivität verwiesen.(Abb.1)



*Abb.1: Datenaustausch der App, Web-Server und Datenbank*

Die Login-Funktion läuft ähnlich wie die Registrierung ab. Der Anwender gibt seine Nutzerdaten ein, die App sendet daraufhin eine Anfrage an den Web-Server, dieser wiederum führt den angefragten PHP-Script aus, der auf die Datenbank zugreift. Der Web-Server sendet dann eine Antwort an die App in einem JSON-Objekt. Diese Antwort wird von der App ausgewertet. Bei einer positiven Antwort gelangt der Nutzer in die Chat-Aktivität und bei einer negativen Antwort verbleibt dieser im Login.

Nachdem der Anwender sich eingeloggt hat befindet er sich in der Hauptaktivität der App der Chat-Aktivität.

Die Chat Aktivität besitzt zwei wichtige Funktionen. Dem Senden einer Nachricht und das Empfangen von Nachrichten. Das Versenden von Nachrichten erfolgt mittels eines Buttons(Send-Button). Sobald der Nutzer eine Nachricht eingegeben hat und diese versendet schickt die App einen Request an den Web-Server. Dieser ruft den angefragten PHP-Script auf und kommuniziert daraufhin mit der Datenbank. Dabei wird die Nachricht in die Nachrichten Tabelle eingetragen. Dieser Eintrag besitzt neben der eigentlichen Nachrichten noch zwei IDs, die Nachrichten-ID und die

Nutzer-ID(Wer hat die Nachricht gesendet?), und einem Zeitstempel. Die beiden IDs werden beim Empfangen von Nachrichten eine wichtige Rolle spielen.

Neben der Funktion des Sendens wird alle zehn Sekunden eine Anfrage zum Server gesendet ob eine neue Nachricht vorhanden ist. Dies erfolgt auch über einen Request und den Aufruf eines PHP-Scripts. Dabei fragt der PHP-Script die Datenbank ab, welche Nachricht ist nicht vom angemeldeten Nutzer, dies wird mittels der Nutzer-ID erfragt. Die darauf ermittelte Nachricht wird wieder an die App per JSON-Objekt gesendet. Die App prüft dann ob diese Nachricht schon einmal angezeigt wurde.

### 3. Aufbau der Datenbank

Die Datenbank besteht aus drei Tabellen. Dabei werden aktuell nur zwei Tabellen von der App verwendet. Die erste Tabelle(user) speichert alle registrierten Nutzer, diese Tabelle ist über die User\_ID mit der Tabelle(messages), welche die Nachrichten speichert, verbunden. Die Nachricht-Tabelle besteht damit erkannt werden kann, welche Nachrichten müssen wenn zugestellt werden.

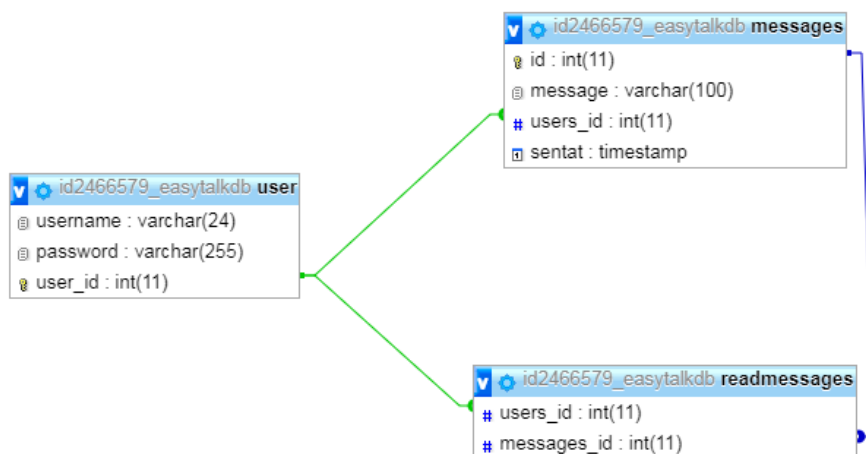


Abb.2: Datenbank EasyTalkDB

## **4. Geplante Funktionen**

Aufgrund der Tatsache das dieses Projekt zu Beginn für zwei Person geplant war, jedoch eine Person ausgeschieden ist, fallen einige Funktionen weg. Diese Funktionen können im nachhinein in das Projekt eingefügt werden.

Zu den geplanten Funktionen zählte das Erstellen von mehreren Chaträumen und das Einladen von Leuten in diese. Auch müssen die Nachrichten noch verschlüsselt werden.

Bestehende Funktionen könnten noch ausgebaut beziehungsweise verbessert werden. Ein Beispiel wäre das Einbinden der dritten Tabelle in der Datenbank und das damit verbundene Empfangen von Nachrichten.

## **5. Installation**

Als Erstes wird die APK-Datei auf den mobilen Endgerät oder virtuellen Endgerät installiert werden. Weiterhin wird ein Web-Server benötigt, dieser kann auf einen normalen Computer mittels XAMPP realisiert werden oder per Webhost-Anbieter. Dabei sollte darauf geachtet werden dass dieser PHP-Scripte ausführen und eine Datenbank bereitstellen kann. Auf dem Web-Server müssen alle PHP-Scripte hinterlegt und die Datenbank eingerichtet werden. Die Datenbank muss eine MySQL-Datenbank sein.

## **6. Programm-Version**

Folgende Programme wurden verwendet:

- Android Studio Version 2.3.3
- PHP-Script Version 7.0.8
- Apache Version 2.4.6
- MySQL Version 5.0.12-dev