

# Studnetz, eine Mobilapplikation zur Vermittlung von Nachhilfeunterricht

Florian Hirtz

27. August 2018

---

## **Vorwort**

Diese Arbeit befasst sich mit der Entwicklung einer Applikation für Mobiltelefone. Die Applikation soll die Vermittlung von Nachhilfeunterricht unter Schülerinnen und Schülern einer Schule vereinfachen. Dabei ist anzumerken, dass die Entwicklung einer vollständig markttauglichen Applikation den Rahmen dieser Arbeit sprengen würde. Das Ziel ist vielmehr das Legen eines Grundsteins, aus welchem ein solches Produkt entstehen könnte.

### **Zu mir**

Ich möchte an dieser Stelle die Gelegenheit ergreifen, ein paar Worte zu mir, dem Autor und Entwickler dieser Arbeit, zu verlieren.

Zum Zeitpunkt, an welchem ich begann die Studnetz-Applikation zu entwickeln, hatte ich zuvor noch nie eine Applikation für Mobiltelefone entwickelt. Bis zu diesem Zeitpunkt habe ich mich hauptsächlich auf die Entwicklung kleinerer Programme für Computer beschränkt. Das dazu nötige Wissen habe ich mir grösstenteils selber angeeignet. So hatte ich mich vor diesem Projekt noch nie mit Datenbanken und Webdevelopment befasst. Ich habe mir dieses Projekt als Herausforderung genommen, um etwas Neues zu lernen und mich als Entwickler weiter zu entwickeln. Dieser Lernprozess hat definitiv stattgefunden. Dies erklärt auch, weshalb teilweise gleiche Probleme an verschiedenen Orten verschieden gelöst wurden oder warum gewisse verwendete Lösungen von anerkannten Praktiken abweichen.

### **Motivation**

Die Motivation für die Entwicklung der Studnetz-Applikation schöpfte ich hauptsächlich aus zwei Quellen. Zum einen war es der Wille, neue Teilgebiete des Programmierens kennenzulernen und zu verstehen, weshalb die Wahl dann sehr schnell auf eine Applikation für Mobiltelefone fiel. Immerhin sind Mobiltelefone in der heutigen Welt kaum mehr wegzudenken und die Fähigkeit, Programme für sie zu entwickeln hat mich schon lange fasziniert. Die zweite Motivation ist dann aus einer Idee von Herrn Bättig entstanden, der während der einführenden Präsentation in die Maturarbeit eine Applikation zur Vermittlung für Nachhilfe erwähnte. Zuerst bin ich skeptisch gewesen, doch schnell habe ich im Gespräch mit anderen festgestellt, dass eine gut entwickelte Applikation in diesem Bereich durchaus Verwendung und Beliebtheit finden könnte. Dies ist dann der Anstoss gewesen, ein solches Projekt in Angriff zu nehmen.

---

## **Danksagungen**

Ich möchte mich hier bei den Personen bedanken, die mich während dieser Arbeit unterstützt haben. Zuerst einmal möchte ich meiner betreuten Lehrperson Andreas Umbach danken. Er stellte mir nicht nur einen Platz auf der Datenbank des Ergänzungsfaches zur Verfügung, sondern half mir stets bei allfälligen Fragen und Unklarheiten weiter. Zudem möchte ich ein Dankeschön an meinen Onkel Peter Arrenbrecht richten, der mir mit seinem Hinweis auf die Firebase API komplett neue Möglichkeiten eröffnet hat. Als letztes gilt noch ein gigantisches Dankeschön an meinen Vater, der mich während dieser ganzen Arbeit als Ratgeber und Korrekturleser immer unterstützte und immer ein offenes Ohr für meine Probleme zeigte.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>4</b>
1.1	Zielsetzung . . . . .	4
<b>2</b>	<b>Konzeptionelle Grundlagen</b>	<b>5</b>
2.1	Client-Server Prinzip . . . . .	5
2.2	Das Modell-View-Presenter Konzept (Passive View) . . . . .	6
<b>3</b>	<b>Die entwickelte Applikation</b>	<b>7</b>
3.1	Features . . . . .	7
3.2	Die entwickelte Applikation . . . . .	8

# Kapitel 1

## Einleitung

### 1.1 Zielsetzung

Das Ziel dieser Arbeit ist das Entwickeln einer Applikation für Android Geräte, die die Vermittlung von Nachhilfeunterricht unter Schülerinnen und Schülern vereinfachen soll. Die Benutzerinnen und Benutzer der Applikation sollen in der Lage sein, sich einen Account innerhalb der Applikation zu erstellen und sich darin einzuloggen. Sie sollen angeben können, in welchen Fächern sie in der Lage sind, anderen Nachhilfe zu geben. Weiter soll eine Funktion vorhanden sein, mit welcher nach anderen Benutzern gesucht werden kann, die in den gewünschten Fächern Nachhilfe anbieten. So sollen Benutzerinnen und Benutzer bei Bedarf gezielt nach Nachhilfelehrern/Nachhilfelehrerinnen suchen können, die ihren Bedürfnissen entsprechen. Weiter soll es ihnen möglich sein, über eine Chatfunktion andere Benutzer zu kontaktieren, wo dann genauere Informationen wie Zeit, Ort und eventuell Preis ausgetauscht werden können.

## Kapitel 2

# Konzeptionelle Grundlagen

Im folgenden Kapitel werden die für diese Arbeit wichtigen Grundkonzepte etwas genauer beschrieben.

### 2.1 Client-Server Prinzip

Das *Client-Server Prinzip* ist ein weit verbreitetes Konzept, um die Aufgaben innerhalb eines Netzwerkes effizient aufzuteilen. Dabei werden die Aufgaben auf zwei im Netzwerk agierende Programme aufgeteilt. Diese Programme werden im Allgemeinen als Client und Server bezeichnet.

Der *Server* hat die Aufgabe, verschiedenste Dienste zur Verfügung zu stellen, welche auf Anfrage ausgeführt werden können. Ein solcher Dienst kann zum Beispiel das Versenden einer Nachricht oder das Aufrufen einer Webseite sein. Der Server selbst ist meist passiv, was soviel bedeute wie, dass der Server einzig auf Anfragen reagiert und nicht selber Anfragen stellt. Ein Server sollte immer in der Lage sein, Anfragen entgegenzunehmen und zu verarbeiten.

Der *Client* selber ist die aktive Komponente des Systems und ist meist die Komponente, mit welcher ein Benutzer/eine Benutzerin direkt interagiert. Der Client ist in der Lage, Anfragen an den Server zu stellen und von dessen Diensten Gebrauch zu machen. Grundsätzlich gibt es in einem solchen Netzwerk nur einen Server, jedoch kann es durchaus mehrere Clients geben. Ein guter Server sollte also auch darauf vorbereitet sein, mehrere Anfragen von verschiedenen Clients parallel zu bearbeiten. [3]

## 2.2 Das Modell-View-Presenter Konzept (Passive View)

Das *Modell-View-Presenter* (MVP) Konzept wie auch das sehr ähnliche *Modell-View-Controller* (MVC) Konzept, sind beide für das Entwickeln von Software entworfen worden. Ihre Idee ist es, die Aufgabenbereiche innerhalb einer Applikation strikt voneinander zu trennen. Dabei wird zwischen drei Typen von Aufgabenbereichen unterschieden:

- Die *View* ist die sichtbare Benutzeroberfläche. Sie hat die Aufgabe, dem Benutzer ein bedienbares Interface zu bieten und soll auf Anfrage den Status seiner einzelnen Komponenten weitergeben. Sie kennt das Model nicht.
- Das *Model* ist der Datenspeicher einer Applikation, der gebraucht wird um die View korrekt darzustellen. Es soll auf Anfrage hin Daten ausgeben können. In diesem Falle kennt das Model weder die View noch den Presenter. Je nach Auslegung des Konzepts hat das Model jedoch auch die Aufgabe, falls sich Datensätze ändern, den Presenter davon zu unterrichten. Das Model kennt in diesem Falle zwar das Model, jedoch die View nicht.
- Der *Presenter* oder *Controller* ist sozusagen der Mittelsmann der beiden anderen Komponenten. Er ist in der Lage Daten aus dem Model anzufordern und kontrolliert anschliessend was mit diesen Daten geschieht. Er hat ebenfalls die Möglichkeit, die angezeigte View zu ändern und deren Status abzufragen. Der Presenter ist in der Lage, sowohl die View als auch das Model zu manipulieren.

Wichtig bei dem MVP Konzept mit einem passiven View ist es, dass nur der Presenter die Möglichkeit hat, auf die beiden anderen Komponenten zuzugreifen. Der View und das Model sollen unter keinen Umständen direkt miteinander kommunizieren. Sämtlicher benötigter Informationsaustausch soll stets vom Presenter kontrolliert werden.[2] Ein grosser Vorteil einer nach diesen Regeln entwickelter Applikation ist, dass die einzelnen Komponenten weitgehend unabhängig von einander sind. Somit kann zum Beispiel der View komplett neu gestaltet werden, ohne dass der Presenter oder das Model geändert werden müssen, damit die Applikation weiterhin funktioniert.

## Kapitel 3

# Die entwickelte Applikation

Die im Rahmen dieser Arbeit entwickelte Applikation ist für Mobiltelefone mit einer Version des Betriebssystems Android mit einem API Level von 17 und darüber entwickelt worden [1]. Sie kann darauf installiert und anschließend ausgeführt werden. Für die Verwendung der Applikation ist eine Verbindung zum Internet notwendig.

### 3.1 Features

Damit die Applikation auch ihren Zweck erfüllen kann, besitzt sie eine Reihe von Features von welchen Benutzer/Benutzerinnen Gebrauch machen können.

- Das *Account-Feature*: Die Applikation bietet den Benutzern/ Benutzerinnen die Möglichkeit, sich einen persönlichen Account zu erstellen und ihn zu personalisieren. Es ist ihnen möglich, auf ihrem Profil ihren Namen, ihre besuchte Schule und ihr Geburtsjahr anzugeben. Weiter können sie auch eine kurze Beschreibung von sich selber verfassen und sie haben die Möglichkeit, falls sie selber Nachhilfe anbieten wollen, Fächer auszuwählen, in welchen sie das tun möchten. Das Profil selber ist für andere Benutzer/Benutzerinnen einsehbar. Die meisten Accountdetails können innerhalb der Applikation auch noch nach der Registrierung bearbeitet werden.
- Das *Such-Feature*: Es ist registrierten Benutzern/ Benutzerinnen möglich, über eine Suchfunktion nach anderen registrierten Benutzern/ Benutzerinnen zu suchen. Dabei kann nach dem Namen und nach ausgewählten Fächern gesucht werden. Die gefundenen Profile der ver-



schiedenen Benutzern/ Benutzerinnen können anschliessend angesehen werden.

- Das *Chat-Feature*: Sollte der Benutzer/die Benutzerin einen anderen Benutzer oder Benutzerin gefunden haben, mit welchem/welcher er/sie Kontakt aufnehmen möchte, so kann ein neuer Chat via das gefundene Profil geöffnet werden. Darin können Benutzer/ Benutzerinnen in Echtzeit kurze Textnachrichten miteinander austauschen. Sollte ein Chat geöffnet sein, kann sowohl der Sender/die Senderin wie auch der Empfänger/die Empfängerin ganz einfach von ihrem eigenen Profil auf ihn zugreifen.

### 3.2 Die entwickelte Applikation

Wenn die Applikation auf dem Mobilgerät ausgeführt wird, öffnet sich der Login-Screen (siehe Abbildung ??). Der Benutzer/Die Benutzerin kann sich nun hier, sollte er/sie bereits einen Account besitzen, mit seinem/ihrer Benutzernamen und Passwort einloggen. Sollte die Person noch keinen Account besitzen, kann sie über einen Schriftzug unterhalb der Logindaten zu einem Registrierungsformular gelangen (siehe Abbildung ??). Im Registrierungsformular wird nach sämtlichen Daten gefragt, die benötigt werden, um einen Account zu erstellen. Dazu gehört ein Benutzername, Vor- und Nachname, eine E-Mail Adresse, ein Passwort, ein Geburtsdatum sowie die momentan besuchte Schule. Ist die Registrierung erfolgreich, kann der Benutzer/die Benutzerin ab sofort sich in seinen/ihren neuen Account einloggen. Nach dem erfolgreichem Einloggen öffnet sich die Hauptseite der Applikation. Sie besteht hauptsächlich aus dem Profil des Benutzers/der Benutzerin (siehe Abbildung ??). Standardmässig findet sich hier ein Profilbild, der Name des Benutzers/der Benutzerin, sein/ihr Geburtsjahr und die besuchte Schule. Diese und noch weitere Angaben wie angebotene Fächer, eine Beschreibung oder ein persönliches Profilbild können in den Einstellungen jederzeit konfiguriert werden, welche über eine Schaltfläche in der oberen rechten Ecke erreicht werden können. Das Profilbild kann dabei entweder direkt mit der Kamera aufgenommen oder aus der Galerie ausgewählt werden. Anschliessend kann der Benutzer/die Benutzerin einen gewünschten ausschnitt im Bild definieren, auf welchen das Bild dann zugeschnitten wird (siehe Abbildung ?? und ??). Die ausgewählten Fächer werden in Form von Emblems auf dem Profil dargestellt. Ebenfalls auf der Hauptseite findet sich ein ausfahrbares *BottomSheet* am unteren Ende des Bildschirms. Es kann über eine runde Schaltfläche in der Mitte oder ein einfaches Streichen nach oben

ausgefahren werden. Im BottomSheet selber finden sich Einträge für alle offenen Chats des Benutzers/der Benutzerin(siehe Abbildung ??). Über einen einfachen Klick können sie geöffnet werden (siehe Abbildung ??). Das BottomSheet ist auch der Ort, an welchem ein Benutzer/eine Benutzerin sehen kann, ob er/sie neue Nachrichten empfangen hat. Neben dem BottomSheet findet sich ebenfalls im rechten unteren Ende eine runde Schaltfläche mit einer Lupe. Über sie kann auf das Such-Feature zugegriffen werden, welche den Benutzer/die Benutzerin auf einen neuen Screen führt. Dort können die Kriterien für die Suche definiert werden (siehe Abbildung ??). Hier können Suchparameter wie Name und Fächer angegeben werden, nach welchen gesucht werden soll. Ist der Benutzer/die Benutzerin zufrieden mit den Einstellungen kann über eine Schaltfläche die Suche ausgeführt werden. Alle Suchergebnisse werden in Form einer Liste daraufhin im Client aufgeführt (siehe Abbildung ??). Sollte den Benutzer/die Benutzerin Interesse an einem der Ergebnisse haben, kann durch einfaches Auswählen des Elementes in der Liste das Profil des/der gefundenen Benutzer/Benutzerin aufgerufen werden (siehe Abbildung ??). Ist der Benutzer/die Benutzerin interessiert, Kontakt mit dem/der gefundenen Benutzer/Benutzerin aufzunehmen, kann nun über eine Schaltfläche auf dem Profil ein neuer Chat geöffnet werden (siehe Abbildung ??). Dort können dann mithilfe kurzer Textnachrichten genauere Informationen über eine mögliche Zusammenarbeit ausgetauscht werden. Die Überlieferung erfolgt dabei in Echtzeit.

# Literaturverzeichnis

- [1] Android Developers. *juses-sdk¿ (What is API Level?)*.  
[https://www.datenschutz-notizen.de/  
die-schweiz-und-die-datenschutzgrundverordnung-0020134/](https://www.datenschutz-notizen.de/die-schweiz-und-die-datenschutzgrundverordnung-0020134/),  
2018. Abgerufen am: 27.08.2018.
- [2] U. Author. *Presentation Patterns : MVC, MVP, PM, MVVM*.  
[https://manojjaggavarapu.wordpress.com/2012/05/02/  
presentation-patterns-mvc-mvp-pm-mvvm/](https://manojjaggavarapu.wordpress.com/2012/05/02/presentation-patterns-mvc-mvp-pm-mvvm/), 2012. Abgerufen am:  
10.05.18.
- [3] fachadmin.de. *Server-Client Prinzip*. [https://www.fachadmin.de/  
index.php?title=Client-Server\\_Prinzip&oldid=3425](https://www.fachadmin.de/index.php?title=Client-Server_Prinzip&oldid=3425), 2011.  
Abgerufen am: 01.05.18.