

Stephan Moser

iPhone App Additions- und Subtraktionstrainer

Bachelor Arbeit

Technische Universität Graz

Institut für Informationssysteme und Computer Medien

Leiter: Univ.-Prof. Dipl.-Ing. Dr.techn. Frank Kappe

Betreuer: Univ.-Doz. Dipl.-Ing. Dr.techn. Martin Ebner

Graz, Oktober 2013

This document is set in Palatino, compiled with pdf \LaTeX 2e and Biber.

The \LaTeX template from Karl Voit is based on KOMA script and can be found online: <https://github.com/novoid/LaTeX-KOMA-template>

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Graz, _____
Date

Signature

Eidesstattliche Erklärung¹

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am _____
Datum

Unterschrift

¹Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008; Genehmigung des Senates am 1.12.2008

Kurzfassung

Im Zuge meiner Bachelorarbeit, wurde eine native iPhone App als Ergänzung zu Benedikt Neuholds Additions- und Subtraktionstrainer entwickelt. Der Funktionsumfang besteht grundsätzlich aus zwei Teilen:

Als Primärfunktion wurde ein Online-Trainer entwickelt, der per Webservice abfragt ob ein User Zugriff auf das System hat oder nicht, und der bei erfolgreicher Anmeldung beim Webservice, die für den User bestimmten Rechenaufgaben übermittelt bekommt. Diese Rechenaufgaben werden durch die App in grafisch ansprechender Weise präsentiert, und der/die BenutzerIn hat die Möglichkeit das Ergebnis einzugeben. Für die Auswertung der Rechenaufgaben werden die Ergebnisse, und auch alle Zwischenergebnisse in Form von Überträgen, mitgeloggt und nach Abschluss des Rechendurchlaufes wieder an das Webservice übermittelt wo das Ergebnis und der Lernfortschritt gespeichert wird.

Die Sekundärfunktion der App ist eine Offline-Übungsmöglichkeit, die in unterschiedlichen Schwierigkeitsstufen unauthorisiert/anonym durchführbar ist, und dem Zwecke der Verbesserung der Rechenfähigkeiten des Benutzers dient.

Abstract

Over the course of my bachelor's thesis, an iPhone App has been developed as a supplement to Benedikt Neuhold's Summation -and Subtraction Trainer. The range of usage possibilities of the App consists basically of two functions:

As a primary function, an Online-Trainer has been developed, asking Neuhold's System for access. If the user's authentication was successful, arithmetical problems are being submitted. These arithmetical problems are presented in an appealing graphical manner, with the possibility for the user to enter the result. For the purpose of the evaluation of the user's performance, the results and all intermediate results such as carries are being logged. After every computation iteration the results are being sent to Neuhold's System again, where they are being stored, together with the user's learning progress.

The secondary function of the App is an offline exercise, with the possibility to train the user's skills anonymously and in different difficulty levels.

Inhaltsverzeichnis

Abstract	iv
1 Einleitung	1
1.1 Gliederung der Arbeit	2
2 Stand der Technik	3
2.1 MathBoard ²	3
2.2 Addition Master: Mathematik Spiel ³	3
2.3 Addition ! ⁴ and Subtraction ! ⁵	4
2.4 Weitere Apps	7
3 Umsetzung	11
3.1 Hauptmenü	11
3.2 Trainer	12
3.2.1 Kommunikationsablauf mit Webservice	17
3.3 Üben	22
3.4 Einstellungen	27
3.5 Hilfe	28
4 Diskussion	30
5 Zusammenfassung und Ausblick	32
Literatur	35

²<https://itunes.apple.com/de/app/mathboard/id373909837?mt=8>.

³<https://itunes.apple.com/de/app/addition-master-mathematik/id672669932?mt=8>.

⁴<https://itunes.apple.com/de/app/addition-!/id447548669?mt=8>.

⁵<https://itunes.apple.com/de/app/subtraction-!/id447548515?mt=8>.

Abbildungsverzeichnis

2.1	Screenshot Mathboard	4
2.2	Screenshot Addition Master	5
2.3	Screenshot Addition !	6
2.4	Screenshot Subtraction !	6
2.5	Screenshot Add & Sub	7
2.6	Screenshot Add & Sub with Springbird	8
2.7	Screenshot Add & Sub For Kids	8
2.8	Screenshot Add Sub K-1	9
2.9	Screenshot Addition - Subtraction	10
2.10	Screenshot Subtraction with Fun	10
3.1	Screenshot Hauptmenü.	12
3.2	Screenshot Login.	13
3.3	Screenshot Logout.	16
3.4	Screenshot Trainer.	17
3.5	Screenshot Feedback.	18
3.6	Kommunikation Webservice	19
3.7	Screenshot Schwierigkeitsgrad.	23
3.8	Screenshot Auswertung.	27
3.9	Screenshot Einstellungen.	28
3.10	Screenshot Hilfe.	29
4.1	Statistik über die Leistung des/der SchülerIn	31

Listings

3.1	SOAP XML Nachricht zur Benutzer Authentifizierung	14
3.2	Objective-C Code zur Erstellung des Hashs	14
3.3	Anfordern einer Rechenaufgabe	18
3.4	Erhalten einer Rechenaufgabe	20
3.5	Übermitteln des Resultats	21
3.6	Antwort auf Übermittlung des Ergebnisses	22
3.7	Generierung von Rechenaufgaben im Übungsmodus	24

1 Einleitung

Zweifelsfrei hat das Aufkommen Mobiler Technologien und Smartphones mittlerweile große Auswirkungen auf unser tägliches Leben. Dazu gehört auch die Art wie wir heutzutage Lernen. Um dieser Tatsache Rechnung zu tragen wurde im Zuge dieser Arbeit eine iPhone Application (kurz App genannt) für Apple's Smartphone Betriebssystem iOS entwickelt, die Benedikt Neuhold bei seiner Diplomarbeit »Adaptives Informationssystem zur Erlernung mehrstelliger Addition und Subtraktion«¹ unterstützen soll.

Konkret geht es darum, dass es durch diese App für SchülerInnen unkompliziert und schnell möglich sein soll Additionen und Subtraktionen zu üben. Dazu melden sich die SchülerInnen über die iPhone App bei Neuhold's System an, und bekommen daraufhin auf ihre Bedürfnisse angepasste Rechenübungen die ihrem derzeitigen Wissensstand entsprechen. Der eigentliche Zweck der App besteht aber darin, dass die Ergebnisse und auch alle Zwischenergebnisse in Form von Überträgen protokolliert werden und in weiterer Folge an das bereits erwähnte System von Benedikt Neuhold zur Analyse weitergeleitet werden.

Da in erster Linie Kinder im Volksschulalter die Adressaten für Addition- und Subtraktionsübungen sind liegt ein wesentlicher Teil der Arbeit darin, die App so einfach wie möglich und dabei grafisch ansprechend zu gestalten, um die langfristige Motivation der SchülerInnen sicherzustellen.

Im folgenden Abschnitt 1.1 wird ein kurzer Überblick auf diese schriftliche Arbeit gegeben.

¹Neuhold, 2013.

1.1 Gliederung der Arbeit

In Kapitel 2 wird kurz darauf eingegangen, welche Arbeiten es zu diesem Thema bereits gibt, und in welcher Form sich diese von der hier diskutierten Arbeit unterscheiden.

Kapitel 3 handelt von der technischen Umsetzung der App, das heißt es wird beschrieben welche Technologien zur Umsetzung der Arbeit verwendet wurden und wie diese im Kontext dieser App angepasst und verwendet wurden.

Gewonnene Ergebnisse sowie aufgetretene Probleme im Vorfeld der Arbeit, während der Umsetzung aber auch in der Nachbereitung werden in Kapitel 4 diskutiert.

Das vorletzte Kapitel 5 fasst die gesamte Arbeit mit all den gewonnenen Erkenntnissen noch einmal zusammen bevor in Kapitel ?? ein Ausblick gewagt wird in welche Richtung sich das Thema des Mobilen Lernens hinentwickeln wird.

2 Stand der Technik

In diesem Kapitel werden Arbeiten zum Thema »Addition und Subtraktion mit mobilen Geräten« vorgestellt. Dabei handelt es sich vorwiegend um aktuelle iPhone Apps aus Apples' iTunes Store.¹ Diese Apps sind gewöhnlich für Kinder im Pflichtschulalter gedacht und dadurch auch meist grafisch ansprechend designt.

In den folgenden Abschnitten werden ein paar ausgewählte Apps vorgestellt.

2.1 MathBoard²

Diese App dient als »Best Practice« App im Bereich Mathematik. Aufgrunddessen wird sie auch von Apple selbst bei diversen Veranstaltungen präsentiert. In Abbildung 2.1 wird ein Screenshot dieser App gezeigt, auf dem sich aber bereits erkennen lässt, dass das Ziel dieser App nicht wirklich mit dem Ziel der in dieser Arbeit präsentierten App korreliert, und nur deswegen hier als Beispiel angeführt wird, weil Apple diese App forciert.

2.2 Addition Master: Mathematik Spiel³

In Abbildung 2.2 ist ersichtlich, dass die Benutzeroberfläche dieser App, und dabei vor allem die Präsentation der Zahleneingabemöglichkeit sehr

¹<https://itunes.apple.com/de/genre/ios/id36?mt=8>.

²<https://itunes.apple.com/de/app/mathboard/id373909837?mt=8>.

³<https://itunes.apple.com/de/app/addition-master-mathematik/id672669932?mt=8>.

2 Stand der Technik



Abbildung 2.1: Screenshot von Mathboard.

ähnlich der in dieser Arbeit vorgestellten App gestaltet wurde. Zum Funktionsumfang gehören hier:

- Trainingsmodus
- Statistik
- Übungsmodus

2.3 Addition !⁴ and Subtraction !⁵

Hierbei handelt es sich um zwei separat existierende Apps vom selben Entwickler zum Thema Addition und Subtraktion. In der Recherche waren diese zwei Apps auch die einzigen, bei denen der/die SchülerIn Überträge zur Rechenerleichterung notieren konnte. Funktionalität:

- 2 oder 3 Summanden

⁴<https://itunes.apple.com/de/app/addition-!/id447548669?mt=8>.

⁵<https://itunes.apple.com/de/app/subtraction-!/id447548515?mt=8>.

2 Stand der Technik

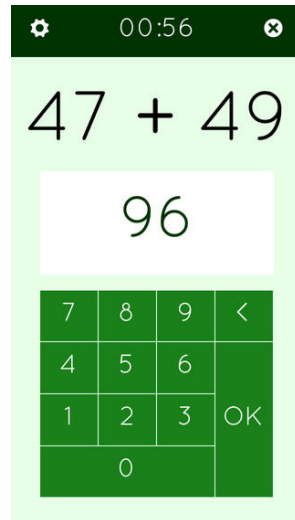


Abbildung 2.2: Screenshot von Addition Master: Mathematik Spiel.

- bis 6 Ziffern pro Summand
- Hilfe zur Problemstellung
- Tipp zur Problemstellung
- Tutorial in dem die App erklärt wird
- Editor für eigene Problemstellungen

In Abbildung 2.3 ist ein Screenshot der App »Addition !« zu sehen. Darauf ist ersichtlich, dass die Überträge über dem ersten Summanden einzutragen sind. Überträge über dem ersten Summanden zu notieren werden ist jedoch nur im englischsprachigen Raum üblich, im deutschsprachigen Raum werden die Überträge üblicherweise unter dem letzten Summanden notiert. In der in dieser Arbeit vorgestellten App ist es möglich die Felder für die Überträge entweder oben oder unten anzeigen zu lassen.

Abbildung 2.4 zeigt einen Screenshot der App »Subtraction !«. Dabei ist eine ausgeklügelte Methode zur Notierung der Überträge bei Subtraktionen ersichtlich.

2 Stand der Technik

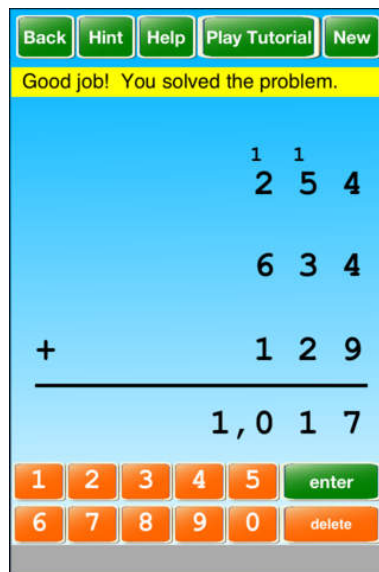


Abbildung 2.3: Screenshot von Addition !.

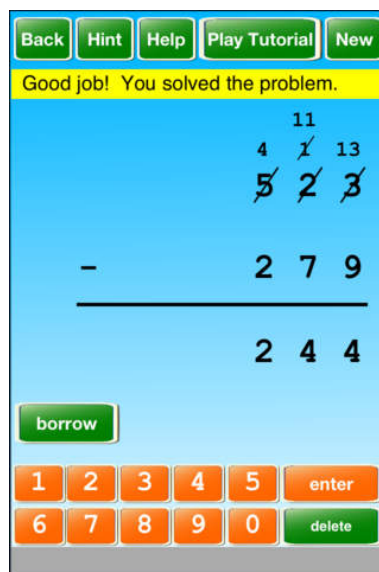


Abbildung 2.4: Screenshot von Subtraction !.



Abbildung 2.5: Screenshot von Add & Sub.

2.4 Weitere Apps

In diesem Abschnitt werden kurz weitere ausgewählte Apps im Bereich des mobilen Lernens vorgestellt.

Abbildung 2.5 zeigt die App »Add & Sub⁶«. Sie ist sehr einfach gehalten und auch in ihrem Funktionsumfang eingeschränkt.

Eine weitere Möglichkeit Mathematik Apps für Kinder attraktiv zu gestalten ist, die Apps als Spiele aufzubauen. Die Abbildungen 2.6 und 2.7 zeigen die Apps »Add & Sub with Springbird⁷« und »Addition & Subtraction For Kids⁸« die vor allem für SchülerInnen bis 10 Jahren auf dieses Prinzip setzt.

Weiters zu erwähnen sind die Apps:

⁶<https://itunes.apple.com/de/app/add-sub/id693077439?mt=8>.

⁷<https://itunes.apple.com/de/app/add-subtract-springbird-mathe/id601505771?mt=8>.

⁸<https://itunes.apple.com/de/app/addition-subtraction-for-kids/id426907035?mt=8>.

2 Stand der Technik

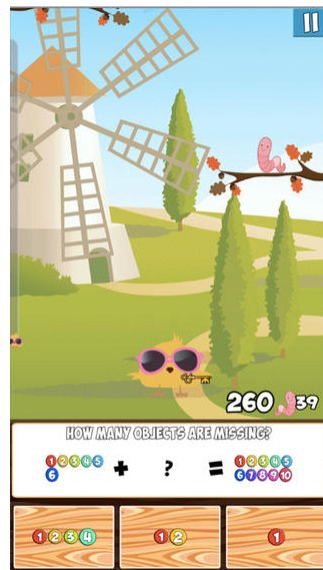


Abbildung 2.6: Screenshot von Add & Sub with Springbird.

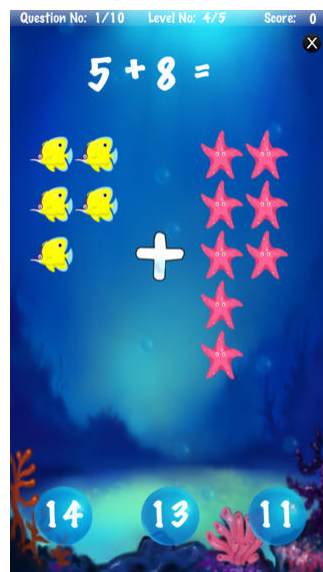


Abbildung 2.7: Screenshot von Add & Sub For Kids.

2 Stand der Technik

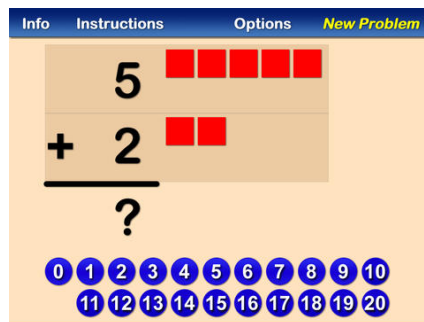


Abbildung 2.8: Screenshot von Add Sub K-1.

- »Add Sub K-1⁹« in Abbildung 2.8
- »Addition - Subtraction¹⁰« in Abbildung 2.9
- »Subtract with Fun¹¹« in Abbildung 2.10

⁹<https://itunes.apple.com/de/app/add-sub-k-1/id486199509?mt=8>.

¹⁰<https://itunes.apple.com/de/app/addition-subtraction/id542109601?mt=8>.

¹¹<https://itunes.apple.com/de/app/subtract-with-fun/id699563137?mt=8>.

2 Stand der Technik

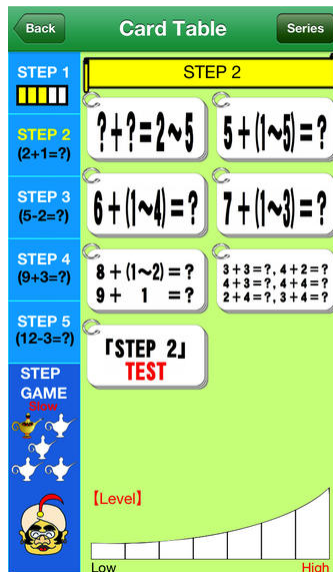


Abbildung 2.9: Screenshot von Addition - Subtraction.



Abbildung 2.10: Screenshot von Subtraction ith Fun.

3 Umsetzung

Das folgende Kapitel beschäftigt sich mit der technischen Umsetzung und einer Erläuterung der einzelnen Module des Additions- und Subtraktions-trainers. Zur Erstellung der App wurde die von Apple integrierte Entwicklungsumgebung »Xcode¹« in der Version »4.6.2« verwendet um die App zu schreiben. Die zu diesem Zweck verwendete Programmiersprache ist zum Großteil »Objective-C²« die als objektorientierte Erweiterung der Programmiersprache »C« angesehen wird.

Der Aufbau der Addition- und Subtraktionstrainer App wurde sehr modular gestaltet. Durch die Verwendung von »ViewControllern³« lässt sich dieser modulare Aufbau sauber beibehalten, was die Erweiterung der App vereinfacht.

Wegen dieser Modularisierung der App werden in den folgenden Abschnitten die technischen Einzelheiten pro Modul erläutert.

3.1 Hauptmenü

Das Hauptmenü dient als Einstiegspunkt in die App. Das heißt sobald der/die BenutzerIn die App startet landet er/sie sofort im Hauptmenü. Dieses gilt rein als Ausgangspunkt um die einzelnen Funktionalitäten der App aufrufen zu können. Das Hauptmenü besteht aus folgendenden Menüpunkten:

¹<http://developer.apple.com/xcode/> [Zugriff am 24.10.2013].

²<http://www.gnu.org/software/gnustep/resources/documentation/Developer/Base/ProgrammingManual/m>

³<http://developer.apple.com/library/ios/documentation/uikit/reference/UIViewController'Class/> [Zugriff a

3 Umsetzung

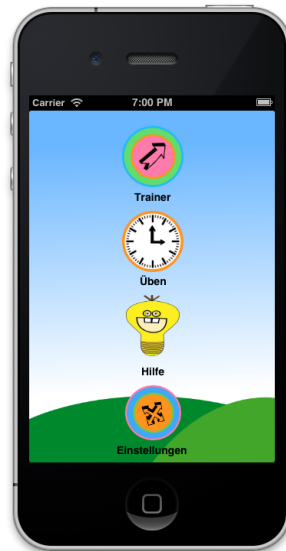


Abbildung 3.1: Screenshot des Hauptmenüs.

Trainer Webbasiertes Trainingssystem für das ein Konto benötigt wird. Abschnitt 3.2 erläutert dieses Modul.

Üben In unterschiedlichen Schwierigkeitsstufen anonym durchführbare Übung. Abschnitt ?? erläutert dieses Modul.

Hilfe Kurzer Hilfetext zur App

Einstellungen Einstellungen wie zum Beispiel die Ausrichtung des Übertrages

Da »Xcode« eine sehr komfortable Möglichkeit bietet diese Menüpunkte zu verlinken, hat das Hauptmenü auch keine weitere Funktionalität in Form von Programmcode. In Abbildung 3.1 sehen sie einen Screenshot des Hauptmenüs.

3.2 Trainer

Der Trainer bildet im eigentlichen Sinn das Herzstück der App. Sobald der Trainer gestartet wird, wird überprüft ob der/die BenutzerIn bereits Zugriff

3 Umsetzung



Abbildung 3.2: Screenshot der Registrierungs bzw. des Logins.

auf das Neuhold's System hat. Ist dies nicht der Fall muss sich der/die BenutzerIn über ein Webinterface für die Nutzung des Trainers registrieren. Sobald die Registrierung erfolgt ist, kann sich der/die Benutzerin am System einloggen. Abbildung 3.2 zeigt einen Screenshot der App auf dem die Registrierung bzw. der Login ersichtlich sind.

Die gesamte Kommunikation zwischen App und Neuhold's Webservice erfolgt über »SOAP⁴« Nachrichten. Das Nachrichtenformat für diesen Zweck ist »XML⁵«. »SOAP« Nachrichten selbst zu erstellen und an das Webservice zu schicken wäre äußerst umständlich und fehleranfällig, weswegen hierzu ein Service namens »Sudzc⁶« verwendet wurde. Dieses Service generiert aufgrund der SOAP Endpoint Definition automatisch Programmcode der beim Programmieren der App aufgerufen werden kann und über diesen das Senden von SOAP Nachrichten vereinfacht wird.

Folgender Codeausschnitt 3.1 zeigt, wie eine SOAP Nachricht für die Au-

⁴<http://www.w3.org/TR/soap9/>.

⁵<http://www.w3.org/XML/>.

⁶<http://sudzc.com>.

3 Umsetzung

thentifizierung aussieht. Der SOAP Endpoint für diese Nachricht ist »isUserAllowed«. Das Feld »username« wird im Klartext übertragen, das Passwort im Feld »password« ist SHA-256 gehasht (Beschrieben in NSA, 2001). Für die Nachvollziehbarkeit, auf welchem System der Additions- und Subtraktionstrainer ausgeführt wird, ist im Feld »idApp« die App ID der jeweiligen App einzutragen (iOS, Android, etc.). »hmacClient« ist ein Hash der sich aus Benutzername, gehashtem Passwort, App ID und App Key zusammensetzt.

```
1 <soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-  
  instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"  
  xmlns:soap="http://mathe.tugraz.at/~georg/Usermanager/public/  
  soap">  
2   <soapenv:Header/>  
3   <soapenv:Body>  
4     <soap:isUserAllowed soapenv:encodingStyle="http://schemas.  
      xmlsoap.org/soap/encoding/">  
5       <username xsi:type="xsd:string">?</username>  
6       <password xsi:type="xsd:string">?</password>  
7       <idApp xsi:type="xsd:int">?</idApp>  
8       <hmacClient xsi:type="xsd:string">?</hmacClient>  
9     </soap:isUserAllowed>  
10  </soapenv:Body>  
11 </soapenv:Envelope>
```

Listing 3.1: SOAP XML Nachricht zur Benutzer Authentifizierung

Codeausschnitt 3.2 zeigt, wie der Hash für die Authentifizierung erstellt wird. Codezeilen 20 und 21 zeigen den Zugriff auf die durch »SudzC« generierten Methoden zur SOAP Kommunikation.

```
1   const char *s = [pw  
    cStringUsingEncoding:NSUTF8StringEncoding];  
2   NSData *pwdata = [NSData dataWithBytes:s length:strlen(s)  
    ];  
3   uint8_t digest[CC_SHA256_DIGEST_LENGTH] = {0};  
4   CC_SHA256(pwdata.bytes, pwdata.length, digest);  
5   NSData *out = [NSData dataWithBytes:digest  
    length:CC_SHA256_DIGEST_LENGTH];  
6   NSString *hash = [out description];  
7   hash = [hash stringByReplacingOccurrencesOfString:@" "  
    withString:@""];
```

3 Umsetzung

```
8      hash = [hash stringByReplacingOccurrencesOfString:@"<"
9              withString:@""];
10     hash = [hash stringByReplacingOccurrencesOfString:@">"
11             withString:@""];
12     NSString *data = [NSString stringWithFormat:@"%s",
13                       uname, hash, @"12"];
14     const char *cKey = [key
15                         cStringUsingEncoding:NSUTF8StringEncoding];
16     const char *cData = [data
17                          cStringUsingEncoding:NSUTF8StringEncoding];
18     unsigned char dHMAC[CC_SHA256_DIGEST_LENGTH];
19     CCHmac(kCCHmacAlgSHA256, cKey, strlen(cKey), cData, strlen
20           (cData), dHMAC);
21     NSData *outhmac = [NSData dataWithBytes:dHMAC
22                         length:CC_SHA256_DIGEST_LENGTH];
23     NSString *hmacHash = [outhmac description];
24     hmacHash = [hmacHash
25                 stringByReplacingOccurrencesOfString:@" " withString:@""];
26     hmacHash = [hmacHash
27                 stringByReplacingOccurrencesOfString:@"<" withString:@""];
28     hmacHash = [hmacHash
29                 stringByReplacingOccurrencesOfString:@">" withString:@""];
30     adSubUsermanager_Soap_ManagementService *
31     userManagerService = [[
32         adSubUsermanager_Soap_ManagementService alloc] init];
33     [userManagerService isUserAllowed:self action:@selector(
34         handleUMService:) username:uname password:hash idApp:12
35         hmacClient:hmacHash];
```

Listing 3.2: Objective-C Code zur Erstellung des Hashs

Bei erfolgreicher Authentifizierung wird der/die BenutzerIn nochmals auf einen Zwischenscreen geleitet auf dem die Möglichkeit besteht sich wieder abzumelden bzw. mit dem Training zu beginnen. Abbildung 3.3 zeigt den korrespondierenden Screenshot. Ergänzend sei hier erwähnt, dass die App sich den aktuellen Benutzer merkt, und diesen automatisch einloggt bis zu dem Zeitpunkt an dem sich der Benutzer vom System manuell über den Menüpunkt »Abmelden« vom System ausloggt. Ob der/die BenutzerIn

3 Umsetzung

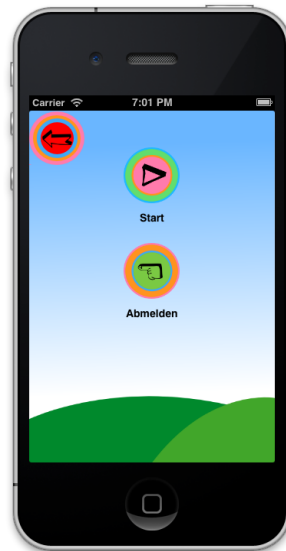


Abbildung 3.3: Screenshot des Logouts

eingeloggt ist wird über »NSUserDefaults⁷« gespeichert.

Beim Start des Trainers wird die Rechenaufgabe wie in Abbildung 3.4 dargestellt. Auf diesem Screen wird dem/der BenutzerIn ein an seine/ihre Bedürfnisse bzw. Kenntnisse angepasste Rechenaufgabe dargestellt. Die Schwierigkeit dieser Rechenaufgabe wird von Neuhold's System auf Basis der bisherigen Rechenleistungen des Benutzers festgelegt und übermittelt. Es werden auf diesem Screen fünf Eingabefelder präsentiert die durch eine jeweilige Berührung editierbar sind. Die drei großen Eingabefelder repräsentieren die Eingabe des Ergebnisses für die jeweilige Stelle der Zehnerpotenz. Die darüberliegenden kleineren Eingabefelder stellen die Überträge beim Rechnen von der jeweilig einen auf die andere Zehnerpotenz dar. Die Position der Übertragsfelder kann je nach Wunsch über die erste Zahl oder unter die Zweite Zahl der übermittelten Rechnung gesetzt werden. Dies ist über das Einstellungs Menü zu definieren. Durch Berühren des roten »X Symbols« werden alle bisher getätigten Eingabe gelöscht, und durch Berühren des grünen »Häkchen Symbols« wird die Eingabe bestätigt und

⁷<https://developer.apple.com/library/mac/documentation/cocoa/Reference/Foundation/Classes/NSUserDefaults>

3 Umsetzung

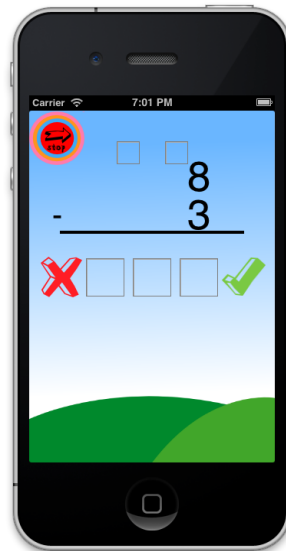


Abbildung 3.4: Screenshot des Trainers

an das Webservice übermittelt. Der/die BenutzerIn erhält sofort Feedback ob eine Rechenaufgabe erfolgreich gelöst wurde. Abbildung 3.5 zeigt den Feedback Screen.

Ist die Übermittlung der Ergebnisse abgeschlossen wird sofort darauf eine weitere Rechenaufgabe präsentiert. Dies geschieht so lange bis der/die BenutzerIn das »Stop Symbol« berührt. Daraufhin landet der/die BenutzerIn wieder im Menü mit der Möglichkeit des Logouts bzw. eines erneuten Trainingsdurchganges.

Die Kommunikation mit dem Webservice im Zuge eines Rechendurchganges enthält mehrere Schritte, die im folgenden Abschnitt 3.2.1 erläutert werden.

3.2.1 Kommunikationsablauf mit Webservice

In diesem Abschnitt werden die einzelnen Schritte in der Kommunikation zwischen App und Webservice erläutert. Abbildung 3.6 zeigt in einem

3 Umsetzung



Abbildung 3.5: Screenshot des Feedbacks

abgewandelten Sequenzdiagramm wie eine erfolgreiche Kommunikation mit dem Webservice abläuft. Ergänzend dazu werden in den folgenden Abschnitten Quelltextsnipsel zur Erklärung des Kommunikationsablaufs verwendet.

Anfordern einer Rechenaufgabe

Um für einen bestimmten Benutzer eine Rechenaufgabe zu erhalten muss das Webservice am Endpunkt »getProblem« durch Übergabe der Benutzer-ID angesprochen werden. Codeausschnitt 3.3 zeigt wie ein solcher Request aussieht. Das Feld »userID« repräsentiert die Benutzer-ID.

```
1 <soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-  
  instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"  
  xmlns:web="http://plusminus.tugraz.at/webservice">  
2   <soapenv:Header/>  
3   <soapenv:Body>
```

3 Umsetzung

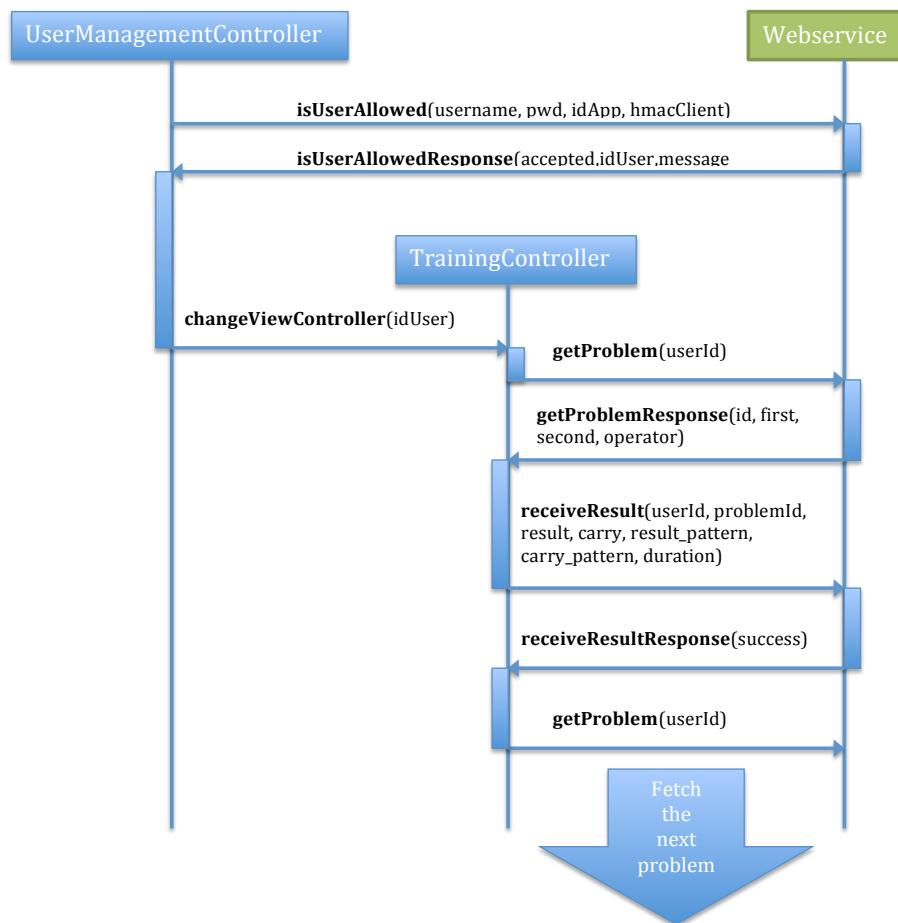


Abbildung 3.6: Ablauf der Kommunikation mit dem Webservice

3 Umsetzung

```
4      <web:getProblem soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
5          <userId xsi:type="xsd:int">1</userId>
6      </web:getProblem>
7  </soapenv:Body>
8 </soapenv:Envelope>
```

Listing 3.3: Anfordern einer Rechenaufgabe

Erhalten einer Rechenaufgabe

Wurde im vorherigen Request eine gültige Benutzer-ID übergeben, bekommt man vom Webservice eine Antwort mit der jeweiligen Rechenaufgabe. In welcher Form eine solche Rechenaufgabe vom Webservice übermittelt wird zeigt Codeausschnitt 3.4.

```
1 <SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="http://plusminus.tugraz.at/webservice" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/">
2   <SOAP-ENV:Body>
3     <ns1:getProblemResponse>
4       <return xsi:type="ns1:WebserviceProblem">
5         <id xsi:type="xsd:int">681116</id>
6         <first xsi:type="xsd:int">6</first>
7         <second xsi:type="xsd:int">4</second>
8         <operator xsi:type="xsd:string">-</operator>
9       </return>
10    </ns1:getProblemResponse>
11  </SOAP-ENV:Body>
12 </SOAP-ENV:Envelope>
```

Listing 3.4: Erhalten einer Rechenaufgabe

Das Feld »id« enthält die eindeutige Identifikationsnummer der jeweiligen Rechenaufgabe die in weiterer Folge zur Übermittlung des Ergebnisses wieder benötigt wird. In den Feldern »first« bzw. »second« werden die für eine Addition bzw. Subtraktion benötigten Operanden übertragen und

3 Umsetzung

im Feld »operator« wird der Operator übermittelt, in diesem Falle entweder »+« oder »-«. Diese Daten werden nach Erhalt geparsed und wie in Abbildung 3.4 in der App dargestellt.

Übermitteln des Ergebnisses

Sobald die Rechnung gelöst, und der Bestätigungs-Button berührt wurde, wird nun das Ergebnis wieder an das Webservice übermittelt. Codeausschnitt 3.5 zeigt die zugehörige SOAP Nachricht.

```
1 <soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-  
  instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"  
  xmlns:web="http://plusminus.tugraz.at/webservice">  
2   <soapenv:Header/>  
3   <soapenv:Body>  
4     <web:receiveResult soapenv:encodingStyle="http://schemas.  
      xmlsoap.org/soap/encoding/">  
5       <userId xsi:type="xsd:int">1</userId>  
6       <problemId xsi:type="xsd:int">681116</problemId>  
7       <result xsi:type="xsd:int">2</result>  
8       <carry xsi:type="xsd:int">0</carry>  
9       <result_pattern xsi:type="xsd:string">een</result_pattern  
10      >  
11       <carry_pattern xsi:type="xsd:string">ee</carry_pattern>  
12       <duration xsi:type="xsd:float">5</duration>  
13     </web:receiveResult>  
14   </soapenv:Body>  
</soapenv:Envelope>
```

Listing 3.5: Übermitteln des Resultats

Wie bereits erwähnt enthält das Feld »userID« die Benutzer Identifikationsnummer und das Feld »problemID« die Rechnungs Identifikationsnummer. Im Feld »result« wird das Ergebnis des Benutzers übermittelt. Das Feld »carry« kann maximal zwei Ziffern lang sein und enthält die vom Benutzer eingetragenen Zahlen in den jeweiligen Übertragsfeldern in der App. Die Einerstelle des »carry« Feldes in der Soapnachricht enthält den Übertrag der beim Rechnen an der Einerstelle in der App entstanden ist und sinngemäß gilt das für die Zehnerstelle des »carry« Feldes. Wird kein Übertrag in der App eingetragen wird automatisch die Zahl null übertragen.

3 Umsetzung

Die Felder »result_pattern« und »carry_pattern« geben an in welches der Eingabefelder in der App der Benutzer eine Zahl eingetragen hat, bzw ob er das Eingabefeld leer stehen gelassen hat. Dargestellt wird das »result_pattern« und das »carry_pattern« als Folge der Buchstaben »e« bzw. »n«, wobei »e« für empty und »n« für number steht. Der Buchstabe im jeweiligen Pattern an der ganz rechten Stelle repräsentiert die Einerstelle, und der Buchstabe ganz links repräsentiert die Hunderterstelle in der App. Als letzts Feld wird »duration« übertragen, welches die benötigte Zeit zur Durchführung der jeweiligen Rechenaufgabe erfasst.

Erhalt der Lösung

Als Antwort auf die Übermittlung des Ergebnisses bekommt man vom Webservice die Nachricht in Codeausschnitt 3.6.

```
1 <SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="http://plusminus.tugraz.at/webservice" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/">
2   <SOAP-ENV:Body>
3     <ns1:receiveResultResponse>
4       <return xsi:type="xsd:boolean">true</return>
5     </ns1:receiveResultResponse>
6   </SOAP-ENV:Body>
7 </SOAP-ENV:Envelope>
```

Listing 3.6: Antwort auf Übermittlung des Ergebnisses

Das Feld »return« enthält entweder true oder false, je nach Erfolg.

3.3 Üben

Den zweiten großen Teil der App stellt der Übungsmodus dar. Über diesen kann der/die BenutzerIn anonym an seinen eigenen Rechenfähigkeit arbeiten. Der Übungsmodus wird über das Hauptmenü aufgerufen und bietet

3 Umsetzung



Abbildung 3.7: Screenshot des Schwierigkeitsgradauswahl im Übungsmodus

drei Schwierigkeitsgrade zur Übung von Additionen und Subtraktionen an. Abbildung 3.7 zeigt den Screen zur Auswahl des Schwierigkeitsgrades. Sobald eine der drei Schwierigkeitsgrade ausgewählt wurde, startet der Übungsmodus auch sofort.

Die Rechenaufgaben im Übungsmodus unterliegen einigen Einschränkungen. Diese Einschränkungen wären:

- Das Ergebnis der Rechenaufgabe darf nicht negativ sein
- Das Ergebnis der Rechenaufgabe darf nicht größer als 1000 sein
- Rechenaufgaben in der Schwierigkeitsstufe »leicht« dürfen nur maximal einstellige Operanden enthalten
- Rechenaufgaben in der Schwierigkeitsstufe »mittel« dürfen nur maximal zweistellige Operanden enthalten
- Rechenaufgaben in der Schwierigkeitsstufe »schwer« dürfen nur maximal dreistellige Operanden enthalten

Aufgrund dieser Einschränkungen war es notwendig bei der Erstellung der Rechenaufgaben viele Fälle zu unterscheiden. Codeausschnitt 3.7 zeigt

3 Umsetzung

wie die Rechenaufgaben generiert werden. Zuerst wird zufällig ein erster Operand und zweiter Operand zwischen 0 und 999, und ein Operator generiert. Aufgrund des Schwierigkeitsgrades wird nun entschieden, ob die jeweiligen Operanden durch 100 für leichte Rechenaufgaben, durch 10 für mittelschwere Rechenaufgaben oder gar nicht für schwere Rechenaufgaben dividiert werden. Dadurch, dass mit Integerzahlen gerechnet wird, gibt es nur ganzzahlige Ergebnisse bei der Division und wir können das Ergebnis der Division sofort verwenden. Weiters wird nun entschieden, dass wenn der erste generierte Operand kleiner als der zweite generierte Operand ist, es aufgrund unserer Einschränkungen nur eine Addition werden darf. Wenn wir nun mit einer Addition weiterarbeiten und das Ergebnis der Addition aber größer als 1000 wäre müssen wir eine Kombination zweier Operanden finden bei der die Summe der Operanden kleiner als 1000 ist. Dies geschieht in Zeile 22. Im weiteren Verlauf in Codeausschnitt 3.7 geht es noch darum, die generierten Operanden in Strings umzuwandeln und aufzubereiten um sie in der App darstellen zu können.

```
1      int plusMinus = (arc4random() % 2); // 0 or 1 for + or -
2      int firstNum = (arc4random() % 1000); // number between 0 and
      999
3      int secondNum = (arc4random() % 1000); // number between 0 and
      999
4
5     NSUserDefaults *defaults = [NSUserDefaults
      standardUserDefaults];
6      int difficultyLevel = [defaults integerForKey:@"
      exerciseDifficulty"]; //fetch difficultyLevel
7
8      if(difficultyLevel == 1) { //easy
9          firstNum = firstNum / 100;
10         secondNum = secondNum / 100;
11     } else if (difficultyLevel == 2) { //medium
12         firstNum = firstNum / 10;
13         secondNum = secondNum / 10;
14     }
15
16     //if difficultylevel = 3, hard, nothing has to be divided
17
18     if (firstNum < secondNum) //if first num is smaller than
        second it can only be a addition
19         plusMinus = 0;          //no negative results allowed
```


3 Umsetzung

```
20
21 if (plusMinus == 0 && (firstNum + secondNum) >= 1000) { // if
    it's an addition and result > 1000
22     while((firstNum + secondNum) > 1000) // find combinations
        of the two operands which are summed
23     {                                     //smaller than 1000
24         firstNum = (arc4random() % 1000);
25         secondNum = (arc4random() % 1000);
26     }
27
28 }
29 NSString *firstNumString = [NSString stringWithFormat:@"%d",
    firstNum]; //convert number to strings
30 NSString *secondNumString = [NSString stringWithFormat:@"%d",
    secondNum]; //for presentation purposes
31 NSString *firstOnes = 0; //initialize with 0
32 NSString *firstTens = 0;
33 NSString *firstHundreds = 0;
34 NSString *secondOnes = 0;
35 NSString *secondTens = 0;
36 NSString *secondHundreds = 0;
37 if ([firstNumString length] == 3) { //if first number has 3
    digits
38     firstOnes = [firstNumString substringWithRange:NSMakeRangeRange
        (2, 1)]; // separate them correctly
39     firstTens = [firstNumString substringWithRange:NSMakeRangeRange
        (1, 1)];
40     firstHundreds = [firstNumString
        substringWithRange:NSMakeRangeRange(0, 1)];
41
42 } else if ([firstNumString length] == 2) { // if first number
    has 2 digits
43     firstOnes = [firstNumString substringWithRange:NSMakeRangeRange
        (1, 1)]; // separate them correctly
44     firstTens = [firstNumString substringWithRange:NSMakeRangeRange
        (0, 1)];
45     firstHundreds = @""; //and empty hundreds field
46 } else { // if first number digits
47
48     firstOnes = [firstNumString substringWithRange:NSMakeRangeRange
        (0, 1)]; // separate them correctly
49     firstTens = @""; //and tens hundreds field
50     firstHundreds = @""; //and empty hundreds field
51 }
```

3 Umsetzung

```
52
53 //same stuff for the second number
54
55 if ([secondNumString length] == 3) {
56     secondOnes = [secondNumString
57         substringWithRange:NSMakeRange(2, 1)];
58     secondTens = [secondNumString
59         substringWithRange:NSMakeRange(1, 1)];
60     secondHundreds = [secondNumString
61         substringWithRange:NSMakeRange(0, 1)];
62
63 } else if ([secondNumString length] == 2) {
64     secondOnes = [secondNumString
65         substringWithRange:NSMakeRange(1, 1)];
66     secondTens = [secondNumString
67         substringWithRange:NSMakeRange(0, 1)];
68     secondHundreds = @"";
69 } else {
70
71     secondOnes = [secondNumString
72         substringWithRange:NSMakeRange(0, 1)];
73     secondTens = @"";
74     secondHundreds = @"";
75 }
76
77 //fill screen in app
78 self.firstOnes.text = firstOnes;
79 self.firstTens.text = firstTens;
80 self.firstHundreds.text = firstHundreds;
81
82 self.secondOnes.text = secondOnes;
83 self.secondTens.text = secondTens;
84 self.secondHundreds.text = secondHundreds;
85
86 //insert + or -
87 self.plusMinus.text = (plusMinus == 0) ? @"+" : @"-";
88 //calculate desired result for evaluation purposes
89 self.desiredResult = (plusMinus == 0) ? (firstNum + secondNum)
90     : (firstNum - secondNum);
```

Listing 3.7: Generierung von Rechenaufgaben im Übungsmodus

Die Darstellung der Rechenaufgaben im Übungsmodus sieht grundsätzlich

3 Umsetzung



Abbildung 3.8: Screenshot des Auswertungsscreens im Übungsmodus

gleich aus wie die Darstellung im Trainingsmodus in Abbildung 3.4. Im Unterschied zum Trainingsmodus bekommt der/die BenutzerIn nach Beendigung des Übungsmodus durch Berühren des Stop-Buttons eine Auswertung seiner Übungsleistung dargestellt. Dazu wird die Anzahl der korrekt gelösten Übungsrechenaufgaben mitgeloggt. Abbildung 3.8 zeigt eine beispielhafte Auswertung eines Übungsdurchlaufs.

3.4 Einstellungen

Unter Einstellungen im Hauptmenü hat der/die BenutzerIn die Möglichkeit die Position des Übertragfeldes im Trainings -und Übungsmodus zu verändern. Im englischsprachigen Raum ist es üblich das Übertragfeld ober dem ersten Operanden zu positionieren. Im deutschsprachigen Raum ist das Übertragfeld gewöhnlich unter dem zweiten Operanden zu finden. Abbildung 3.9 zeigt den Einstellungsscreen.

3 Umsetzung



Abbildung 3.9: Screenshot des Einstellungsscreens

3.5 Hilfe

Im Menüpunkt Hilfe bekommt der/die BenutzerIn eine kurze Erklärung der Funktionalität der App. Abbildung 3.10 zeigt den Hilfescreen.

3 Umsetzung



Abbildung 3.10: Screenshot des Hilfescreens

4 Diskussion

Das folgende Kapitel geht auf die Vorteile und auch auf Defizite der entwickelten App ein und diskutiert Erweiterungsmöglichkeiten der App, die im Rahmen der Entwicklung nicht berücksichtigt wurden.

Der Fokus der Entwicklung, der in dieser Arbeit vorgestellten App sollte deutlich erkennbar sein, nämlich die Klarheit und Einfachheit in der Bedienung und Darstellung der Inhalte. Die Umsetzung der App in der beschriebenen Art und Weise ist mehreren Tatsachen geschuldet. Zum einen gelten als Adressaten der App vor allem Kinder im Volksschulalter, die entweder gerade beim Erlernen der Grundrechenarten sind, oder Defizite dabei aufweisen, und zum anderen die mobile Plattform selbst. Da Schüler oft schon zu rechnen beginnen bevor sie überhaupt lesen können, wurde der Aufbau und das Design der App so gewählt, dass man eigentlich auch ohne lesen zu können die App bedienen kann. Um Kinder zu ermutigen sich mit dieser App zu befassen wurde auch der Fokus im Design auf kindliche Symbole und Grafiken gelegt. Hierbei sei erwähnt, dass das grafische Design der App von zwei Schülerinnen der Ortweinschule Graz entwickelt wurde. Zum anderen gelten bei der Erstellung von Apps für Apples mobiles Betriebssystem iOS gewisse Richtlinien¹ die einzuhalten sind. Diese schränken den Entwickler ein, zu komplexe User Interfaces zu entwerfen, um so die Bedienung der App einfach und konsistent über alle Bildschirme zu halten.

Der Additions- und Subtraktionstrainer dient einzig und allein der Darstellung von Rechenaufgaben. Diese kommen wie bereits in Abschnitt 3.2 erwähnt entweder von Neuhold's Webservice, oder werden, wie Abschnitt 3.3 erläutert, von der App nach gewissen Kriterien generiert. Als Erweiterung dieser App, wäre es durchaus denkbar einen Menüpunkt einzubauen,

¹<https://developer.apple.com/library/ios/documentation/userexperience/conceptual/mobilehig/>.

4 Diskussion

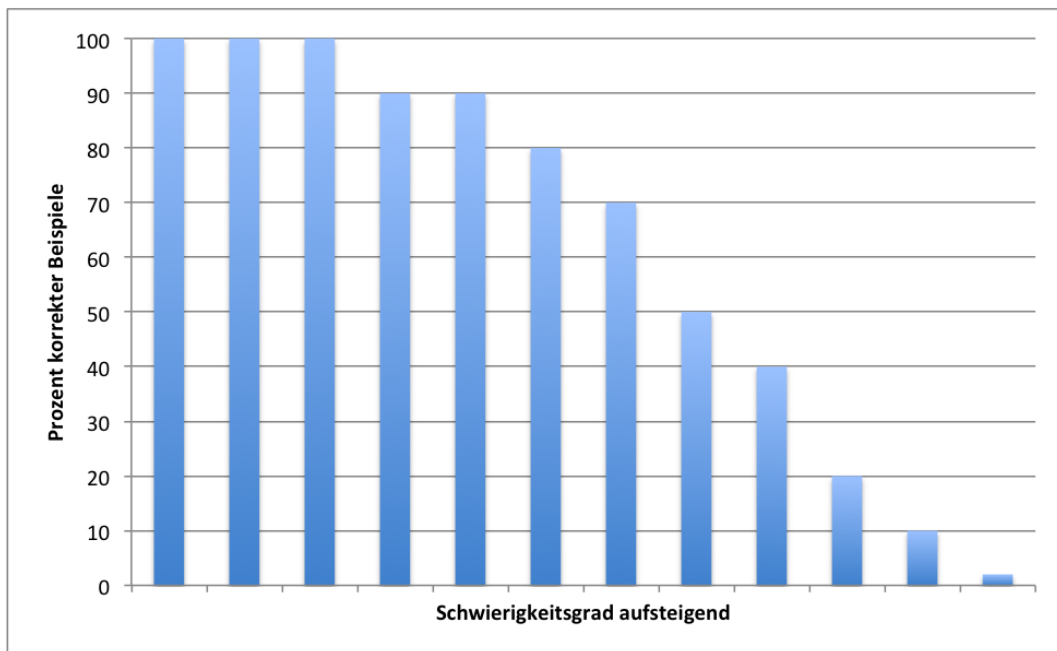


Abbildung 4.1: Statistik über die Leistung des/der SchülerIn

über den man eine Statistik einsehen kann. Denkbar hierbei wäre zum Beispiel anzuzeigen, wieviel Prozent aller gestellten Rechenaufgaben vom Schüler korrekt gelöst wurden, oder weiters eine grafische Darstellung wieviel Prozent der Rechenaufgaben mit steigendem Schwierigkeitsgrad erfüllt wurden. Abbildung 4.1 zeigt wie so ein Diagramm aussehen könnte.

Vorstellbar wäre auch ein textueller Hinweis(zB. »Du solltest zweistellige Subtraktionen noch etwas üben denn du weißt ja, Übung macht den Meister«), der auf Wunsch in der App angezeigt wird, über die Schwerpunkte die sich ein Schüler beim Üben legen sollte. Um diese gerade diskutierten Erweiterungen umzusetzen Bedarf es jedoch Erweiterungen an Neuhold's Webservice, denn wie bereits erwähnt speichert die App keinen Verlauf der Rechenerfolge des Benutzers. Da diese Statistiken im Sytem jedoch vorliegen, dürfte es keine größeren Schwierigkeiten bereiten weitere Schnittstellen des Webservice für die App bereitzustellen. Eine weitere vorstellbare Ergänzung in der App wäre eine Rangliste für den Übungsmodus.

5 Zusammenfassung und Ausblick

In diesem Kapitel wird noch einmal kurz zusammengefasst, was in dieser Arbeit beschrieben wurde. Bevor man sich der Aufgabe stellt eine mobile Applikation für ein Smartphone zu entwickeln, sollte man sich zuerst einen Überblick verschaffen, ob es, und in welcher Form es bereits Arbeiten zu diesem Thema gibt. Dieser Anforderung wurde in Kapitel 2 Rechnung getragen, und einige ähnlich Arbeiten recherchiert. In Kapitel 3 wurden die technischen Details der Entwicklung erläutert. Der Schwerpunkt dieser Arbeit lag in der Entwicklung der App, warum dieses Kapitel auch das umfangreichste in dieser Arbeit darstellt, wobei zum Verständnis der Applikation Abbildung 3.6 auf Seite 19 einen guten Überblick darstellt. In diesem Kapitel findet man auch Screenshots zur entwickelten App. Mögliche Ergänzungen zur App wurden in Kapitel 4 diskutiert.

Die Anwendung von Technologien im mobilen Bereich steckt noch immer in ihren Kinderschuhen, nimmt aber auch gleichzeitig immer mehr Fahrt auf, wie es auch die Arbeit von Ebner, 2013 zeigt. Aufgrund des großen Potenzials von »Mobile Learning«, können wir uns erwarten, dass die Entwicklungsgeschwindigkeit in diesem Forschungsgebiet weiter rapide steigen wird.

Um nun einen Ausblick auf die Entwicklung des »Mobile Learning« zu wagen, sollte man sich zuerst kurz überlegen, was man unter »Mobile Learning« eigentlich verstehen sollte. In der Arbeit von Brown und Diaz, 2010 wurde eine Einteilung von »Mobile Learning« Geräten gemacht, die wie folgt aussieht:

Highly Mobile Device Gerät in der Größe eines Mobiltelefons, das man in der Hosentasche unterbringen kann: »Feature Phones« (reine Sprach und Text Services), Smartphones und andere ähnliche Geräte

Very Mobile Device Netbooks, Pads, Slates

Mobile Device Größere Geräte wie Notebooks

Die Verwendung des Begriffs »Mobile Learning« wird aufgrund der Tatsache, dass es sich hierbei um eine Smartphone App handelt, auf »Highly Mobile Devices« bezogen. Es ist vorstellbar, dass »Mobile Learning« gerade in Entwicklungsländern mit ihren sehr jungen Altersstrukturen aufgrund vieler Kinder, seinen großen Durchbruch erleben könnte. Hierbei zu erwähnen sind diverse Initiativen von großen Hardwareherstellern wie Microsoft¹ Nokia² und Huawei³, zur Verbreitung von günstigen Smartphones in diesen Ländern.

Die Bildungsqualität leidet dort unter den hohen Schülerzahlen wogegen man mit gezielter Förderung und auch Förderung durch maßgeschneiderte Rechenbeispiele, wie es die in dieser Arbeit beschriebene App macht, einen großen Schritt bei der Entwicklung nach vorne machen kann.

¹http://www.epo.de/index.php?option=com_content&view=article&id=1809:microsoft-will-smartphone-fu

²<http://www.n24.de/n24/Nachrichten/Netzwelt/d/1471900/nokia-setzt-auf-entwicklungslaender.html>.

³<http://ht4u.net/news/27090'huawei'bringt'windows'phone'4afrika'-`smartphone'fuer'entwicklungslaend>

Appendix

Literatur

- Brown, Malcolm und Veronica Diaz (Mai 2010). *Mobile Learning: Context and Prospects*. URL: <https://net.educause.edu/ir/library/pdf/ELI3022.pdf> (siehe S. 32).
- Ebner, Martin (Mai 2013). *Lehrbuch für Lernen und Lehren mit Technologien (2. Auflage)*. URL: <http://13t.tugraz.at/index.php/LehrbuchEbner10/pod/current> (siehe S. 32).
- Neuhold, Benedikt (Juli 2013). *Adaptives Informationssystem zur Erlernung mehrstelliger Addition und Subtraktion* (siehe S. 1).
- NSA (Juli 2001). *Descriptions of SHA-256, SHA-384, and SHA-512*. URL: <http://csrc.nist.gov/groups/STM/cavp/documents/shs/sha256-384-512.pdf> (siehe S. 14).