

Systemtechnik Labor

5BHIT 2017/18

GK10.2 "Mobile Dienste"

Laborprotokoll

Marvin Ertl

11. April 2018

Bewertung:

Betreuer: Michael Borko

Version: 1.0

Begonnen: 21.03.2018

Beendet: 11.04.2018

Inhaltsverzeichnis

1	Einführung	3
1.1	Ziele	3
1.2	Voraussetzungen	3
1.3	Aufgabenstellung	3
1.4	Bewertung	3
2	Implementierungsumgebung	4
2.1	UWP	4
2.2	Visual Studio	4
3	REST - Schnittstelle	5
3.1	Deployen	5
3.2	Verwendung	6
4	Mobilen Applikationen	6
4.1	UML	6
4.2	Register	6
4.3	Login	6
4.4	Mainpage	6
4.5	Deployen	6
5	Testen	6
5.1	UI Automation	6
5.2	Testaufbau	6
6	Konfiguration	7
6.1	Optionen	7
6.2	Variablen	7
7	Kommandos	8
7.1	makefig	8
7.2	vardef	8
8	Anwendung	9
8.1	Tabellen	9
8.1.1	TabularX	9
8.2	Aufzählung	10
8.2.1	Outlines	10
8.3	Glossar	10
8.4	Zitate	10
8.5	Quelltext	11
8.5.1	Listings	11
8.5.2	Minted	12
	Literaturverzeichnis	12

1 Einführung

Diese Übung gibt einen Einblick in Entwicklungen von mobilen Applikationen.

1.1 Ziele

Das Ziel dieser Übung ist eine Anbindung einer mobilen Applikation an ein Webservices.

Die Anbindung soll mit Hilfe eines RESTful Webservice umgesetzt werden.

1.2 Voraussetzungen

- Grundlagen einer höheren Programmiersprache und XML
- Grundlegendes Verständnis über Entwicklungs- und Simulationsumgebungen
- Verständnis von RESTful Webservices
- Download der entsprechenden Entwicklungsumgebung

1.3 Aufgabenstellung

Es ist eine mobile Anwendung zu implementieren, die sich an das Webservice aus der Übung GK9.3 "Cloud-Datenmanagement" anbinden soll. Dabei müssen die entwickelten Schnittstellen entsprechend angesprochen werden.

Es ist freigestellt, welche mobile Implementierungsumgebung dafür gewählt wird. Empfohlen wird aber eine Implementierung auf Android.

1.4 Bewertung

- Gruppengröße: 1 Person
- Anforderung **"überwiegend erfüllt"**
 - Dokumentation und Beschreibung der angewendeten Schnittstelle
 - Anbindung einer mobilen Applikation an die Webservice-Schnittstelle
 - Registrierung von Benutzern
 - Login und Anzeige einer Willkommensnachricht
- Anforderungen **zur Gänze erfüllt"**
 - Simulation bzw. Deployment auf mobilem Gerät
 - Überprüfung der funktionalen Anforderungen mittels Regressionstests

2 Implementierungsumgebung

Als Implementierungsumgebung wurde die Universelle Windows-Plattform (UWP) gewählt, um den Einblick in eine neue Umgebung zu bekommen.

2.1 UWP

Die UWP bietet die Möglichkeit eine Anwendung nur einmal zu entwickeln und auf allen Windows Geräten auszuführen. Dadurch kann die, in dieser Übung, entwickelte App auf vielen unterschiedlichen Geräten verwendet werden, unter anderem auf mobilen Geräten, wie Smartphones und Tablets, auf PCs, HoloLens und viele weitere. Im Fall dieser Übung wurde die Anwendung aber nur mit PCs und mobilen Geräten getestet.

Um diese Universellen Apps zu ermöglichen, bietet UWP die selbe Kern-API auf allen Geräten an. Dadurch kann auch ein App Store als einheitliche Vertriebsplattform auf unterschiedliche Geräteformen verwendet werden.



Abbildung 1: Darstellung der UWP

Dennoch kann für jedes Gerät spezieller Code geschrieben werden sodass die Anwendung ein unterschiedliches Verhalten je nach Gerät aufweist. Außerdem bietet UWP universelle Steuerelemente, womit eine Benutzeroberfläche entwickelt werden kann, welche sich an die Auflösung, DPI-Dichte, Eingabeart von Geräten anpasst.

UWP ermöglicht das Entwickeln mittels Visual C++, C#, Visual Basic und JavaScript, wobei mit den Programmiersprachen Visual C++, C# und Visual Basic XAML verwendet werden kann um die Benutzeroberfläche zu definieren. [1]

XAML beschreiben

Bei dieser Übung wurde C# mit XAML zur Umsetzung der Aufgabe verwendet.

2.2 Visual Studio

Als am besten geeignetes Tool erweist sich Visual Studio, welches eine optimale Unterstützung für die Entwicklung von UWP Apps bietet. Bei dieser Übung wurde Visual Studio 2017 Enterprise Edition verwendet, wobei darauf zu achten ist, dass zur Verwendung von Emulatoren für andere Geräte, wie zum Beispiel Smartphones, HoloLens, ... , Visual Studio 2015 installiert sein muss. Zwar können die Emulatoren für Visual Studio 2017 gedownloadet werden, diese werden aber im Editor dann nicht angezeigt,

wenn Visual Studio 2015 nicht installiert ist.

Im Visual Studio Installer kann im Reiter **Einzelne Komponenten** Emulatoren für die gewünschte Windows 10 Version gedownloadet werden.



Abbildung 2: Installieren von Emulatoren

In Visual Studio kann anschließend die App auf den gewünschten Gerät ausgeführt werden.

3 REST - Schnittstelle

Als REST - Schnittstelle wurde die entwickelte Anwendung für die Übung GK9.3 verwendet. Diese wurde nur leicht angepasst, sodass JSON Objekte gesendet werden.

3.1 Deployen

Die Anwendung ist im Unterverzeichnis "Cloud-Datenmanagement-Json" zu finden. Diese kann einfach deployed werden, da es sich um ein mit Maven erstelltes Programm handelt.

Mittels folgendem Befehl wird die Anwendung auf den Port 8080 lokal gestartet.

3.2 Verwendung

4 Mobilen Applikationen

4.1 UML

4.2 Register

4.3 Login

4.4 Mainpage

4.5 Deployen

5 Testen

5.1 UI Automation

5.2 Testaufbau

6 Konfiguration

6.1 Optionen

landscape	Richte das Dokument vertikal aus.
minted	Nutze das minted Paket zur Quelltextdarstellung.
natbib	Nutze NatBib zur Literaturverwaltung.
nobib	Deaktiviere die Literaturverwaltung.
nofonts	Nutze die Standard \LaTeX Schriftarten.
noglo	Deaktiviere Akronyme und das Glossar.
nologos	Zeichne keine Logos auf der Titelseite.
notitle	Zeichne keine Titelseite.
notoc	Zeichne kein Inhaltsverzeichnis.
notable	Zeichne keine Tabelle auf der Titelseite.

6.2 Variablen

Variablen werden über Kommandos gesetzt, die als Parameter den gewünschten Wert erhalten.

1 `\myvariable{value}`

Kommando	Inhalt
mysubtitle	Untertitel oder Zugehörigkeit
mysubject	Thema / Fach, welches bearbeitet wird
mycourse	Kurs / Klasse welche(r) besucht wird
myteacher	Betreuende Lehrkraft
myversion	Aktuelle Version des Dokuments
mybegin	Datum des Beginns
myfinish	Datum an dem die Arbeit beendet wurde

7 Kommandos

7.1 makefig

```

1 \makefig{images/hit-logo.png}{height=2cm}{
2   Mit Beschreibung und Label % (Optional)
3 }{
4   fig:caption-label          % (Optional)
5 }
```

Auflistung 1: makefig



Abbildung 3: Mit Beschreibung und Label

7.2 vardef

```

1 $$e^{i*\pi} = -1$$
```

$$e^{i*\pi} = -1$$

```

2 \begin{vardef}
3   \addvardef{$e$}{Eulersche Zahl}
4   \addvardef{$\pi$}{Kreiszahl}
5   \addvardef{$i$}{Imaginäre Einheit}
6 \end{vardef}
```

e ... Eulersche Zahl
 π ... Kreiszahl
 i ... Imaginäre Einheit

Auflistung 2: vardef

8 Anwendung

Hier sollen die Schritte der Laborübung erläutert werden. Hier sind alle Fragestellungen der Lehrkraft zu beantworten. Etwaige Probleme bzw. Schwierigkeiten sollten ebenfalls hier angeführt werden.

In diesem Fall werden einige \LaTeX -Elemente dokumentiert, welche bei der Kreation von Protokollen behilflich sein könnten.

8.1 Tabellen

Header	Kopf
Lorem	Ipsum dolor sit amet, consetetur sadipscing elitr
Ipsum	At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus
Dolor	Consetetur sadipscing elitr, sed diam nonumy

Tabelle 1: Tabular

8.1.1 TabularX

TabularX erlaubt die Angabe der Größe der Tabelle und bietet zudem den Reihentyp X, der die verbleibende Größe neben anderen Reihen mit anderen X Reihen teilt.

ACHTUNG: Die Verwendung von `\codein`, `\mintinline` oder `\lstinline` ist in einer TabularX Umgebung nicht möglich!

Header	Kopf
Lorem	Ipsum dolor sit amet, consetetur sadipscing elitr
Ipsum	At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus
Dolor	Consetetur sadipscing elitr, sed diam nonumy

Tabelle 2: TabularX

8.2 Aufzählung

- Element einer Aufzählung
 - Erstes eingerücktes Element einer Aufzählung
 - Zweites eingerücktes Element einer Aufzählung

8.2.1 Outlines

- Element einer Aufzählung
 - Erstes eingerücktes Element einer Aufzählung
 - Zweites eingerücktes Element einer Aufzählung

8.3 Glossar

Zur Verwaltung des Glossars wird standardmäßig die Datei `glossaries.tex` verwendet, wobei sowohl Definitionen als auch Akronyme definiert werden können.

Als Beispiel wurde ein Akronym für Systemtechnik (SYT) und eine Definition zu SYT selbst hinzugefügt.

```
1 \newacronym{ac-syt}{SYT}{Systemtechnik}
2 \newglossaryentry{sy}{
3     name={Systemtechnik},
4     description={\enquote{Als Systemtechnik bezeichnet man verschiedene Aufbau- und
5         ↳ Verbindungstechniken, aber auch eine Fachrichtung der Ingenieurwissenschaften.
6         ↳ Er bedeutet in der Unterscheidung zu den Mikrotechnologien die Verbindung
7         ↳ verschiedener einzelner Module eines Systems und deren Konzeption.}
8         ↳ \cite{wiki:syt}}
9 }
```

Im Dokument selbst kann ein Akronym mittels `\gls{ac-syt}` verwendet werden. Beachte, dass ein Akronym welches bereits im Dokument verwendet wurde, bei der ersten Verwendung ausgeschreiben und danach immer gekürzt wird.

Mit `\gls{sy}` kann zum Beispiel eine Referenz zur Definition von [Systemtechnik](#) hinzugefügt werden.

8.4 Zitate

Zitate sollten gesammelt in der Datei `bib.bib` verwaltet werden.

8.5 Quelltext

```
1 \begin {code}[] {java}
2 // Ich bin ein Kommentar!
3 public static void main(String[] args) {
4     System.out.println("Ich bin ein Array!")
5 }
6 \end {code}
```

Auflistung 3: Java Code

Die Darstellung von Quelltext im Text ist über das Kommando `\codein[options]{lang}{code}` möglich.

Eine einzelne Zeile kann mittels

```
1 \codeline [options]{lang}{code}
```

eingefügt werden.

8.5.1 Listings

```
1 \begin {lstlisting}[language=Java, caption=Java Lstlisting]
2 // Ich bin ein Kommentar!
3 public static void main(String[] args) {
4     System.out.println("Ich bin ein Array!")
5 }
6 \end {lstlisting}
```

Auflistung 4: Java Lstlisting

8.5.2 Minted

Benötigt die Option minted.

Umgebung

```
1 \begin {minted}[options]{java}
2 // Ich bin ein Kommentar!
3 public static void main(String[] args) {
4     System.out.println("Ich bin ein Array!")
5 }
6 \end {minted}
```

Auflistung 5: Minted Umgebung

Zeile

```
1 \mint[options]{lang}|code|
```

Auflistung 6: Minted Einzeiler

```
1 \mintinline[options]{lang}{code}
```

Auflistung 7: Minted Inline

Literaturverzeichnis

- [1] *Einführung in die Universelle Windows-Plattform - UWP app developer*. [Online; accessed 11. Apr. 2018]. 2018. URL: <https://docs.microsoft.com/de-de/windows/uwp/get-started/universal-application-platform-guide#writing-code>.

Abbildungsverzeichnis

1	Darstellung der UWP	4
2	Installieren von Emulatoren	5
3	Mit Beschreibung und Label	8

Tabellenverzeichnis

1	Tabular	9
2	TabularX	9

Auflistungsverzeichnis

1	makefig	8
2	vardef	8
3	Java Code	11
4	Java Lstlisting	11
5	Minted Umgebung	12
6	Minted Einzeiler	12
7	Minted Inline	12