

Maturaarbeit

Entwicklung einer Blackberry-Applikation

Gymnasium Thun-Schadau

Manuel Meinen 14gG

August 2013

Betreuende Lehrkraft: Martin Rathgeb

Typ der Maturaarbeit : Produkt mit Werkbericht



Abstract

Mit der Erfindung des Blackberrys im Jahre 1999 läutete der RIM-Gründer Mike Lazaridis ein neues Zeitalter ein, das Zeitalter des mobilen Computings. Noch nie in der Geschichte der Menschheit konnten sich die Menschen, welche für den wirtschaftlichen Fortbestand der Menschheit unabdingbar sind, besser organisieren als im Zeitalter des mobilen Computings. RIM, beziehungsweise Blackberry, war schon immer darauf bedacht die Produktivität seiner Kunden durch elektronische Organisations- und Kommunikationsgadgets zu steigern.

Nebst den sogenannten Super-Apps, wie zum Beispiel der Kalender-Applikation, der E-Mail-Applikation oder der Kontakte-Applikation, welche von Blackberry selber entwickelt wurden, bietet das Blackberry-Betriebssystem platz für Applikationen von Drittanbietern.

Genau mit diesem Punkt befasst sich meine Maturaarbeit. Meine Leitfrage lautet nämlich wie folgt:

Wie entwickle ich erfolgreich eine eigene Blackberry-Java-Applikation?

Unter diesem Motto begann ich mit der Programmierarbeit an einer Tourismus-Applikation für Thun. Meine Applikation trägt den Namen BBThun. Ähnlich wie Apple vor all seine Produkte einen "I" setzt, setzt Blackberry vor seine Produkte ein "BB", was eine Abkürzung für Blackberry ist.

BBThun stellt touristische Informationen zur Stadt Thun übersichtlich gegliedert dar. Die BBThun-Applikation beinhaltet folgende Kategorien:

- Sehenswürdigkeiten
- Museen und Galerien
- Zahlen und Fakten
- Geschichtliches
- Restaurants

Diese Arbeit gibt zum einen grundsätzliche Informationen über Blackberry, befasst sich aber grundsätzlich mit dem Entwicklungsprozess einer eigenen Applikation. Als konkrete Folge dieser Arbeit ist die Applikation BBThun entstanden.

Vorwort

Ich habe mich für dieses Maturaarbeitsthema entschieden, da ich zum einen ein absoluter Blackberry-Fan bin und zum anderen sehr an Programmierarbeit und grundsätzlich an Computertechnik interessiert bin. Die Frage, die mir am häufigsten gestellt wurde während der Zeit der Maturaarbeit lautete:

“Wieso Blackberry?“

Nun ich bin schon seit drei Jahren ein stolzer Besitzer eines Blackberry Torch 9800. Ich war schon vom ersten Moment an begeistert von den Blackberry-Produkten. Zudem besitze ich seit etwa vier Monaten ein Blackberry Playbook und war sofort auch von den auf QNX basierenden Blackberry-Produkten begeistert.

Aufgrund meiner Begeisterung für Blackberry und weil ich denke, dass Blackberry enormes Potenzial hat, habe ich mich dazu entschieden für Blackberry zu entwickeln.

Da ich erst seit einem Jahr im Informatikunterricht in der Programmiersprache Java unterrichtet werde und dadurch meine Programmierfähigkeiten ziemlich beschränkt sind, würde ich mich selbst als blutigen Programmieranfänger bezeichnen. Nichts desto trotz war ich erstaunt wie weit man mit meinen Programmierfähigkeiten und Dokumentationen aus dem Internet kommen kann. Deshalb möchte ich alle, die grundlegende Java-Kenntnisse haben, etwas logisch denken können und sich für Blackberry interessieren dazu ermutigen nicht nur davon zu träumen, was man alles machen könnte, sondern selbst Hand anzulegen und es einfach mal zu versuchen. Die Entwickler-Foren im Internet sind sehr hilfreich und die erfahrenen Entwickler sind oftmals sehr hilfsbereit und freuen sich, wenn junge Leute sich für solche Sachen interessieren.

Schlussendlich möchte ich mich noch bei allen bedanken, die mich bei meiner Maturaarbeit unterstützt haben. Zuerst möchte ich mich bei der Abteilung für Stadtmarketing der Stadt Thun bedanken, die mir grosszügigerweise sämtliche Inhalte für meine Applikation kostenlos zur Verfügung gestellt hat. Des weiteren bin ich sehr dankbar für all die hilfreichen Foren wie Stackoverflow, das Blackberry Supportforum oder das Crackberry Forum, die mir mit ihren Beiträgen enorm geholfen haben. Zudem bedanke ich mich recht herzlich bei all meinen Freunden und meiner Familie, die mich immer unterstützt haben wo sie konnten. Und nicht zuletzt bedanke ich mich bei meinem Betreuer Herrn Rathgeb, der mich bei meiner Maturaarbeit unterstützt und trotzdem nicht unter Druck gesetzt hat.

Inhaltsverzeichnis

1	Geschichtliches	1
1.1	Research In Motion (RIM)	1
1.2	Blackberry	2
1.3	Blackberry OS 5-7	3
1.4	Blackberry 10	3
2	Blackberry Entwicklungsumgebung	5
2.1	Blackberry Java Plug-in für Eclipse	5
2.2	Blackberry Software Development Kit (SDK)	5
2.3	Blackberry Simulator	5
3	Beispiel	6
3.1	Hello World	6
4	Ideenfindung	8
4.1	Informationssammlung	8
4.2	Strukturierung	8
5	Entwicklung meiner eigenen Applikation	9
5.1	Iterative Development	9
5.2	Testing	10
5.2.1	Auf dem Simulator	10
5.2.2	Auf meinem Blackberry	11
6	Vertrieb	12
6.1	Blackberry World	12
6.2	Developer Lizenz	12
6.3	Wartung	13
7	Verzeichnisse	14
7.1	Literaturverzeichnis	14
7.2	Internet Verzeichnis	14
7.3	Quellenverzeichnis der Abbildungen	14
8	Anhang	I
8.1	Glossar	I
9	Arbeitsprozessbericht	III
9.1	Erlernetes	III
9.2	Aufgetretene Schwierigkeiten	III
9.3	Verbesserungen	III

10 Eidesstattliche Erklärung

IV

1 Geschichtliches

1.1 Research In Motion (RIM)

Im Jahre 1984 gründete Mike Lazaridis, ein ehemaliger griechischstämmiger Student der University of Waterloo in Kanada, eine Firma namens Research In Motion (kurz RIM), welche den Verkauf von neuen Kommunikationsgeräten zum Ziel hatte. 1999 brachte RIM, mit dem Blackberry 850 (Siehe Abbildung 1.1), das erste Blackberry-



Smartphone auf den Markt. Das Gerät konnte jedoch nicht zur Telefonie verwendet werden. Trotzdem waren seine technischen Spezifikationen für seine Zeit bahnbrechend. Nebst den produktivitätssteigernden Grundfunktionen wie Kalender, Aufgabenliste und

Wecker verfügte das Blackberry 850 über einen E-Mail-Dienst. Somit konnte RIM einen grossen Teil von Geschäftskunden für sich gewinnen.

Als Apple im Jahre 2007 das iPhone 3 auf den Markt brachte, brachen für RIM schwierige Zeiten an. Smartphones waren nun keine reinen Geschäftstools mehr. Die Menschen wollten auf einmal Touchscreens anstelle von Hardwaretastaturen. Zudem boten damals die Blackberry Smartphones praktisch keine Unterhaltungsmöglichkeiten. Circa ein Jahr nach dem Release des iPhones 3 brachte RIM mit dem Blackberry Storm das erste Blackberry mit Touchscreen auf den Markt. Doch da war es schon zu spät. Viele hatten entweder schon ein iPhone oder wollten sich gerade eins kaufen weil ihre Bekannten auch schon eins hatten. Zudem war das Blackberry OS¹ 5 nicht gut genug an die Touchbedienung angepasst. So kam es, dass RIM immer mehr Marktanteile verlor.

Als wäre des Tragischen nicht schon genug gewesen, kam im

Jahre 2009 das mobile Betriebssystem Android auf den Markt. Somit war das Zeitalter des Ubiquitous Computings² definitiv angebrochen. Im Businessbereich war RIM nach wie vor führend. RIMs hohe Sicherheitsstandards und die komfortablen Administrationsmöglichkeiten machten die Blackberrys zu beliebten Businesssmartphones. Doch dann setzte ein weiterer Trend ein, welcher RIM zu schaffen machte. Immer



Abbildung 1.2: Logo von Research In Motion

¹Operating System (Betriebssystem)

²Rechnerallgegenwart

mehr Menschen hatten zwei Smartphones. Ein Blackberry für die Arbeit und ein iPhone oder ein Android Gerät fürs Privatleben. Das führte dazu, dass immer mehr Firmen den Mitarbeitern erlaubten ihr privates Smartphone anstelle des Businesssmartphones zu verwenden. Dieser Trend nennt man die BYOD³-Tendenz.

Am 30. Januar 2013, mit dem Release von Blackberry 10, gab RIM bekannt, dass RIM künftig Blackberry heißen werde. Blackberry 10 war eine komplette Neuentwicklung mit verschiedensten neuen Funktionen, unter anderem auch Blackberry Balance, welche dem BYOD-Trend angepasst war. Somit war es möglich auf einem Gerät Geschäftliches von Privatem sauber zu trennen.

1.2 Blackberry

Blackberry war von 1999 bis anfangs 2013 lediglich ein Produkt der Firma RIM. Seit dem 30. Januar 2013 ist Blackberry jedoch eine eigene Firma mit dem Sitz in Kanada. RIM wurde in Blackberry umbenannt, da die Marke Blackberry sehr bekannt war und einen guten Ruf hatte, wo hingegen die Firma RIM öfters Negativschlagzeilen machte da sie immer mehr Marktanteile verlor. Der Release von Blackberry 10 und der gleichzeitige Namenswechsel wurde als kompletter Neubeginn vermarktet. Blackberry OS 1-7 wurden von RIM selbst entwickelt. Daher waren sie alle auf einander aufgebaut und Applikationen waren meistens aufwärtskompatibel. Blackberry 10 wurde jedoch von QNX entwickelt, einer Firma, die von Blackberry aufgekauft worden war.

Im Jahre 2011 brachte RIM das erste Blackberry Tablet unter dem Namen Blackberry Playbook auf den Markt. Das Tablet wies ein neuartiges Betriebssystem namens Blackberry Tablet OS auf. Das Blackberry Tablet OS war das erste von QNX entwickelte Produkt, welches durch RIM vermarktet wurde. Das Blackberry Tablet OS diente später als Grundlage für das Blackberry 10 Betriebssystem. Das erkennt man daran, dass die Entwicklerversion von Blackberry 10, welche auf dem Entwicklergerät Blackberry 10 Dev Alpha lief, dem Blackberry Tablet OS extrem glich.



Abbildung 1.3: Blackberry Tablet OS



Abbildung 1.4: Blackberry 10 Dev Alpha

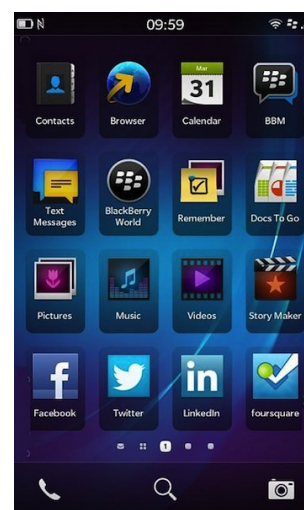


Abbildung 1.5: Blackberry 10

³Bring Your Own Device

1.3 Blackberry OS 5-7

Die Blackberry Betriebssysteme OS 5 bis 7 sind heutzutage bereits veraltet. Die aktuelle Betriebssystemversion ist Blackberry 10. Dennoch ist es so, dass die meisten Blackberry Besitzer ein Blackberry mit OS 5 bis 7 haben. Statistiken von Ende 2012 zeigen, dass lediglich 1.5% aller Blackberry Kunden ein Blackberry mit der Betriebssystemversion 4 besitzen. Wenn man nun eine Applikation für Blackberry OS 5 entwickelt, dann deckt man 98.5% aller Blackberry Kunden die nicht Blackberry 10 haben ab, da die API⁴ von Blackberry OS 5 bis OS 7 aufwärtskompatibel ist. Das heisst, wenn man eine Applikation für Blackberry OS 5 programmiert, dann läuft sie auch auf OS 6 und OS 7, da die API immer nur gewachsen ist und keine Funktionen verschwunden sind. Da Blackberry 10 eine Neuentwicklung ist, ist seine API grundlegend anders und nicht mehr mit den älteren APIs kompatibel.

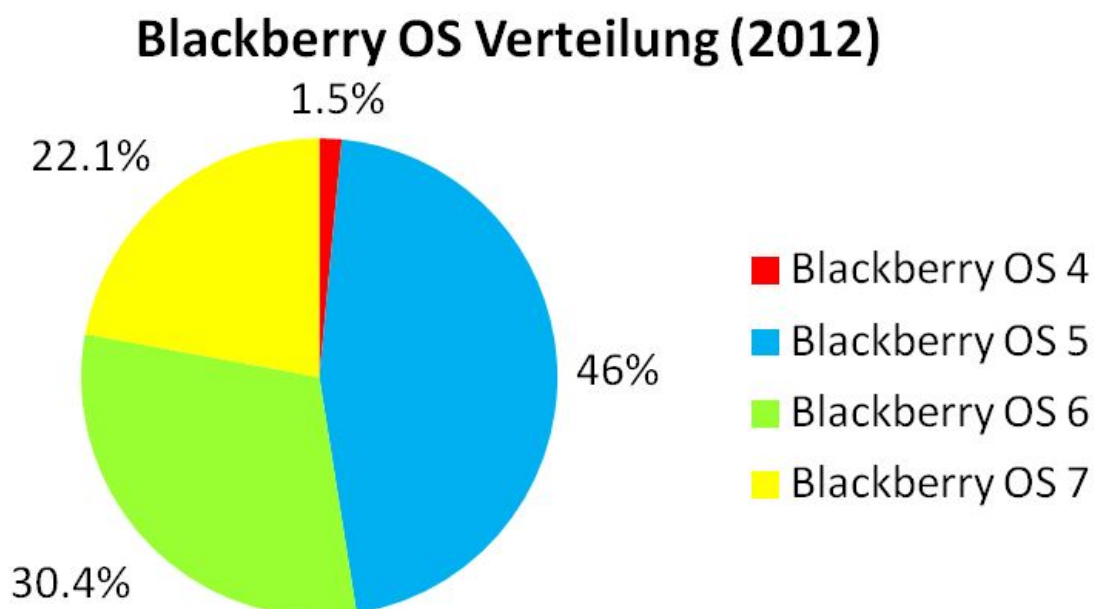


Abbildung 1.6: Prozentuale Verteilung der Betriebssystemversionen Stand 2012

1.4 Blackberry 10

Kurz nach dem Release des Blackberry Playbooks im Jahre 2011 wurde bekannt, dass die beiden Gründer Mike Lazaridis und Jim Balsillie, schon seit geraumer Zeit ein neues Betriebssystem am entwickeln waren, welches damals BBX genannt wurde. 2012 übernahm schliesslich der deutschstämmige Thorsten Heins das traditionsreiche Unternehmen und brachte es auf Vordermann. Er führte das Projekt, ein neues Betriebssystem basierend auf QNX zu entwickeln, fort. Am 30. Januar 2013 war es nun soweit. Thorsten Heins stellte in Toronto Blackberry 10 und zugleich zwei Geräte, welche auf Blackberry 10 laufen sollen, vor. Gleichzeitig wurde Blackberry 10 auch noch in New York City, London, Paris, Johannesburg und Dubai vorgestellt.

⁴Application Programming Interface (Programmierschnittstelle)

Blackberry 10 kam zuerst nur auf dem Fulltouch Smartphone namens Blackberry Z10 (Siehe Abbildung 1.7) heraus. Blackberry war sich jedoch bewusst, dass sich die Blackberry Kunden nicht mit einem Touchscreen zufrieden geben, denn ein Kaufargument für Blackberry Kunden ist die einzigartige Hardwaretastatur. Deshalb kündigte Thorsten Heins am selben Tag noch ein zweites Blackberry Gerät an, das Blackberry Q10 (Siehe Abbildung 1.8). Das Blackberry Q10 verfügt über einen kleinen, quadratischen Touchscreen und besitzt darunter eine vollwertige QWERTY⁵-Tastatur. Wie man sich das von Blackberry gewohnt ist, sind die beiden Smartphones qualitativ sehr hochstehend. Diese Qualität hat jedoch ihren Preis. Blackberry war sich aber bewusst, dass die Märkte der Zukunft die der Schwellen- oder Entwicklungsländern sind. Um diese neuen Märkte zu erobern entschloss sich Blackberry dazu ein günstiges Smartphone anzubieten. Das Smartphone nennt sich Blackberry Q5 (Siehe Abbildung 1.9) und ist von den Spezifikationen her ähnlich wie das Q10. Jedoch ist es billiger produziert worden, da es praktisch vollständig aus Kunststoff gefertigt wurde.

Ein günstiges Smartphone zu produzieren war für Blackberry in zweierlei Hinsicht ein cleverer Schachzug. Zum einen können sich so nun auch Menschen in Schwellenländern, wie Indien, ein Blackberry leisten und zum anderen gibt es immer mehr Menschen aus Schwellenländern, die sich dazu entschlossen Applikationen zu entwickeln, um so Geld zu verdienen. Wenn sie nun ein Blackberry Q5 haben und für dieses Applikationen entwickeln, dann können die Menschen aus Industrieländern, die sich ein Blackberry Q10 gekauft haben und bereit sind für Applikationen zu bezahlen, davon profitieren.



Abbildung 1.7: BlackBerry Z10



Abbildung 1.8: BlackBerry Q10



Abbildung 1.9: BlackBerry Q5

⁵QWERTY ist das Standard-Tastaturlayout in englischsprachigen Ländern

2 Blackberry Entwicklungsumgebung

2.1 Blackberry Java Plug-in für Eclipse

Blackberry stellt den Applikationenentwicklern ein Plug-in¹ zur Verfügung, welches dem Entwickler verschiedenste Tools bietet. Standardmässig wird dem Entwickler beim Download des Plug-ins, ein SDK² und ein Simulator installiert. Wenn man Eclipse³ noch nicht installiert hat, dann wird dieser gerade mit dem Plug-in mit installiert. Wenn man heute das Plug-in installiert, dann wird die SDK-Version 7.1 mit installiert, welche zur Entwicklung von Blackberry OS 7 Applikationen dient. Ältere SDK-Versionen können in Eclipse individuell nachinstalliert werden.

Eine Auswahl zu den SDKs passenden Blackberry Simulatoren wird mit dem Plug-in mit installiert. Weitere Simulatoren können von der Blackberry Website bezogen werden.

2.2 Blackberry Software Development Kit (SDK)

Die Blackberry SDKs werden benötigt um Applikationen für Blackberry Smartphones zu entwickeln und zu testen. Durch die Installation eines SDKs wird zum einen die von Blackberry bereitgestellte Klassenbibliothek in Eclipse integriert und zum anderen werden neue Menüpunkte kreiert, welche es zum Beispiel erlauben den Simulator von Eclipse aus zu starten oder zurückzusetzen.

2.3 Blackberry Simulator

Ein Blackberry Simulator ist ein Programm, welches das Ausführen von Applikationen auf einem virtuellen Gerät ermöglicht. Der Simulator bietet ein RTE⁴, welches die genau gleichen Bedingungen bietet wie ein reales Blackberry-Gerät. Eine Applikation, welche auf dem Simulator läuft, kann somit auch auf einem Blackberry-Gerät, welches auf die Standardeinstellungen zurückgesetzt wurde, zum Laufen gebracht werden. Der Simulator kann zudem virtuelle Netzwerke, eingehende und ausgehende Anrufe, sowie zahlreiche andere Ereignisse simulieren. Diese Eigenschaften machen den Blackberry Simulator zu einem Wertvollen Testing-tool.

¹Softwareteil, welcher in ein grösseres Programm eingefügt wird

²Software Development Kit

³Programmierprogramm

⁴Runtime Environment (Laufzeitumgebung)

3 Beispiel

3.1 Hello World

Wenn man beginnt eine neue Programmiersprache zu lernen oder wenn man sich in ein API einarbeiten will, dann wird einem meistens empfohlen als erstes eine Hello-World-Anwendung zu entwickeln. Diese Anwendung kann lediglich den Text *"Hello World!"* anzeigen. Der Sinn solch einer Anwendung ist, dass man mit den Grundprinzipien dieser Art der Programmierung vertraut wird. Denn in praktisch allen Anwendungen muss irgendwo ein Text angezeigt werden und wenn man *"Hello World!"* anzeigen kann, dann kann man auch jeden anderen Text anzeigen.

In der Applikationsentwicklung ist es grundsätzlich so, dass eine Applikation mindestens eine Application-Klasse und eine Screen-Klasse benötigt. Die Application-Klasse beinhaltet die main-Methode, welche ein neues Objekt der Application-Klasse instantiiert. Der Konstruktor der Application-Klasse wird somit ausgeführt und erzeugt einen neuen Screen aus der Screen-Klasse und zeigt diesen an. Dadurch wird der Konstruktor der Screen-Klasse ausgeführt und ein Text wird angezeigt.

Listing 3.1: HelloWorldApp.java

```
1 // Der Package-Name gibt an, wo die Java-Dateien im scr-Ordner
2 // abgespeichert wurden.
3 package com.ManuelMeinen;
4 // Der folgende Import ist notwendig um die UiApplication-Klasse
5 // zu erweitern.
6 import net.rim.device.api.ui.UiApplication;
7 // Die Klasse HelloWorldApp erweitert die UiApplication-Klasse.
8 public class HelloWorldApp extends UiApplication {
9     // Der Konstruktor erzeugt einen neuen Screen und zeigt
10    // diesen an.
11    public HelloWorldApp() {
12        pushScreen(new HelloWorldScreen());
13    }
14    // Die main-Methode wird als erstes ausgeführt und
15    // erzeugt eine neue Instanz der HelloWorldApp-Klasse.
16    public static void main(String[] args) {
17        HelloWorldApp myApp = new HelloWorldApp();
18        myApp.enterEventDispatcher();
19    }
20 }
```

Listing 3.2: HelloWorldScreen.java

```

1 package com.ManuelMeinen ;
2
3 import net.rim.device.api.ui.component.RichTextField ;
4 import net.rim.device.api.ui.container.MainScreen ;
5
6 public final class HelloWorldScreen extends MainScreen {
7 // Das in der HelloWorldApp-Klasse instantiierte
8 // HelloWorldScreen-Objekt fuehrt nun den folgenden Konstruktor
9 // aus .
10 public HelloWorldScreen () {
11 //Dem Screen wird ein Titel gesetzt .
12     setTitle ("Hello_World" );
13 // Dem Screen wird ein neues Textfeld
14 // hinzugefuegt welches den Text "Hello World!"
15 // anzeigt .
16     add (new RichTextField ("Hello_World!" ));
17 }
18 }

```

Führt man nun diese Applikation auf dem Blackberry Simulator aus, dann sieht man folgendes Ergebnis (Siehe Abbildung 3.1). Die Anwendung kann wie jede andere Blackberry-Anwendung auch, entweder durch Drücken des Return-Knopfes oder durch Drücken des Blackberry-Knopfes und anschliessendes Auswählen des Menüpunktes *Close* verlassen werden.

Möchte man nun diese Applikation auch auf einem realen Gerät testen, dann kann man dies mittels eines Programms namens Javaloader tun. Man muss dazu sein Blackberry per USB-Kabel mit dem Computer verbinden und kann dann so die COD¹-Datei auf das Blackberry laden. Der Javaloader hat jedoch keine grafische Oberfläche. Das heisst, dass er nur per Kommandozeile bedient werden kann. Wenn einem das zu umständlich ist, dann gibt es jedoch eine Alternative. Das Programm heisst JLGUI² und bietet eine grafische Oberfläche für den Javaloader. So kann die COD-Datei per Drag-And-Drop³ auf dem Blackberry installiert werden. Das Programm ist kostenlos erhältlich unter www.controlgenie.com/JLGUI.zip.

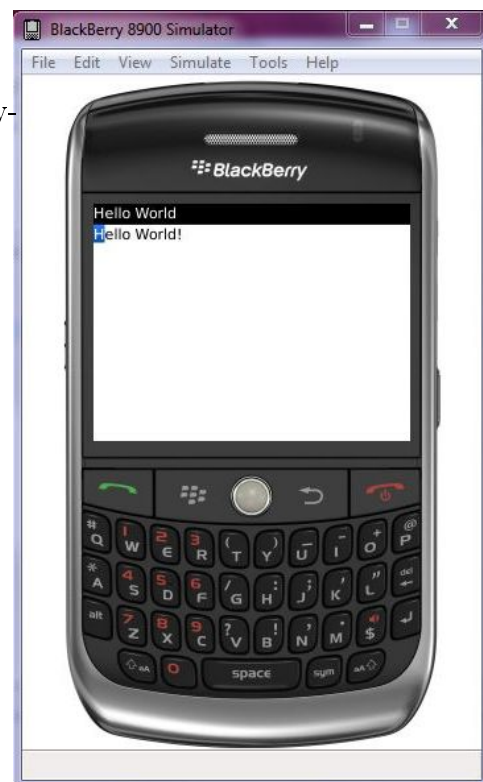


Abbildung 3.1: HelloWorld auf dem Blackberry Simulator

¹Für Blackberry ausführbares Applikationsformat

²Javaloader Graphical User Interface

³ziehen und ablegen

4 Ideenfindung

Ich entschloss mich als erstes grösseres Projekt eine Tourismus-Applikation für die Stadt Thun zu entwickeln. Eine solche Applikation hat verschiedene Vorteile. Eine Tourismus-Applikation ist für einen Programmieranfänger wie mich praktisch, weil man technisch gesehen ziemlich viele Freiheiten hat. Von einfacher Informationsdarstellung über GPS-Integration bis hin zu augmented reality¹ ist alles möglich. Applikationen, welche letztere zwei Punkte beinhalten, werden jedoch meistens nicht von Privatpersonen sondern von Firmen entwickelt. Aufgrund der zu kurzen Zeit und meiner mangelhaften Programmierfähigkeiten entschloss ich mich, mich lediglich auf eine strukturierte Informationsdarstellung zu beschränken.

4.1 Informationssammlung

Da meine Maturaarbeit eine technische und nicht eine historische Arbeit ist, habe ich mich dazu entschlossen sämtliche Informationen von offenen Quellen zu beziehen. Die Abteilung für Stadtmarketing hat mir freundlicherweise die Genehmigung erteilt, Informationen von ihrer Website (www.thun.ch) in meiner Applikation zu verwenden.

4.2 Strukturierung

Meine Applikation besteht aus zwei Klassen, welche in weitere Klassen erweitert werden. Zum einen beinhaltet meine Applikation die abstrakte UiApplication-Klasse. Da diese Klasse nicht instantiiert werden kann, muss sie immer erweitert werden. Zum anderen beinhaltet meine Applikation mehrere Screen-Klassen, welche jeweils von der Klasse mainScreen abgeleitet worden sind. Die mainScreen-Klasse ist zwar nicht abstrakt, trotzdem macht es keinen Sinn, sie direkt zu instantiiieren, da ein mainScreen nur beschränkte Möglichkeiten hat. Daher ist es sinnvoller neue Screen-Klassen zu erzeugen, welche die mainScreen-Klasse erweitern und genau diese Methoden enthalten, welche für den Screen relevant sind.

Grundsätzlich kann man meine Screen-Klassen in zwei Hauptgruppen unterteilen. Es gibt ButtonScreen-Klassen und InfoScreen-Klassen. Jedoch gibt es in einer Hauptgruppe mehrere Unterklassen. Da nicht alle InfoScreens das selbe Layout haben und nicht alle ButtonScreens Buttons, welche auf die selbe Art Screen verweisen, enthalten, ist es einfacher verschiedene ButtonScreen- beziehungsweise InfoScreen-Klassen zu erzeugen.

¹erweiterte Realität

5 Entwicklung meiner eigenen Applikation

5.1 Iterative Development

Unter dem Begriff Iterative Development versteht man eine Entwicklungsstrategie, welche eine schrittweise Entwicklung und Verbesserung der Software vorsieht. Das Softwareprojekt wird bei dieser Strategie in stets wiederkehrenden Zyklen entwickelt. Am Ende jedes Zyklus wird die Software auf ihre Funktionsfähigkeit geprüft. Sollte die Software am Ende eines Zyklus nicht funktionieren, dann weiss man genau in welchem Zyklus einem ein Fehler unterlaufen ist. Somit kann verhindert werden, dass die Applikation ein kompletter Misserfolg wird. Im schlimmsten Falle könnte man die Applikation einfach auf den Stand des letzten Zyklus zurücksetzen.

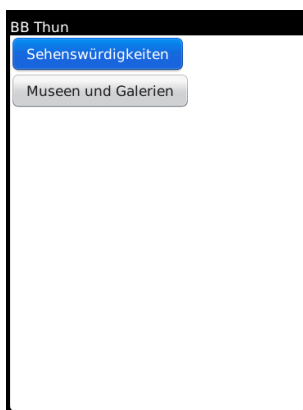


Abbildung 5.1: BBThun v1.0

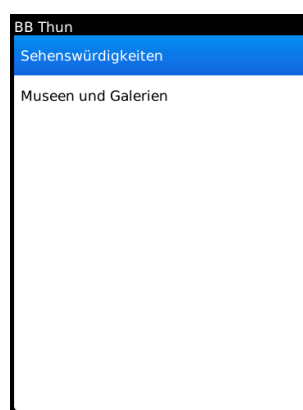


Abbildung 5.2: BBThun v1.1

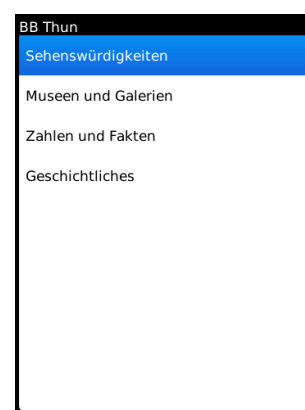


Abbildung 5.3: BBThun v1.2

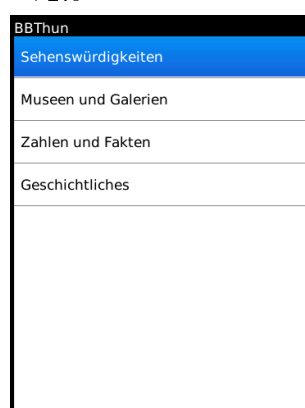


Abbildung 5.4: BBThun v1.4



Abbildung 5.5: BBThun v1.5

Die Abbildungen 5.1-5.5 zeigen den Main-Screen meiner Applikation am Ende eines Zyklus. In Abbildung 5.1 sind erst zwei Knöpfe zu erkennen, da die Applikation zu diesem Zeitpunkt erst zwei Kategorien hatte. Zudem haben die Knöpfe dort noch ein etwas unschönes Layout, da es sich bei diesen Knöpfen um Standard-ButtonField-Objekte handelt. In einer späteren Phase habe ich eine neue Art Knöpfe verwendet. Die Knöpfe der Version 1.1-1.5 nennt man ListStyleButtonField. Diese ListStyleButtonField-Klasse ist eine Erweiterung der Field-Klasse, welche in der Blackberry API nicht zu finden ist. Ein Open-Source-Programmierer hat diese Klasse geschrieben und sie im Internet zur Weiterverwendung bereitgestellt. Die Version 1.2 beinhaltet zudem gegenüber seiner Vorgängerversion zwei neue Kategorien. Ab der Version 1.4 haben die ListStyleButtonField-Knöpfe ein neues Layout erhalten. Zwischen jedem ListStyleButtonField habe ich ein SeparatorField eingefügt um die Felder optisch voneinander abzutrennen. Ein SeparatorField ist ein Feld, welches eine horizontale Linie zeichnet und den Fokus nie erhält. Dadurch ist es möglich mit einem Trackpad¹ durch die Knopfleiste zu navigieren, sodass immer nur die Knöpfe den Fokus erhalten.

5.2 Testing

Ein wichtiger Punkt in der Applikationsentwicklung ist das stetige Testen der Applikation auf ihre Funktionalität. Um die Anwendung zu testen gibt es zwei Möglichkeiten. Zum einen kann man die Applikation auf dem Blackberry-Simulator laufen lassen oder man kann sie auch auf ein reales Blackberry-Gerät laden. Beide Arten des Testings haben ihre Vorteile.

5.2.1 Auf dem Simulator

Das Testen auf dem Blackberry-Simulator bezweckt vor allem das Aufdecken von Syntax- oder Logikfehlern im Sourcecode. Wenn man in Eclipse eine Applikation ausführt, dann wird sie standardmässig kompiliert und auf den Simulator geladen. Beim Kompilieren können bereits erste Warnungen oder Fehlermeldungen angezeigt werden. Wenn man nun die Anwendung im Simulator startet, dann läuft diese in den meisten Fällen genau so wie sie auch auf einem realen Gerät laufen würde. Jedoch ist die Performance auf dem Simulator auch noch vom Computer abhängig und somit nicht sehr aussagekräftig.

¹Optischer Sensor zum Bedienen des Betriebssystems

5.2.2 Auf meinem Blackberry

Obschon das Testen auf dem Simulator fast immer dasselbe Ergebnis gibt wie auf einem realen Gerät, macht es dennoch Sinn, die Applikation hin und wieder auch auf einem realen Blackberry zu testen.

Bei meiner Applikation hatte ich zum Beispiel das Problem, dass ich die in Abbildung 5.6 gezeigte Fehlermeldung bekam, als ich eine Applikation auf meinem Blackberry laufen lassen wollte, nachdem sie auf dem Simulator ohne Probleme gelaufen war.

Die Fehlermeldung sagte mir, dass die Applikation versuchte auf eine gesicherte API zuzugreifen. Als ich nach dieser Fehlermeldung gegoogelt hatte, erfuhr ich, dass man Applikationen, welche auf gesicherte APIs zugreifen wollen mittels eines Signature-Tools verschlüsseln müsse. Um dieses Tool nutzen zu können, musste ich ein Blackberry Vendor-Portal² einrichten und Signing-Keys³ beantragen. Anschliessend musste ich die Signing-Keys installieren und konnte nun den Code verschlüsseln und auf meinem Blackberry ausführen.

Applikationen die auf heikle APIs zugreifen, müssen durch das Signature-Tool asymmetrisch verschlüsselt werden um zu verhindern, dass ein Hacker meine Applikation hackt und somit durch die gesicherten APIs meinem Gerät Schaden zufügen könnte. Da bei einer asymmetrischen Verschlüsselung der Entschlüsselungsschlüssel nicht effizient vom Verschlüsselungsschlüssel abgeleitet werden kann, kann ein Angreifer meine Applikation nicht hacken, selbst wenn er meinen Verschlüsselungsschlüssel hat. Die Verschlüsselung einer Applikation ist jedoch nicht nötig um sie auf dem Simulator auszuführen, da niemand einen Simulator hacken will.

Das Testen auf einem realen Gerät hat zudem den Vorteil, dass man so die Applikation besser auf Benutzerfreundlichkeit testen kann, da die Bedienung auf dem Simulator nicht genau der auf einem realen Gerät entspricht.

Ich kompiliere und teste meine Applikation immer auf dem Simulator wenn ich nur kleine Änderungen vornehme, da das Testen auf einem realen Gerät stets einen Zusatzaufwand bedeutet. Am Ende eines Entwicklungszyklus lade ich meine Applikation jeweils auf mein Blackberry um sicher zu gehen, dass alles gut funktioniert.

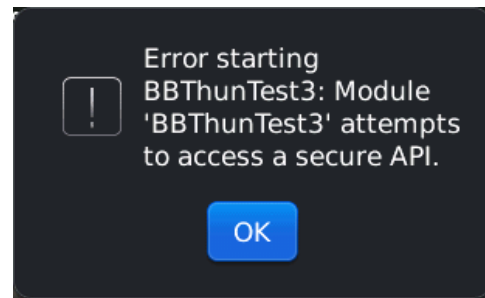


Abbildung 5.6: Fehlermeldung

²Benutzerkonto um Applikationen in der Blackberry World anbieten zu können

³Schlüsseldateien zur Verschlüsselung von Code

6 Vertrieb

6.1 Blackberry World

Die Blackberry World ist eine Plattform, welche den Verkauf und Vertrieb von Applikationen für Blackberry Smartphones und das Blackberry Playbook ermöglicht. Applikationen können auf drei verschiedene Arten angeboten werden. Entweder komplett kostenlos, kostenlos für eine bestimmte Probezeit oder kostenpflichtig. Bei der Variante mit einer Probezeit kann man zwischen 7 oder 30 Tagen Probezeit auswählen.

Für meine Applikation habe ich die kostenlose Variante gewählt. Zum einen weil ich so weniger Probleme mit dem Urheberrecht habe und zum anderen weil ich mir eine Abschreckung durch anfallende

Kosten bei den Kunden nicht leisten kann. Da ich sämtliche Texte von www.thun.ch kostenlos bezogen habe, wäre es nicht richtig für die Weiterverbreitung dieser Texte in meiner Applikation Geld zu verlangen. Zudem kann diese Applikation nur von deutschsprachigen Touristen, welche ein Blackberry mit Blackberry OS 5-7 besitzen genutzt werden. Da Thun nicht zu den bekanntesten Feriendestinationen der Schweiz gehört, ist das Kundensegment ziemlich beschränkt. Wenn ich nun pro Download noch einen Franken verlangen würde, dann würde wohl niemand mehr meine Applikation herunterladen. Obschon ein Franken nicht viel ist, haben dennoch viele Menschen Hemmungen für virtuelle Güter Geld auszugeben.



Abbildung 6.1: Logo von Blackberry World

6.2 Developer Lizenz

Anders als bei allen anderen mobilen Betriebssystem Herstellern ist die Registrierung, um Applikationen anbieten zu können, bei Blackberry komplett kostenlos. Apple verlangt von seinen Entwicklern pro Jahr 99 USD¹. Android und Microsoft verlangen nur einmalige Registrationsgebühren. Android verlangt 25 USD und Microsoft 99 USD. Wobei Microsoft seit dem Release von Windows Phone 8 immer wieder Sonderangebote hat, um Entwickler für sich zu gewinnen.

Um in der Blackberry World eine Applikation anbieten zu können, muss man sich ein

¹US-Dollar

Vendor Portal einrichten. Anschliessend muss man einige Informationen zur Applikation angeben und verschiedene Bilder hochladen. Bei den Informationen handelt es sich primär um den Namen der Applikation und eine Beschreibung in englischer Sprache. Optional kann auch noch eine anderssprachige Beschreibung angefügt werden. Zudem muss man auswählen für welche Betriebssystemversionen, Geräteversionen, Mobilfunkanbieter und für welche Länder die Applikation angeboten werden soll. Bei den Bildern, die man hochladen muss, handelt es sich um ein Icon und um ein sogenanntes Featured Image, was nichts weiter ist als ein Werbebild für die Applikation. Optional können noch Screenshots angefügt werden um einem potenziellen Kunden einen Einblick in die Applikation zu gewähren.

6.3 Wartung

Wenn man in der Blackberry World eine Applikation anbietet, dann hat man die Möglichkeit pro Applikation verschiedene Bundles² hinzuzufügen. In meinem Fall habe ich nur ein Bundle mit einer Datei darin. Sämtliche UI³-Komponente meiner Applikation füllen die ganze Breite des Bildschirms aus und haben somit keine feste Grösse. Das heisst, dass das gesamte Layout unabhängig vom Bildschirmformat ist. Zudem sind sämtliche Funktionen, auf welche meine Applikation zugreift, also Browser und Telefonfunktion, auf allen Blackberry Geräten verfügbar. Das heisst, dass ich nur ein Bundle brauche welches auf allen Blackberry OS 5-7 Geräten läuft. Dadurch ist es so, dass ich durch ein Update dieses Bundles die App-Versionen auf allen Geräten gleichzeitig ersetze.

Ich kann jedoch auch ein neues Bundle hinzufügen, welches nur für eine bestimmte Auswahl von Blackberry Geräten verfügbar ist. Ein Beispiel dafür wäre eine App-Version, welche eine GPS-Integration unterstützt. Falls ich dieses Vorhaben realisieren wollte, könnte ich einfach eine neue Version meiner Applikation entwickeln, welche diese Funktion unterstützen würde und dann hätte ich die Möglichkeit, diese neue App-Datei als neues Bundle hochzuladen. Dadurch wäre es mir möglich, durch die Auswahl einzelner Geräte, nur ein Update auf diese Geräte zu bringen, welche GPS unterstützen.

²Paket aller Dateien die für eine Applikation notwendig sind

³User Interface

7 Verzeichnisse

7.1 Literaturverzeichnis

Autor	Namen des Buches
Klaus Löwenhagen	Programmieren für den BlackBerry
Bill Foust	BlackBerry Java Application Development
Chris King	Advanced BlackBerry 6 Development
Karl G. Kowalski	BlackBerry Application Development for Dummies

7.2 Internet Verzeichnis

http://docs.blackberry.com/en/developers/?userType=21
http://www.blackberry.com/developers/docs/5.0.0api/
http://stackoverflow.com/
www.gieson.com
http://us.blackberry.com/
http://crackberry.com/
www.wikipedia.org

7.3 Quellenverzeichnis der Abbildungen

Abbildung	URL
Abbildung 1.1	http://news.softonic.de/
Abbildung 1.2	http://upload.wikimedia.org/wikipedia/de/0/09/Research_In_Motion.svg
Abbildung 1.5	http://i-cdn.phonearena.com/images/articles/73677-image/Screenshots-for-BlackBerry-10.jpg
Abbildung 1.4	http://www2.pcmag.com/media/images/288589-blackberry-10-dev-alpha.jpg
Abbildung 1.7	http://i.computer-bild.de/imgs/4/7/3/1/9/7/5/Smartphone-Blackberry-Z10-745x559-9ca2ffb81d61a26b.jpg
Abbildung 1.8	http://bestboyz.de/wp-content/uploads/2013/01/nseries_black_front-1.jpeg
Abbildung 1.9	http://www.connect.de/
Abbildung 6.1	http://www.pocketberry.com/wp-content/uploads/2011/11/blackberry-app-world-logo.png

Abbildungsverzeichnis

0.1	BBThun Icon	1
1.1	Erstes Blackberry Smartphone	1
1.2	Logo von Research In Motion	1
1.3	Blackberry Tablet OS	2
1.4	Blackberry 10 Dev Alpha	2
1.5	Blackberry 10	2
1.6	Prozentuale Verteilung der Betriebssystemversionen Stand 2012	3
1.7	Blackberry Z10	4
1.8	Blackberry Q10	4
1.9	Blackberry Q5	4
3.1	HelloWorld auf dem Blackberry Simulator	7
5.1	BBThun v1.0	9
5.2	BBThun v1.1	9
5.3	BBThun v1.2	9
5.4	BBThun v1.4	9
5.5	BBThun v1.5	9
5.6	Fehlermeldung	11
6.1	Logo von Blackberry World	12

8 Anhang

8.1 Glossar

RIM	Research In Motion
OS	Operating System (Betriebssystem)
BYOD	Bring Your Own Device
API	Application Programming Interface (Programmierschnittstelle)
SDK	Software Development Kit
RTE	Runtime Environment
COD	Für Blackberry ausführbares Applikationsformat
JLGUI	Javaloader Graphical User Interface (Grafische Benutzeroberfläche für den Javaloader)
UI	User Interface (Benutzeroberfläche)

Listings

3.1	HelloWorldApp.java	6
3.2	HelloWorldScreen.java	7

9 Arbeitsprozessbericht

9.1 Erlerntes

Durch diese Arbeit habe ich die grundlegenden Prinzipien der Applikationsentwicklung kennengelernt. Zudem gab es mir einen Einblick in die Blackberry-API und mir wurde bewusst, dass die Programmierarbeit trotz umfangreicher API und guter Dokumentation nicht ganz so einfach ist, wie ich mir das zuvor vorgestellt habe. Des Weiteren realisierte ich, wie einfachste Überlegungsfehler einen tagelang blockieren können, wo hingegen in Zeiten des allgemeinen Erfolges mir das Programmieren wie von selbst geschah. Dadurch wurde es sehr schwierig, mich an einen Zeitplan zu halten. Dennoch kam ich alles in allem mit der Programmierarbeit rechtzeitig zu einem Ziel.

9.2 Aufgetretene Schwierigkeiten

Aller Anfang ist schwer. So ist es auch in der Blackberry-Programmierung. Am Anfang hatte ich vor allem mit einfachen Aufgaben zu kämpfen wie etwa mit dem Einlesen von Textsegmenten aus Textdateien. Zudem musste ich mich an die in der Programmierwelt üblichen Begriffe gewöhnen. Es brauchte eine Weile bis ich verstanden hatte, was zum Beispiel ein FieldChangeListener macht. Des Weiteren hatte ich lange Mühe mit der Vorstellung, dass sämtliche UI-Komponente von der selben Klasse abgeleitet wurden. Daher dauerte es eine Weile, bis ich begriffen habe, dass zum Beispiel ein Bild, ein Knopf und ein Textfeld gewisse gemeinsame Eigenschaften haben, da sie alle Felder sind und somit von der Field-Klasse abgeleitet wurden.

9.3 Verbesserungen

Selbstverständlich ist meine Applikation noch nicht perfekt, so wie die wenigsten Applikationen bei ihrem ersten Release schon perfekt sind. Ich denke, das erste Update sollte vor allem ein schöneres Layout mit sich bringen und Bilder zu den Sehenswürdigkeiten. Denn von den Funktionen her ist meine Applikation schon nicht schlecht. Doch die Benutzeroberfläche sieht noch eher wie ein Forschungsprojekt aus als wie ein fertig entwickeltes Produkt.

10 Eidesstattliche Erklärung

Hiermit erkläre ich, die vorliegende Arbeit selbstständig und unter Angabe aller benötigten Quellen verfasst zu haben.

Datum und Unterschrift