

Fluggast-Informationen-App
Ein Multi-Plattform App-Projekt von A-Z

Florian F. Schmidt

© 2015 Florian Ferdinand Schmidt
2.Auflage 2017

Autor: Florian Ferdinand Schmidt, Traun
Lektorat, Korrektorat: Karin Schmidt, Traun
Weiterer Mitwirkender: Philip Rabler, Linz

Kommentare und Fragen können Sie gerne an mich richten:
E-Mail: mail@schmidt-florian.at

Druck und Vertrieb im Auftrag der Autorin/des Autors: Buchschmiede von Dataform Media GmbH,
Wien
www.buchschmiede.at
978-3-99049-799-9 (Paperback)
978-3-99049-800-2 (Hardcover)
978-3-99049-801-9 (e-Book)
Printed in Austria

Die Informationen in diesem Buch wurden mit größtmöglicher Sorgfalt erarbeitet. Dennoch können Fehler nicht vollständig ausgeschlossen werden. Verlag und Autor übernehmen keine juristische Verantwortung oder irgendeine Haftung für eventuell verbliebene Fehler und deren Folgen. Alle Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt und sind möglicherweise eingetragene Warenzeichen. Wir richten uns im wesentlichen nach den Schreibweisen der Hersteller. Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Alle Rechte vorbehalten einschließlich der Vervielfältigung, Übersetzung, Mikroverfilmung sowie Einspeicherung und Verarbeitung in elektronischen Systemen.

Inhaltsverzeichnis

Einleitung	iii
1. Vorwort	iii
2. Danksagung	iii
1. Allgemein	1
1.1. Begriffsdefinition App	1
1.2. Plattformen	1
2. Das Projekt	3
2.1. Projektablauf	3
2.2. Innovation	3
2.3. Mock Ups	3
2.4. Funktionen der App	3
2.5. Architektur	7
2.6. Zeitplan	8
3. Entwicklungsumgebungen	9
3.1. Betriebssystem	9
3.2. xCode	10
3.3. Android Studio	10
3.4. Visual Studio 2015	11
3.5. BlackBerry Packager	11
3.6. Adobe Creative Suite	11
3.7. Virtualisierung vs. Emulator	12
3.8. GIT und GitLab	13
3.9. Markdown	16
3.10. L ^A T _E X	17
4. Android Entwicklung	19
4.1. MVC	21
4.2. BlackBerry Portierung	24
4.3. Codedokumentation	25
5. iOS Entwicklung	31
5.1. MVC	33
5.2. Weitere Funktionen der IDE	34
5.3. Codedokumentation	35
6. Datenübertragung und Verarbeitung	37
6.1. Problemstellung	37
6.2. XML	41
6.3. JSON	42
6.4. REST	43
6.5. CSV	44

7. Lokalisierung	47
7.1. Zielgruppenanalyse	47
7.2. Lokalisierung vs. Übersetzung	48
7.3. Implementierung unter Android	48
7.4. Implementierung unter iOS	51
8. Design, UI und Tests	55
8.1. Do's and Don's	55
8.2. Beispiele für schlechtes UI	56
8.3. Softwaretest	57
8.4. Warum testen?	58
8.5. Tipps für UAT	59
9. Veröffentlichung	61
9.1. Abnahme	61
9.2. Der App Store	62
9.3. Google Play	68
9.4. Amazon App Shop	71
9.5. BlackBerry World	72
9.6. Vermarktung	73
10. Projektauswertung	75
A. Skizzen Anfang 2014	77
B. Sonstiges	79
Literaturverzeichnis	81

Einleitung

1. Vorwort

Dieses Buch basiert auf meiner Diplomarbeit aus dem Jahre 2015. Sämtliche Kapitel in diesem Werk lehnen sich an dem Beispiel einer „Kunden (Passagier)- Informations- App“ eines österreichischen Regionalflughafens an. Wenn Sie die App zu der ursprünglichen Diplomarbeit ausprobieren möchten, finden Sie hierzu einen QR-Code (Abbildung 1) weiter unten.

Was ist dieses Buch und was ist es nicht?

Dieses Buch stellt eine grobe Zusammenfassung bzw. Checkliste dar, wie Sie eine App konzipieren, entwickeln, veröffentlichen und vermarkten (**können**)! Es beinhaltet viele Tipps und Hinweise und ist gerade für Neulinge in diesem Gebiet hilfreich, damit diese erfassen können, worauf man bei solchen Projekten achten muss.

Es wurde in vier Teile gegliedert:

- Projektplanung
- Implementierung
- Vorbereitung und Durchführung der Veröffentlichung sowie Marketingmaßnahmen
- Projektauswertung

Es wurde bewusst auf Programmieranteile verzichtet und nur die wichtigsten Dinge werden in Beispielen hervorgehoben. Daher richtet sich dieses Werk vorrangig an Programmierer mit Erfahrung in objektorientierter Programmiersprachen.

2. Danksagung

An dieser Stelle möchte ich mich bei all jenen Personen bedanken, durch welche ich dieses Buch überhaupt erst in dieser Form verwirklichen konnte: DI (FH) Markus Kugler, Ingo Hagedorn, Christoph Kraska, Sepp Hohner, Gerhard Duda, Karin und Harald Schmidt, und natürlich allen anderen, die ich hier leider nicht erwähnen konnte.



Abbildung 1.: QR-Code auf die App

1. Allgemein

1.1. Was ist eine App?

Diese Frage ist gar nicht so leicht zu beantworten. „App“ ist die Kurzform von Application. Application bedeutet Anwendung. Also ist eine App eine Computeranwendung.

Der Duden geht bei diesem Begriff schon eher auf die aktuelle Bedeutung ein:

„(...) zusätzliche Applikation, die auf bestimmte Mobiltelefone heruntergeladen werden kann. [5]“

Aber selbst diese Definition muss in Zukunft wahrscheinlich geändert werden. Es ist davon auszugehen, dass sich Apps weiter auf „Smart Devices“ ausdehnen werden. Unter „Smart Devices“ versteht man alle vernetzten Gadgets wie Smartphones, Tablets, Smart Watches, digitale Brillen usw [32].

Jedoch ist auch anzumerken, dass der Begriff „App“ auch zunehmend wieder für Anwendungen auf klassischen PCs Verwendung findet (vgl. Mac App Store und Microsoft Windows 10 [4])

Abschließend kann man jedoch behaupten, dass „App“ derzeit primär eine speziell für Mobiltelefone angepasste Anwendung beschreibt.

1.2. Android, BlackBerry, iOS, Windows Phone, ...

Waren früher am Markt hauptsächlich Nokia und Siemens (Mobile) vertreten, so kämpfen heute eine Unzahl von Herstellern um die Gunst der Kunden. Trotzdem haben sich hier zwei bis drei große Softwareplattformen etabliert: Android, iOS und Windows Phone.

Android (griechisch androides, menschenähnlich) wurde 2005 von Google gekauft und wird seitdem gemeinsam mit der „Open Handset Alliance“ weiterentwickelt. Es basiert auf dem Linux Kernel und verwendet bis Version 5 die Dalvik Engine als VM für die Ausführung von Apps und seit Version 5 ART (Android Runtime). Android steht für verschiedene Prozessorarchitekturen (u.a. x86 und ARM) zur Verfügung.

Google hat als Programmiersprache eine dem Java Syntax identische Sprache entwickelt. Daher können auch sehr viele Java Klassen benutzt werden und Java Entwickler können sehr leicht Anwendungen für Android entwickeln. Die meisten Schnittstellen hat Google jedoch aus Geschwindigkeitsgründen in C(++) programmiert.

Ebenfalls kann man mit dem Native Development Kit auch in C(++) geschwindigkeitskritische Anwendungen entwickeln [26].

iOS

iOS basiert auf Apples Unixoiden Betriebssystem OS X. Sie teilen sich den Darwin- Kernel sowie mehrere untere Schichten. Als GUI nutzt iOS jedoch Cocoa Touch statt Cocoa.

Als (Haupt-)Programmiersprachen werden Objective-C und Swift genutzt. Objective- C ist eine

strikte Obermenge zu C. Das heißt, man kann im Code C und Objective-C fast beliebig mischen. Man kann sogar jede beliebige Programmiersprache nutzen (Es gibt sogar C# IDEs [34])

Obwohl Apple bereits im Jahr 2015 die Version 2.x von Swift veröffentlicht hat (und unter die Apache-2.0-Lizenz stellte), werden 2016 immer noch viele Projekte mit Objective- C entwickelt.

Der Sourcecode wird beim Building beider Sprachen in Maschinensprache kompiliert.

BlackBerry

BlackBerry hat zwar einen Marktanteil von unter 5 % [18], ist jedoch gerade im Businessbereich (noch) sehr stark vertreten. Die aktuelle Plattform (BlackBerry 10) unterstützt neben nativen BlackBerry Apps (geschrieben in C(++)) und Web- (HTML5 abkHTMLHypertext Markup Language) Apps auch Android Apps (durch Nutzung eines Android Subsystems). Seit kurzem ist es sogar möglich, mit dem BlackBerry BAR Packager (siehe Kapitel 4.2 BlackBerry Portierung auf Seite 24). APK Dateien zu konvertieren, signieren und in der BlackBerry World offiziell einzureichen [11].

Windows Phone

Da Microsoft 2014 das Kerngeschäft von Nokia kaufte, wurde Microsoft im Mobilen Bereich konkurrenzfähig gegenüber Android und iOS. Die Benutzeroberfläche von Windows Phone ist in sogenannten Hubs unterteilt. Diese Hubs stellen die Basis für weitere Funktionen dar, die direkt in das Betriebssystem integriert wird. Sie lassen sich auch als „Live-Kacheln“ als interaktive Kacheln auf dem Startbildschirm einbinden [23].

Windows Phone	Kernel
Windows Phone 7	Windows CE - Kernel
Windows Phone 8/ 10	Windows NT - Kernel

2. Das Projekt

2.1. Wie läuft das Projekt ab?

“ Das Geheimnis des Erfolges ist die Beständigkeit des Ziels.

BENJAMIN DISRAELI (1804-1881)

2.2. Innovationen

Wichtig an erfolgreichen App-Projekten ist deren innovativer Charakter. Zum Zeitpunkt der Veröffentlichung der App gab es z.B. in Österreich keinen einzigen Regionalflughafen (der Flughafen Wien hat bereits seit einiger Zeit eine eigene App) mit einer eigenen App. Bis zu diesem Zeitpunkt konnten unsere Kunden nur schwierig mit ihren Smartphones die normale Website nutzen, um z.B. den aktuellen Flugplan abzurufen.

2.3. Mock Ups - Erste Skizzen

Für die Präsentation von Anwendungen eignen sich Mock-Ups (Skizzen) besonders. Sie stellen die geplanten Funktionalitäten dar, ohne auf das UI, Design und die Plattform zu stark einzugehen. Sie finden im Anhang [A Skizzen Anfang 2014](#) auf Seite 77 ein Beispiel Mock-Up der App.

2.4. Funktionen der App

Flugplan

Der Flugplan stellt das zentrale Feature der App dar. Dieser wird nach dem Öffnen der App automatisch für den aktuellen Tag in Form einer Liste angezeigt. Unser Passagier kann innerhalb weniger Sekunden auf diese Informationen zugreifen. Durch ein Auswahlménü (Navigation Drawer oder Section Header) kann zwischen den Tagen umgeschaltet werden. Um weitere Informationen anzuzeigen, wählt der Passagier einen Flug aus. In der Detailansicht wird ein Flugzeugbild, das aktuelle Wetter am jeweils anderen Flughafen, sowie die Prognose für 3 Tage dargestellt. Ebenfalls ist ein Share Dialog für die „Sozialen Netze“ implementiert.

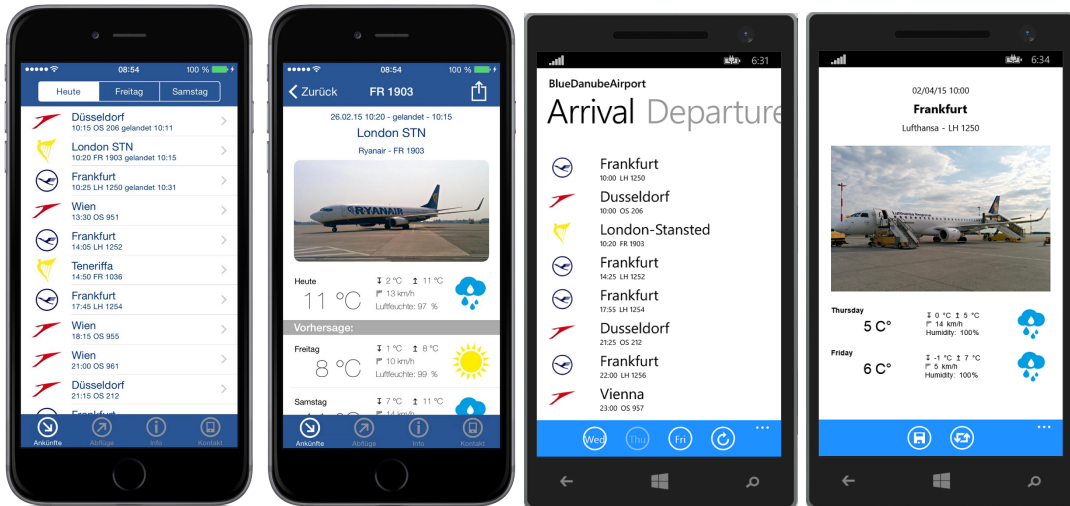


Abbildung 2.1.: Flugplan und Detailansicht

Info-Bereich

Der Info-Bereich ist ein digitaler Informationsschalter! Er bildet im Großen und Ganzen die Inhalte der Homepage ab, jedoch mit kurzen, für Smartphonedisplays optimierten Texten. Er ist in folgende Hauptkategorien eingeteilt:

- Am Flughafen
Bank, Gastronomie, Duty Free, Informationsschalter, Kinder, Konferenzräume, Reisezone
- Anreise
Autovermietung, Bahnverbindung, Busverbindung, Flughafentransfer, Taxi
- Check-In
Allgemein, Check-In Zeiten, Flüssigkeitsbestimmungen, Notpass, Sondergepäck, verbotene Gegenstände, Vorabend Check-In
- Parken
Allgemein, Parkplan, Parktarife, Tarifrrechner
- Tourismus

Diese Daten werden aus mehreren CSV Dateien in den lokalen Speicher in Form eines Dictionary-Trees (siehe Abbildung 2.2) geladen. Im Info-Bereich sind diese Informationen dann in Listenform und mittels Suchfunktion abrufbar. Je nach Typ wird bei Auswahl eines Begriffes eine Detailseite mit Text (und Bild), eine lokale oder entfernte PDF Datei oder eine Website angezeigt. Ebenfalls kann dort durch Action-Buttons die entsprechende Abteilung angerufen, gemailt oder die Website im Browser geöffnet werden.

Wetter und Webcam

Hier wird im oberen Bereich die Webcam des Aprons angezeigt. Diese aktualisiert sich alle fünf Sekunden selbstständig. Unterhalb wird eine Liste mit der Wetterprognose für die nächsten 7 Tage am Flughafen dargestellt.

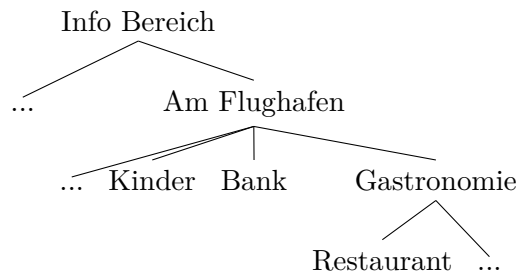


Abbildung 2.2.: Baumstruktur

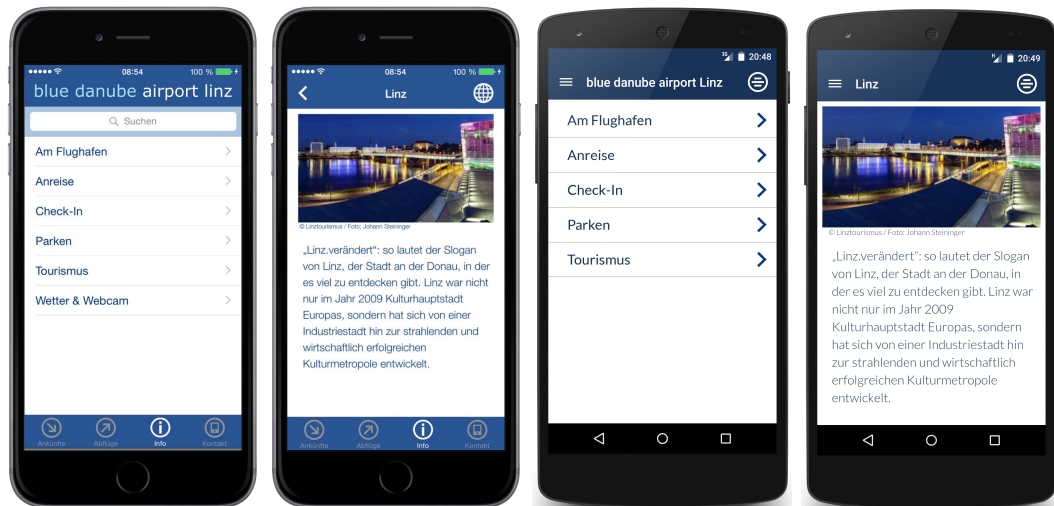


Abbildung 2.3.: Flughafeninformation und Detailansicht

Kontakt

Über die Seite Kontakt kann schnell und unkompliziert mit dem Flughafen kommuniziert werden. Neben einem Routenplan sind auch Buttons für Telefon, Facebook, YouTube, Google+ sowie Feedback vorhanden.

Sprachen

Die Anwendung ist in 3 Sprachen lokalisiert: Deutsch, Englisch und Tschechisch. Mehr hierzu in Kapitel 7 [Lokalisierung](#) auf Seite 47.

Einstellungen und Impressum

Je nach Plattform sind Einstellungen (FaceBook Login) und Impressum/ Datenschutzzinformationen innerhalb App oder in der Apple Einstellungen App vorhanden.

2. DAS PROJEKT



Abbildung 2.4.: Webcam und Wetter

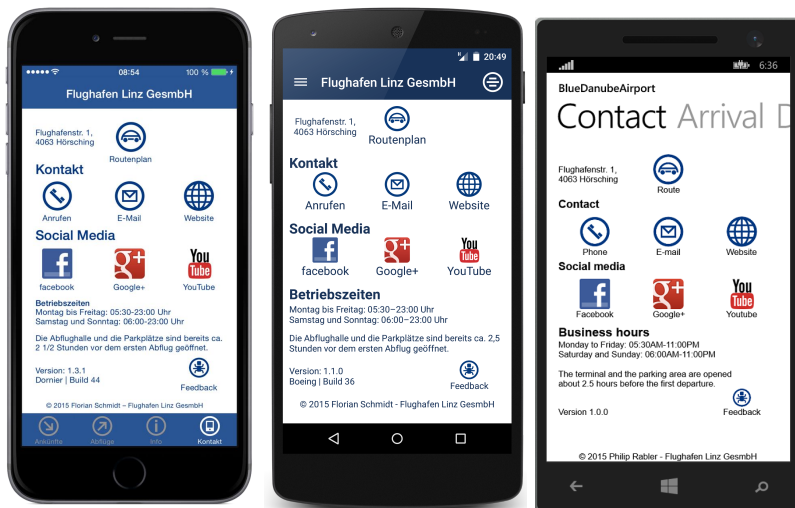


Abbildung 2.5.: Kontakt

Unterstützte Plattformen

Zu den unterstützten Betriebssystemen zählen iOS 7+, Android 4.0.3+ sowie BlackBerry 10.2.1+.

Einschränkungen

Da gerade auf der Android Plattform die Streuung der Hardware-/Softwarekombination sehr groß sein kann, ist es nicht möglich, eine 100% Kompatibilität mit allen Geräten zu gewährleisten.

Die Wetterdaten kommen von einem kostenlosen Anbieter. Dieser garantiert eine 95 % Verfügbarkeit p.a.. Sollte dieser Dienst in Zukunft kostenpflichtig oder eingestellt werden, muss auf Alternativen ausgewichen werden.



BlackBerry Z30

2.5. Architektur

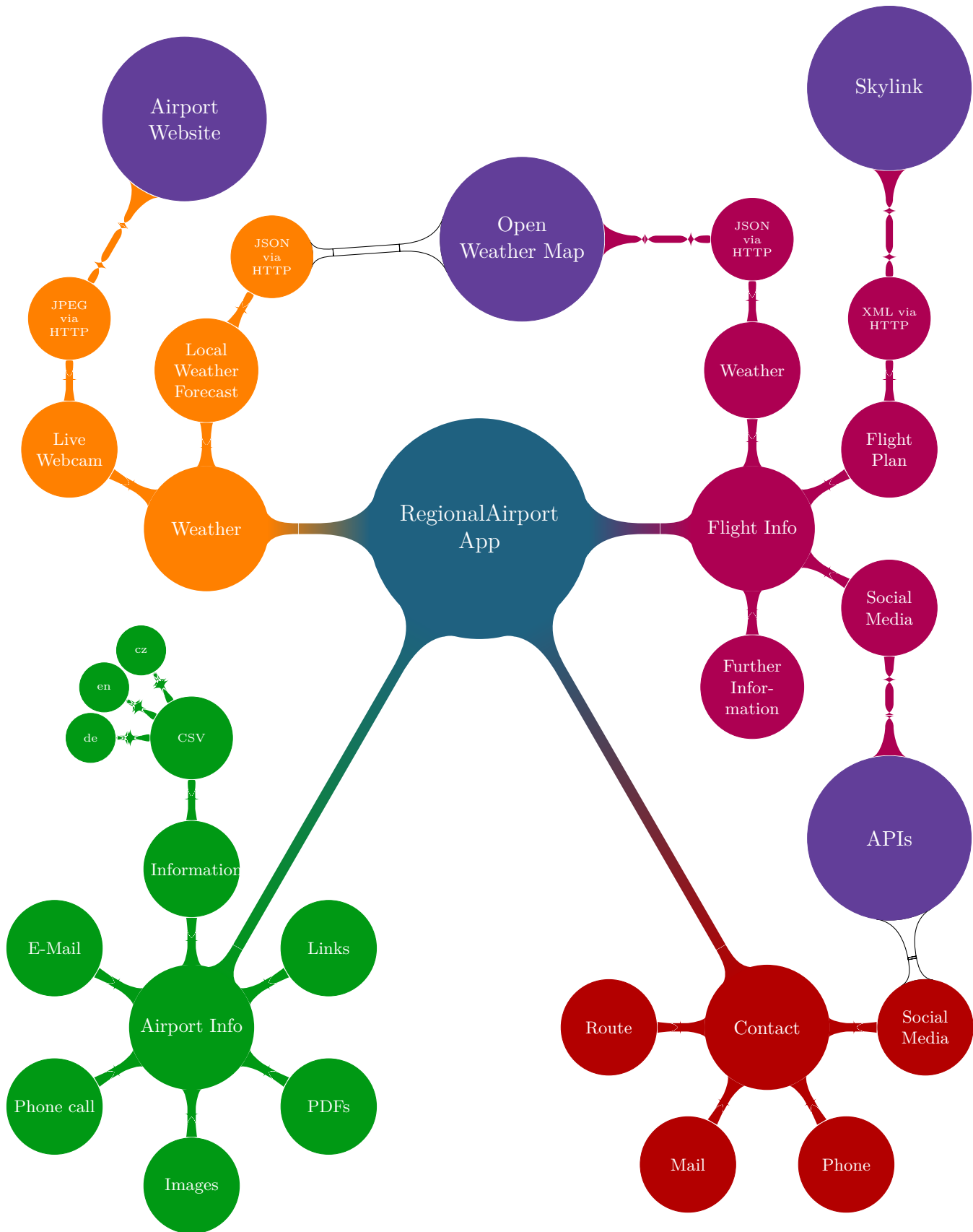


Abbildung 2.6.: Architektur der App

2.6. Projektzeitplan

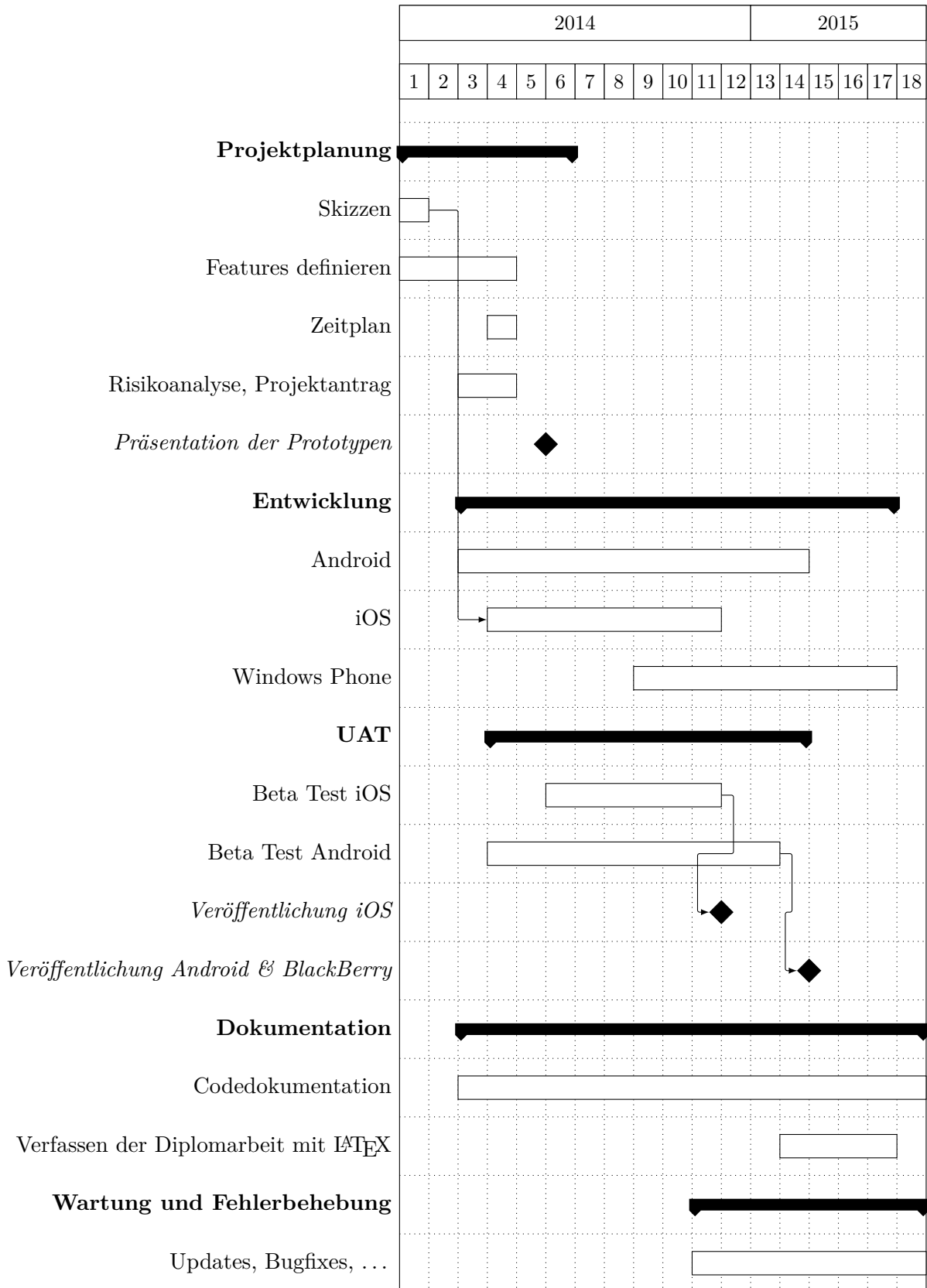


Abbildung 2.7.: Projektzeitplan Linz Airport App

3. Wie ist unser Entwickler-PC konfiguriert?

Um eine gute App zu entwickeln, reicht eine IDE alleine nicht aus. Man benötigt eine Vielzahl von Programmen, angefangen vom kleinen Tool bis zur großen Softwaresuite. In diesem Kapitel finden Sie eine kurze Beschreibung der wichtigsten Werkzeuge.

3.1. Welche(s) Betriebssystem(e) benötige ich?

Diese Frage kann man nur mit: "Kommt d'rauf an" beantworten: Wenn Sie Apps für iOS entwickeln möchten, kommen Sie an Apples OS X in der aktuellsten Version nicht vorbei. Um dieses Betriebssystem (legal und sinnvoll) nutzen zu können, benötigen Sie einen Mac (Apple-Computer).

Sollten Sie zusätzlich für Windows (Phone) entwickeln, ist Windows 8.1 oder 10 Pro (in einer virtuellen Maschine) nötig. Eine VM ist auch hilfreich, wenn sie ein spezielles Windows Tool nutzen.

Um ausschließlich für Windows (Phone) und/ oder Android zu entwickeln, reicht ein aktueller Windows PC.

Tabelle 3.1 gibt eine Hilfestellung bei der Wahl des Betriebssystems:

Betriebssystem	iOS	Android	Win Phone	BlackBerry
OS X	ja	ja	ja (mit VM)	ja
Windows 7	nein	ja	nein	ja
Windows 8 / 8.1	nein	ja	eingeschränkt	ja
Windows 8.1 Pro	nein	ja	ja	ja
Windows 10 Pro	nein	ja	ja	ja

Tabelle 3.1.: Plattformverfügbarkeit von IDEs

* Der Windows Phone Emulator ist nur unter Prozessoren mit Intel VT-x und Windows 8.1 bzw. 10 Pro verfügbar (Hyper-V).

Wie nutzt man Windows optimal auf einem Mac?

Unter Intel-Macs gibt es zwei Möglichkeiten, Windows auszuführen:

- Nativ mittels Apple BootCamp

Jeder Intel Mac hat das BootCamp Dienstprogramm installiert. Dieses partitioniert die Festplatte und erstellt eine Treiber CD oder einen Treiber USB-Stick. Danach kann man Windows durch Neustarten des Systems und Öffnen des Betriebssystem-Auswahlmenüs (ALT Taste sofort nach dem Einschalten gedrückt halten) starten.

Vorteil: Maximale Geschwindigkeit, da Windows direkt auf die Hardware und den maximalen Arbeitsspeicher zugreifen kann.

Nachteile: Neustart nötig, Datenaustausch komplizierter

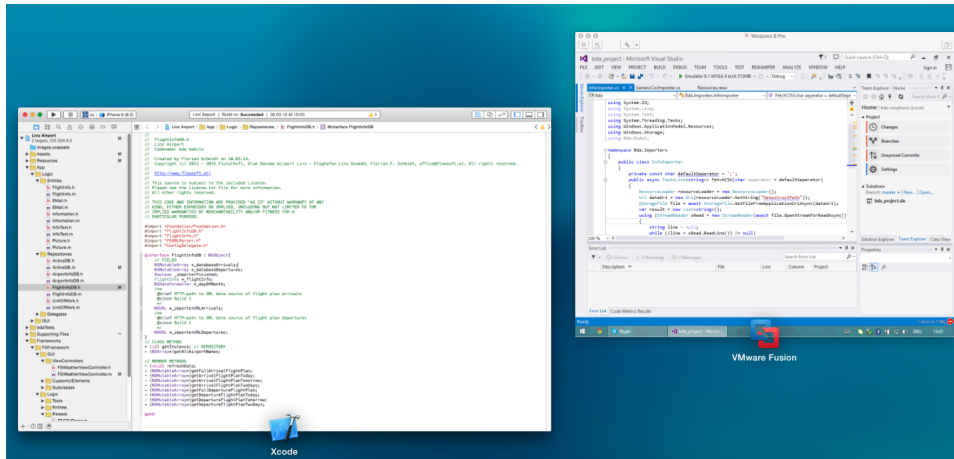


Abbildung 3.1.: VMWare Fusion

- Mittels virtuellen Maschine

Mithilfe von VM-Software wie „VMWare Fusion“ (siehe Abbildung 3.1) oder „Parallels Desktop for Mac“, kann man während des Betriebs von OS X zwischen beiden oder mehreren Betriebssystemen ohne Neustart wechseln. Beide Programme beherrschen „Drag and Drop“ und „Copy and Paste“ zwischen dem Host und den Gastsystemen.

Vorteile: Es ist die Daten- und Informationsübertragung zwischen mehreren Betriebssystemen sehr leicht und schnell. Windows Programme können sich unter OS X integrieren
 Nachteile: Trotz Intel VT-x ist (je nach Anwendungsfall) ein Leistungsverlust möglich.
 Doppelter Arbeitsspeicherbedarf (zwei Betriebssysteme laufen zeitgleich)

Weitere Informationen über Virtualisierung und Emulation finden sie am Ende dieses Kapitels unter [Virtualisierung vs. Emulator](#) auf Seite 12.

3.2. Apple xCode

xCode ist die Entwicklungsumgebung von Apple. Mit dieser kann man unter anderem Anwendungen für OS X, iOS, die Kommandozeile (auch für Linux/ Unix) und Webanwendungen entwickeln. Die Software kann über den Mac App Store¹ kostenlos heruntergeladen werden. Da der Download ca. 2,5 GB groß ist, sollte er über eine Breitbandverbindung erfolgen. In xCode ist bereits ein Client für GIT (siehe Abschnitt [3.8 GIT und GitLab](#) auf Seite 13) und Subversion integriert.

3.3. IntelliJ Android Studio

Am 8. Dezember 2014 wurde von Google die finale Version 1.0 von Android Studio veröffentlicht (Download und Installation siehe Abschnitt [4 Android Entwicklung](#) auf Seite 19). Seit diesem Zeitpunkt ist es die empfohlene Entwicklungsumgebung von Android, da in Zukunft die Unterstützung des Android SDK für Eclipse eingestellt wird. Eine der größten Neuerungen, neben der stark verbesserten IntelliSense, stellt das Build-Tool Gradle dar. Hiermit werden immer die aktuellsten Pakete beim Builden heruntergeladen [13]. Inzwischen wurde von Google Bereits Version 2.0 veröffentlicht.

¹<https://itunes.apple.com/at/app/xcode/id497799835?mt=12>

3.4. Microsoft Visual Studio 2015 + Windows Phone SDK

Visual Studio stellt die Standardentwicklungsumgebung für Windows Phone 8.1 / 10 dar. Bei der Installation muss darauf geachtet werden, dass das Windows Phone Paket mitinstalliert wird. Seit Ende 2014 kann Microsoft Visual Studio Community 2015 für Studenten, Schüler, Forscher, Lehrer, Trainer sowie Individualentwickler und kleine Entwicklerstudios bis 5 Mitarbeiter kostenlos heruntergeladen² kommerziell und nichtkommerziell genutzt werden [3].

3.5. BlackBerry Packager

Um Apps für BlackBerry 10 zu veröffentlichen, muss man nur das AndroidStudio Plugin, sowie den Simulator von BlackBerry herunterladen³. Für den Simulator muss VMWare Fusion installiert sein. Informationen zur Verwendung finden Sie im Abschnitt [4.2 BlackBerry Portierung](#) auf Seite [24](#).

3.6. Adobe Creative Suite

Die Adobe Creative Suite bzw. Creative Cloud stellen die Standardanwendungen in Bildbearbeitung, Layout und Design dar. Jedes Bild sollte vor der Einbindung in die App mit Adobe Photoshop optimiert werden. Bildbearbeitung beinhaltet unter anderem nachfolgende Punkte [21]:

- Den richtigen Ausschnitt wählen
Was ist interessant an dem Bild und was nicht? Will man alles präsentieren?
- Das richtige Seitenverhältnis
Am besten 10:15 oder 16:9 (Standards). Kameras unterschiedlicher Hersteller machen oft unterschiedliche Seitenverhältnisse. Kontrollieren, damit das Bild nicht gestreckt, gezerzt oder mit Rändern versehen wird.
- Die richtige Größe
Es macht keinen Sinn, wenn das Bild mit 5000px Breite und (10 MB Speicherbedarf) gespeichert wird und dann nur 200 px breit am Display angezeigt wird. Dies verursacht zusätzliche Ladezeit und unnötigen Speicherbedarf. Ebenfalls darf ein Launchscreen auf einem Retinadisplay nicht im Code hochskaliert werden!
- Das richtige Dateiformat
Je nach Anwendungsfall benötigt man JPEG, PNG oder vielleicht sogar Vektorgrafiken.
- Optimierung
Hier nachschärfen, dort mehr Kontrast und auch den Rotstich entfernen. Oft machen solche Kleinigkeiten einen Riesenunterschied!

Um die „Produktfotos“ in den Stores etwas interessanter zu gestalten, kann man mit Adobe Illustrator Frames (Screenshots in Telefonfotos integrieren) mit Marketingtexten versehen (siehe Abbildung [9.2](#) auf Seite [70](#)).

²<http://go.microsoft.com/fwlink/?LinkId=517284>

³<https://developer.blackberry.com/android/tools/>

3.7. Virtualisierung vs. Emulator

Als Emulator wird in der Computertechnik ein System bezeichnet, das ein System in bestimmten Teilaspekten nachbildet. Software-Emulatoren sind Programme, die einen Computer nachbilden und es so ermöglichen, Software für diesen Computer auf einem Computer mit einer anderen Architektur zu verwenden. So können z. B. Spiele für ältere Spielekonsolen auf einem PC oder einer neueren Spielekonsole ablaufen. Ein weiteres Beispiel: Ein Softwareentwickler kann bei der Entwicklung eines Programmes für ein Gerät (z. B. ein Handy), das eine andere Architektur als der Entwicklungs-Rechner hat, dieses im Emulator testen und korrigieren, ohne es jedes Mal auf das Gerät kopieren zu müssen.[\[22\]](#)

Emulator für mobile Anwendungen

Einer der wichtigsten Aspekte beim Programmieren für Apps sind die Handy Emulatoren. Durch sie wird es erst möglich, ein Programm so auszutesten, wie es mit einem realen Gerät kaum möglich wäre. Die Vorteile der Emulatoren sind:

- schnellere Durchführung des Programms bei Windows und iOS
- Geschwindigkeiten des Internets zu manipulieren
- verschiedene Lokalisierungen zu testen.
- andere Modelle bzw. andere Bildschirmauflösungen können getestet werden

Diese und einige mehr Vorteile bietet ein Emulator. Natürlich gibt es nicht nur Vorteile, sondern auch Nachteile.

Zum Beispiel:

- es fallen Kosten bei diversen Plattformen an.
- langsamere Durchführung, beispielsweise bei Android
- Kein 100 % er Ersatz für physische Geräte
- zeitweise werden falsche Informationen beim Debuggen angezeigt.
- Speicherbelastung beim Computer

Wenn man Apps in einem App - Store geben möchte, sollte man trotz des Testens mittels Emulatoren auch noch reelle Geräte zu sich nehmen, da virtuelle Handys keinen Schutz vor Fehler garantieren. Auch wenn das Programm beispielsweise bei einem Emulator funktioniert, heißt es nicht, dass es auch wirklich auf einem echten Handy ohne Probleme läuft.

Virtuelle Maschinen (VM)

Eine virtuelle Maschine ist in der Informatik eine Nachbildung eines Rechensystems. Die Schicht zwischen dem Rechner und der VM wird Hypervisor oder Virtual Machine Monitor genannt und seine Implementierung erfolgt über verschiedene Arten:

- rein hardwarebasierend
- rein softwarebasierend
- Kombination von hard - softwarebasierend

Der Hypervisor erlaubt in der Regel mehrere virtuelle Maschinen auf einem physischen Rechner. Systembasierende virtuelle Maschinen bilden einen Rechner so vollständig nach, dass sich Betriebssysteme genauso wie auf einem realen PC ausführen lassen z.B Hyper-V von Microsoft. Im

Gegensatz zu systembasierenten VM werden bei prozessbasierenten VM nur einzelne Programme ausgeführt. Die Vorteile von virtuellen Maschinen gegenüber realen Maschinen sind:

- unterschiedliche Betriebssysteme können gleichzeitig auf einer physischen Maschine betrieben werden.
- Man braucht nicht für jedes System eine eigene Hardware
- Es können Betriebssysteme ausgeführt werden, die alleine mit der Hardware nicht funktionieren würden.

Aber nicht nur Vorteile bietet eine VM, sondern auch Nachteile, wie z.B.:

- Angriffsmöglichkeiten von Viren und Malware wird erhöht, da ein neues Ziel namens Hypervisor anvisiert werden kann.
- Mögliche Probleme mit der Lizenzierung von Betriebssysteme (OS X darf nur auf Apple Rechnern in Virtuelle Maschinen ausgeführt werden)

Tipp beim Benutzen von Hyper-V in virtuellen Maschinen

Um den Windows Phone Emulator unter Microsoft Visual Studio 2015 nutzen zu können, benötigen Sie neben Windows in der Pro Version (nur hier ist Hyper-V integriert) einen Prozessor, der Hardwarevirtualisierung unterstützt. Bei der Ausführung innerhalb einer virtuellen Maschine erhalten Sie eine Fehlermeldung des Dienstes Microsoft Hyper-V. Die Intel VT-x Technologie ermöglicht jedoch die Kaskadisierung innerhalb von VMs.

Diese Option muss in VMWare Fusion extra aktiviert werden. Hierzu wird im Paket der VM ein Parameter in einer Konfigurationsdatei hinzugefügt (siehe 3.1).

```
1 [...]
2 hypervisor.cpuid.v0 = "FALSE"
3 [...]
```

Quellcode 3.1: NameDerVM.vmwarevm/NameDerVM.vmx

3.8. GIT und GitLab

Was ist GIT

GIT (dt. Blödmann) ist eine freie Software zur verteilten Versionsverwaltung [27]. Damit ist man in der Lage, die Änderungen im Code nachzufolgen (Diff), mit anderen Entwicklern leichter zusammenzuarbeiten und im Falle von Fehlern zu den letzten Commits zurückzukehren (Reset oder Rebase). Ebenfalls hat man Backups des gesamten Sourcecodes. Um Dinge „auszuprobieren“, kann man einen Fork/Branch abspalten und später die Änderungen in den Master zurückführen (Merge).



Git

Installation

In OS X (mit installiertem xCode) und den meisten Linux Distributionen gehört Git zum Lieferumfang. Um Git unter Windows zu nutzen, muss man Zusatzsoftware installieren.

Ein praktisches GUI- Werkzeug von Git ist SourceTree von Atlassian. Es ist OpenSource Software und kann für OS X und Windows von <http://www.sourcetreeapp.com/> kostenlos heruntergeladen werden.

Verwendung von Git

Um Git zu bedienen, gibt es mehrere Möglichkeiten. Entweder man nutzt die Kommandozeile, die Git-Integration innerhalb der IDE oder eine GUI für Git. Eine Zusammenfassung der wichtigsten Befehle für Git finden Sie auf den nächsten Seiten [19].

```

1 $ mkdir projectname # Projektordner anlegen
2 $ cd projectname   # in diesen Ordner wechseln
3 $ git init         # git repository anlegen
4
5 # Alle Dateien in diesem Verzeichnis und Unterverzeichnisse zum Repository
  hinzufügen:
6 $ git add .
7
8 # Änderungen commiten (festschreiben) mit der Meldung Init commit:
9 $ git commit -m 'Init commit'
```

Quellcode 3.2: GIT Repository anlegen und erster Commit

```

1 $ cd projectname   # in den Projektordner wechseln
2
3 # git repository mit remote verbinden:
4 $ git remote add origin https://www.example.com/user/repo.git
5
6 $ git push # Alle Commits an den Remote übermitteln
7 $ git pull # Alle Commits vom Remote auf den lokalen Rechner holen
8 $ git fetch # Wie pull nur zusätzlich mit merge
```

Quellcode 3.3: GIT Repository mit Remote-Repository verbinden

```

1 $ mkdir projectname # Projektordner anlegen
2
3 # git repository von remote klonen:
4 $ git clone https://www.example.com/user/repo.git
```

Quellcode 3.4: GIT Repository von Remote-Repository klonen

Git vergibt jedem Commit eine eindeutige Commit-ID. Diese wird genutzt, um z.B. Commits rückgängig zu machen:

```

1 $ cd projectname # in den Projektordner wechseln
2
3 $ git reset e912073d # Commit rückgängig machen:
```

Quellcode 3.5: GIT Commit rückgängig machen

```

1 # date.pdf aus allen bisherigen commits entfernen
2 $ git filter-branch -f --index-filter 'git rm --cached --ignore-unmatch datei.pdf'
3
4 $ git push -u origin master -f # Push auf den Server erzwingen
5
6 $ git status # Änderungen am Arbeitsverzeichnis anzeigen
7
8 $ git commit --amend # Änderungen an den letzten Commit "anhängen"
9
10 $ git clean -f # Löscht alle ungetrackten Dateien aus dem Arbeitsverzeichnis
```

Quellcode 3.6: GIT weitere Befehle

Die Verwendung innerhalb der IDEs wird in den jeweiligen Kapiteln erörtert.