CharBuilder
Mobile Applikationen

Fachhochschule Bielefeld
Campus Minden
Studiengang Informatik

Beteiligte Personen:

Name	Matrikelnummer
Christopher Kluck	1078455
Philipp Clausing	1078231

---

16. Januar 2018

Inhaltsverzeichnis

1 Einleitung

In diesem Dokument wird die Entwicklung der Applikation CharBuilder für das Wahlpflichtfach Mobile Applikationen dargestellt. Die Aufgabenstellung hat uns sowohl die Wahl der Plattform als auch die Art der App offengelassen. Da wir in der Vorlesung die Grundlagen der App-Entwicklung

2 Stand der Technik

Im nachfolgenden wird erläutert welche Technologien wir für die Erstellung der CharBuilder App verwendet und warum wir uns für die entschieden haben. Hierbei haben wir den Text in die Unterkategorien Android(Betriebssystem), Kotlin(Programmiersprache), Gson(Json Java-Lib), Gradle(Build Tool), Android Studio(IDEA) aufgeteilt, dies sind die wichtigsten Technologien die wird nutzen um unsere App zu entwickeln.

2.1 Android

Wie in der Einleitung geschrieben haben wir uns für Android als Betriebssystem entschieden. Da verschiedene Version dessen auf unterschiedlichen Geräten verteilt sind muss man sich als Entwickler darüber hinaus auch für ein Mindest-API-Level entscheiden. Dies stellt die niedrigste Android Version dar, unter welcher die App ausgeführt werden kann. Bei der Wahl des API-Levels spielen verschiedene Faktoren eine Rolle, unter anderem wie viele der Android Geräte welche Version des Betriebssystems ausführen oder ob der Entwickler Funktionen nutzen möchte die erst ab einer bestimmten Version verfügbar sind.
Wir haben uns entschieden, unsere Applikation lediglich für Android Betriebssysteme der Version 5.0(API Level 21) oder höher zu entwickeln. Die Version 5 ist bereits am 12.11.2014 veröffentlicht worden und bringt einen großen Wandel in der Benutzeroberfläche durch Googles Material Design. Dieses ist an den Gestaltungsstil "Flat Designängelehnt und minimalistisch gehalten. Die Entscheidung trafen wir aufgrund der weitreichenden Änderungen in dieser Version, des bereits 3 Jahre in der Vergangenheit liegenden Veröffentlichungsdatums und der breiten Verteilung von 80,7% aller Android Geräte.

2.2 Kotlin

Kotlin ist eine neue Programmiersprache von JetBrains aus dem Jahr 2016. Wie auch Java kompiliert Kotlin zu JVM Bytecode. Es lässt sich dadurch sehr gut in das bestehende Java Ökosystem einbinden und kann alle Java Bibliotheken verwenden. Seit dem 17. Mai 2017 ist Kotlin eine von Android offiziel unterstützte Sprache.
Beide Teammitglieder haben zuvor noch nicht mit Kotlin gearbeitet, konnten sich jedoch aufgrund der Ähnlichkeit zu Java schnell zurechtfinden und gute Erfahrungen sammeln. Die Programmiersprache bietet dem Programmierer viele Vorteile wodurch sie immer beliebter wird unter Androidentwicklern. Der weitreichenste Vorteil ist es null-Referenzen zu verhindern, die Sprache bietet einem "Null Safety". Eine weitere Änderung ist auch die Möglichkeit Datenklassen zu erstellen, diese sind Klassen welche lediglich Daten halten und keinerlei Methoden selbst implementieren. Ein Beispiel dieser kann man im Kapitel Implementierung finden. Der Großteil der Appentwicklung geschah mit Kotlin v1.15 später

wurde dann ein Update auf Kotlin v1.2 durchgeführt. Dies liegt an dem Entwicklungszyklus von Kotlin, so wurde immer mit dem aktuellsten stabilen Release gearbeitet.

2.3 Gson

Gson ist eine Java Bibliothek zur Serialization und Deserialization. Sie wird genutzt um Java Objekte in JSON umzuwandeln oder auch JSON zu Java Objekten zu wandeln. Hierbei ist die Möglichkeit Generics zu verwenden äußerst wichtig, da diese in der CharBuilder Applikation mehrmals zum Einsatz kommt. Wir haben diese Bibliothek ausgesucht da sie uns bereits bekannt war von früheren Projekten und wir positive Erfahrungen gemacht haben. Desweiteren wird die Software bereits seit 2008 entwickelt und konnte seitdem durch viele Revisionen und Verbesserungen überzeugen. Für die App verwendet wir die Gson Version 2.8.2.

2.4 Gradle

Gradle ist ein weit verbreitetes Build-Tool welches automatisiert arbeitet und unterstützung für mehrere verschiedene Sprachen bietet. Es wird typischerweise für Android Applikationen verwendet welche mit Android Studio entwickelt werden, da das Tool tief in die Entwicklungsumgebung integriert ist. Zu unserer Entscheidung diese Programm zu werden kann nicht viel gesagt werden, da es wie oben erwähnt bereits integriert war und eine alternative neben großem Mehraufwand keine Vorzüge geboten hätte.

2.5 Android Studio

Android Studio ist die offizielle Entwicklungsumgebung für native Androidprogrammierung. Es bietet neben den bekannten IDE Funktionen wie Syntax Highlighting, Autovervollständigung, Instant Run(Es wird nur der veränderte Teil neu kompiliert) und Debugger auch einen Android Emulator um verschiedene Geräte und Android Version zu simulieren. Die IDE basiert auf IntelliJ's Softwareprodukten. Mit der Version 3.0.1 wurde die Entwicklungsumgebung für Kotlin angepasst, dies ist auch die Version welche wir zur Entwicklung unserer App verwendeten. Dabei liefert die neue Version ebenfalls ein Programm um Java-Code direkt in Kotlin-Code umzuwandeln, dies erleichtert dem Entwickler anfangs den Einstieg sollte im späteren verlauf aber nur selten genutzt werden um eine einheitliche Codequalität zu erreichen.

3 Anforderungen

3.1 Idee

Die App stellt einen digitalen Pen und Paper Charakterbogen dar. Es wird möglich sein, neue Charaktere anzulegen und die bereits bestehenden zu verwalten. Der Nutzer soll außerdem die Möglichkeit haben, seine Charaktere auf andere Geräte zu übertragen. Zu Beginn werden wir uns auf das „Star Wars : Am Rande des Imperiums"- Regelwerk konzentrieren.

3.2 Must have

Unter “Must have“ ist jede Funktionalität aufgelistet welche **unbedingt** umgesetzt werden muss um das Projekt als erfolgreich bezeichnen zu können. Diese Kategorie wird oft auch “Minimale Anforderungen“ genannt.

- Erstellung eines neuen Charakters
- Löschen eines bestehenden Charakters
- Wechsel zwischen bereits angelegten Charakteren
- Änderungen an Charakteren(Talentsteigerungen, etc.)
- Würfeltool
- Regeltexte,die bei der Erstellung helfen
- PDF Export

3.3 Should have

Die Kategorie “Should have“ fasst all die Punkte unter sich zusammen welche für ein erfolgreiches Projekt nicht unbedingt erforderlich sind, aber dennoch ein wichtiger Bestandteil sein können. Als Beispiel sei eine Funktionalität zu sehen welche das Produkt lediglich erweitert, von welcher die Grundfunktionalitäten allerdings nicht abhängig sind.

- Importieren und Exportieren von Charakteren in JSON
- Verlauf von Abenteuern, die ein Charakter schon bestanden hat
- Orte/Länder, die ein Charakter schon besucht/bereist hat.
- Möglichkeit, die Charaktere in der Google Cloud zu speichern um von den anderen Geräten darauf zugreifen zu können
- Option die generierten Dokumente auszudrucken

3.4 Could have

Features welche unter “Could have“ aufgelistet sind, werden nur umgesetzt wenn alle Punkte unter “Must have“ und “Should have“ bereits abgearbeitet sind. Sie sind vollkommen optional.

- Möglichkeit Charaktere für andere Regelwerke zu erstellen
- Vorgefertigte Charaktere

3.5 Won't have

"Won't have" beinhaltet jene Punkte welche **nicht** umgesetzt werden. Sie wurden von den Entwicklern oder Auftraggebern zu Projektspezifikationbeginn ausgeschlossen.

- Gruppenverwaltung(Möglichkeit für Spielleiter, seine Gruppenmitglieder einzusehen
- Spielbare Abenteuer

4 Architektur

5 Implementierung

6 Test und Usability

7 Zusammenfassung