

Projekt Mastermind



Dokument-Name	Pflichtenheft - Mastermind
Version	2.0
Erstellt am	12.03.2017
Autoren	Maximilian Bauknecht
Ablage	/home/max/Nextcloud/Dokumente/DEKRA/Theorie/Projekt MasterMind/SRS.odt

Team Bezeichnung	Gruppe A
Mitglieder	Anja Ohlhäuser Philipp Holzner Philippe Käufer Maximilian Bauknecht

Änderungshistorie

Version	Datum	Autor	Änderungsgrund
1.0	12.03.2017	Bauknecht	Erstellung
1.0	21.03.2017	Bauknecht	Freigabe durch Auftraggeber
1.1	03.04.2017	Bauknecht	Hinzufügen UseCase Diagramme
2.0	17.05.2017	Bauknecht	Testplan und Klassenmodell

Inhaltsverzeichnis

1.Nicht-Funktionale Anforderungen.....	4
1.1.Verfügbarkeit.....	4
1.2.Performance.....	4
1.3.Ergonomie.....	4
1.4.Wartbarkeit.....	4
2.Funktionale Anforderungen.....	5
2.1.UseCase Übersicht.....	5
2.2.UseCase Diagramm.....	6
2.3.UseCase Details.....	7
3.Endgültiges Klassenmodell.....	17
4.Priorisierung.....	18
5.Testplan.....	18
5.1.Testbasis.....	18
5.2.Teststrategie.....	18
5.2.1.Äquivalenzklassenbildung:.....	19
5.2.2.Grenzwertanalyse:.....	20
5.2.3.Error Guessing:.....	20
5.3.Testabschlusskriterien.....	20
5.4.Testplan.....	20
5.5. Fehlermanagement.....	22
6.Code Example.....	22
6.1.Klassen:.....	22
6.2.Funktionen:.....	23
6.3.Einstellungen:.....	24
7.Deployment Vorgaben.....	24

Abkürzungs.erzeichnis

App: Eine Applikation (kurz App). In diesem Projekt eine Programm für das Betriebssystem Android.

APK Datei: Datei in der sich die Daten der App befinden. Mit ihr wird die App auf dem Android installiert.

OOP: Objektorientierte Programmierung

1. Nicht-Funktionale Anforderungen

1.1. Verfügbarkeit

1. Zur App Verfügbarkeit werden folgende Aussagen getroffen:
2. Die App startet bei Aktivierung durch den Benutzer und hat keine Einschränkungen in Abhängigkeit der Uhrzeit.
3. Es wird keine Verbindung zu Netzwerkdiensten benötigt. Dadurch werden keine Wartungsfenster benötigt.
4. Die App wird für Android Geräte mit der Android Version 4.1 bis 7.1 und einer Displayauflösung von 480x800 bis 720x1280 Pixeln optimiert. Abweichend von diesen Parametern wird keine Kompatibilität gewährleistet.

1.2. Performance

Es werden folgende Performance Kategorien definiert. Diese beziehen sich immer auf die Zeit, die zwischen Eingabe des Benutzers und Rückmeldung der App verstreichen darf.

1. <1s
2. 1s - 5s
3. >5s

Die Zuordnung zwischen Usecases und Performancekategorien erfolgt in Anlage 1.

1.3. Ergonomie

Zur Bedienbarkeit der App werden folgende Kriterien festgelegt:

1. Jede Funktion der App ist mit maximal 3 Benutzerinteraktionen erreichbar.
2. Die App ist mit einem Finger bedienbar. Eine Multitouch-Unterstützung ist nicht geplant.
3. Die Bedienung der Benutzeroberfläche ist selbsterklärend gestaltet (Sprechende Bezeichnungen der Schaltflächen)
4. Zu jeder Zeit kann eine Bedienungsanleitung geöffnet werden.
5. Die Hauptfunktionen sind ergonomisch angeordnet.

1.4. Wartbarkeit

Folgende Vereinbarungen werden zur Wartbarkeit der App getroffen:

1. Die App wird nach den Prinzipien der OOP geschrieben.
2. Die Methoden und Klassen werden mithilfe von JavaDoc kommentiert und dokumentiert.

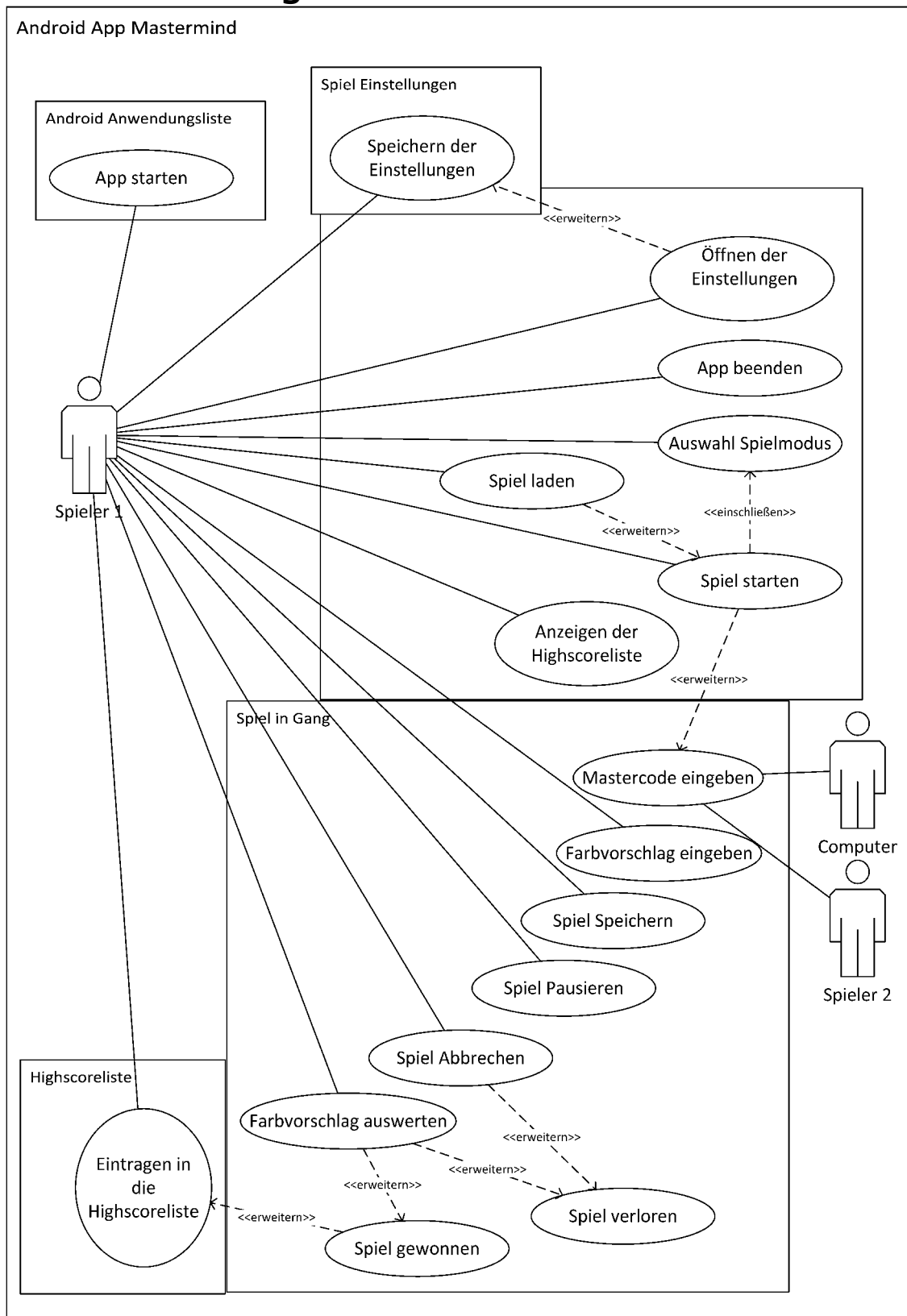
3. Die App kann durch eine neue APK Datei aktualisiert werden. Es gibt keine In-App Update Funktionalität.

2. Funktionale Anforderungen

2.1. UseCase Übersicht

Nr.	Titel	Performance-Klasse	Priorität
1	Appstart	3	2
2	Auswahl Spielmodus	1	3
3	Öffnen der Einstellungen	1	3
4	Speichern der Einstellungen	2	3
5	Spiel starten	2	2
6	Mastercode eingeben	1	4
7	Farbvorschlag eingeben	2	2
8	Farbvorschlag auswerten	3	3
9	Spiel pausieren	2	3
10	Spiel abbrechen	2	2
11	Spiel speichern	3	3
12	Spiel laden	3	3
13	Aktion Rückgängig machen	2	4
14	Spiel gewonnen	2	2
15	Spiel verloren	2	2
16	Eintrag in Highscore Liste	2	4
17	Anzeigen der Highscore Liste	2	3
18	App beenden	3	3

2.2. UseCase Diagramm



2.3. UseCase Details

Nr.	1
Titel	Appstart
Kurzbeschreibung	Benutzer startet die App
Akteure	Smartphone Benutzer
Auslösendes Ereignis	Klick auf App Icon im Launcher
Vorbedingungen	Keine
Nachbedingungen	Keine
Performanceklasse	3
Priorisierung	2
Ablauf	Benutzer klick auf App Icon. Ladescreen erscheint. Hauptmenü erscheint.

Nr.	2
Titel	Auswahl Spielmodus
Kurzbeschreibung	Der Benutzer wählt den Spielmodus (Gegen Computer oder Mensch spielen)
Akteure	Benutzer
Auslösendes Ereignis	Keines
Vorbedingungen	Hauptmenü erscheint
Nachbedingungen	Keine
Performanceklasse	1
Priorisierung	3
Ablauf	Benutzer wählt einen Spielmodus aus der Liste aus.

Nr.	3
Titel	Öffnen der Einstellungen
Kurzbeschreibung	Der Benutzer hat die Ansicht der Einstellungen geöffnet um die Parameter: <ol style="list-style-type: none">1. Größe des Spielfelds2. Ändern des Aussehens3. Anpassen der Spielregeln
Akteure	Benutzer
Auslösendes Ereignis	Klick auf Einstellungssymbol im Hauptmenü
Vorbedingungen	Kein laufendes Spiel
Nachbedingungen	Keine
Performanceklasse	1
Priorisierung	3
Ablauf	Benutzer ist im Hauptmenü. Benutzer klickt auf Einstellungssymbol. Anzeigen der aktuellen und möglichen Parameter.

Nr.	4
Titel	Speichern der Einstellungen
Kurzbeschreibung	Speichern der gesetzten Einstellungen
Akteure	Benutzer
Auslösendes Ereignis	Der Benutzer klickt auf die Speichern Schaltfläche in der Einstellungsansicht
Vorbedingungen	Einstellungsansicht wurde geöffnet
Nachbedingungen	Keine
Performanceklasse	2
Priorisierung	3
Ablauf	Benutzer drückt auf die Speichern Schaltfläche in der Optionsansicht. Die App speichert die gesetzten Parameter und Ändert die Einstellungen in der Laufzeit. Nach erfolgreichem Speichern wird das Hauptmenü gezeigt.

Nr.	5
Titel	Spiel starten
Kurzbeschreibung	Spiel startet mit den gespeicherten Parametern
Akteure	Benutzer
Auslösendes Ereignis	Klick auf die „Spiel starten“ Schaltfläche
Vorbedingungen	Hauptmenü wird angezeigt
Nachbedingungen	Keine
Performanceklasse	2
Priorisierung	2
Ablauf	Benutzer drückt auf „Spiel starten“ im Hauptmenü. Die App zeigt die Spielansicht mit den eingestellten Parametern.

Nr.	6
Titel	Mastercode eingeben
Kurzbeschreibung	Der zweite Spieler gibt den Mastercode ein
Akteure	Spieler zwei
Auslösendes Ereignis	Spiel gestartet
Vorbedingungen	Der Spielmodus für zwei Spieler wurde gewählt
Nachbedingungen	Keine
Performanceklasse	1
Priorisierung	4
Ablauf	Die Spielansicht wurde geöffnet. Der zweite Spieler gibt den Mastercode ein und bestätigt diesen.

Nr.	7
Titel	Farbvorschlag eingeben
Kurzbeschreibung	Der Benutzer wählt für die einzelnen Spalten die gewünschte Farbe aus
Akteure	Benutzer
Auslösendes Ereignis	Spiel gestartet
Vorbedingungen	Mastercode wurde gesetzt (automatisch oder manuell)
Nachbedingungen	Der Benutzer betätigt die Schaltfläche „Eingabe prüfen“
Performanceklasse	2
Priorisierung	2
Ablauf	Benutzer wählt eine Spalte aus. Benutzer wählt eine der zur Verfügung stehenden Farben aus. Wiederholung für alle verfügbaren Spalten. Betätigen der Schaltfläche „Eingabe prüfen“

Nr.	8
Titel	Farbvorschlag auswerten
Kurzbeschreibung	Die App wertet die Eingabe aus und gibt dem Benutzer die Rückmeldung über seine Eingabe.
Akteure	App
Auslösendes Ereignis	Betätigen der Schaltfläche „Eingabe prüfen“
Vorbedingungen	Die Eingabe wurde entsprechend der festgelegten Regeln und Einstellungen korrekt gemacht
Nachbedingungen	Keine
Performanceklasse	3
Priorisierung	3
Ablauf	App wertet die Ausgabe aus. Benutzer bekommt die Benachrichtigung über sein Ergebnis. Benutzer bestätigt sein Ergebnis.

Nr.	9
Titel	Spiel Pausieren
Kurzbeschreibung	Das Spiel wird pausiert und der Benutzer hat die Möglichkeit im Pausenmenü seinen aktuellen Spielstand zu speichern oder zu verwerfen.
Akteure	Benutzer
Auslösendes Ereignis	Benutzer klickt auf Schaltfläche „Spiel pausieren“
Vorbedingungen	Spiel läuft
Nachbedingungen	Keine
Performanceklasse	2
Priorisierung	3
Ablauf	Benutzer ist im laufenden Spiel. Benutzer klickt auf „Spiel pausieren“ Schaltfläche. Benutzer wählt eine Option aus dem Pausenmenü.

Nr.	10
Titel	Spiel Abbrechen
Kurzbeschreibung	Der Benutzer hat im Pausenmenü die Option „zurück zum Hauptmenü“ betätigt.
Akteure	Benutzer
Auslösendes Ereignis	Betätigung der Schaltfläche „zurück zum Hauptmenü“ im Pausenmenü.
Vorbedingungen	Pausenmenü wurde geöffnet
Nachbedingungen	Benutzer bestätigt das Abbrechen des aktuellen Spiels
Performanceklasse	2
Priorisierung	2
Ablauf	Benutzer betätigt „zurück zum Hauptmenü“. Benutzer bestätigt das Abbrechen des aktuellen Spiels. Spiel wird abgebrochen. Hauptmenü wird angezeigt.

Nr.	11
Titel	Spiel speichern
Kurzbeschreibung	Der aktuelle Spielfortschritt wird mit den aktuellen Spieleinstellungen auf dem Endgerät gespeichert.
Akteure	Benutzer, App
Auslösendes Ereignis	Benutzer wählt die Option „Spiel speichern“ im Pausenmenü
Vorbedingungen	Pausenmenü geöffnet
Nachbedingungen	Benutzer bestätigt das be-/überschreiben des gewählten Speicherplatzes. Benutzer quittiert Erfolg oder Misserfolg der Operation.
Performanceklasse	3
Priorisierung	3
Ablauf	Benutzer wählt „Spiel speichern“ im Pausenmenü. Benutzer wählt einen Speicherplatz. App speichert den aktuellen Zustand des Spiels und dessen Einstellungen an diesem Speicherplatz. App Meldet dem Spieler ob das Speichern erfolgreich oder nicht erfolgreich war.

Nr.	12
Titel	Spiel laden
Kurzbeschreibung	Ein gespeicherter Spielfortschritt wird von dem Endgerät geladen.
Akteure	Benutzer, App
Auslösendes Ereignis	Betätigen der Schaltfläche „Spiel laden“ im Hauptmenü.
Vorbedingungen	App gestartet
Nachbedingungen	Laden des gewählten Spielstandes vom Benutzer bestätigt. Spiel mit gewählten Einstellungen gestartet. App Meldet Erfolg oder Misserfolg der Operation.
Performanceklasse	3
Priorisierung	3
Ablauf	Benutzer betätigt „Spiel laden“ Schaltfläche im Hauptmenü. Benutzer wählt einen Spielstand zum Laden. Benutzer bestätigt das Laden des Spielstandes. App stellt gewählten Spielstand wieder her. App Meldet Erfolg oder Misserfolg der Operation.

Nr.	13
Titel	Aktion rückgängig machen
Kurzbeschreibung	Die letzte vom Benutzer getätigte Spielaktion wird rückgängig gemacht.
Akteure	Benutzer
Auslösendes Ereignis	Benutzer drückt auf die Schaltfläche „Aktion rückgängig machen“. Benutzer bestätigt die Aktion.
Vorbedingungen	Spiel gestartet
Nachbedingungen	Benutzer bestätigt Aktion.
Performanceklasse	2
Priorisierung	4
Ablauf	Benutzer betätigt Schaltfläche „Aktion rückgängig machen“. Benutzer bestätigt diese Aktion. App macht den Zug rückgängig. App sperrt aktuelles Spiel für die Highscore Liste.

Nr.	14
Titel	Spiel gewonnen
Kurzbeschreibung	Der Benutzer hat den Mastercode vor Erreichen der maximalen Versuche erraten und das Spiel gewonnen.
Akteure	Benutzer
Auslösendes Ereignis	Der Benutzer hat den Mastercode vor Erreichen der maximalen Versuche erraten.
Vorbedingungen	Farbvorschlag auswerten
Nachbedingungen	Wenn nicht gesperrt, in Highscoreliste eintragen.
Performanceklasse	2
Priorisierung	2
Ablauf	Benutzer lässt Farbvorschlag auswerten. App vergleicht Farbvorschlag mit Mastercode. Mastercode und Vorschlag stimmen überein. App benachrichtigt Benutzer über Sieg.

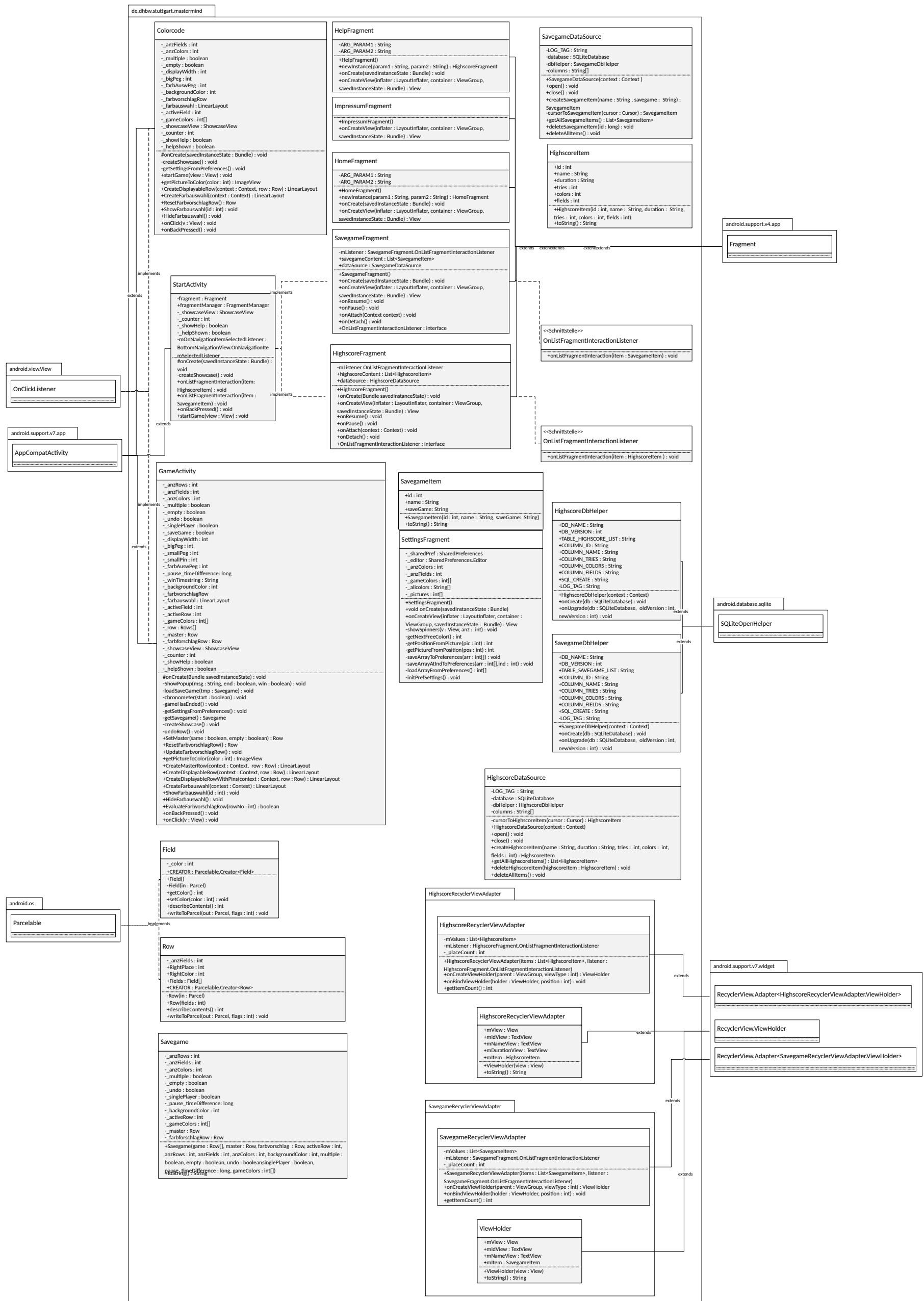
Nr.	15
Titel	Spiel verloren
Kurzbeschreibung	Der Benutzer hat den Mastercode vor Erreichen der maximalen Versuche nicht erraten und das Spiel verloren.
Akteure	Benutzer
Auslösendes Ereignis	Der Benutzer hat die maximale Anzahl an Versuchen überschritten.
Vorbedingungen	Farbvorschlag auswerten
Nachbedingungen	Spiel beenden
Performanceklasse	2
Priorisierung	2
Ablauf	Benutzer lässt Farbvorschlag auswerten. App vergleicht Farbvorschlag mit Mastercode. Mastercode und Vorschlag stimmen nicht überein. Der Benutzer hat die maximale Anzahl an Versuchen überschritten. App benachrichtigt Benutzer über Niederlage.

Nr.	16
Titel	Eintrag in Highscore Liste
Kurzbeschreibung	Benutzer trägt seinen (Spitz-) Namen in die Highscoreliste ein.
Akteure	Benutzer
Auslösendes Ereignis	Spiel erfolgreich abgeschlossen.
Vorbedingungen	Spiel gewonnen. Die Funktion „Aktion rückgängig machen“ nicht benutzt.
Nachbedingungen	Ein gültiger Name wurde eingegeben
Performanceklasse	2
Priorisierung	4
Ablauf	Benutzer gewinnt aktuelles Spiel. Benutzer gibt seinen Namen in der Highscoreliste ein. Spiel ist beendet.

Nr.	17
Titel	Anzeigen der Highscoreliste
Kurzbeschreibung	Anzeigen der aktuellen Highscore Liste
Akteure	Benutzer
Auslösendes Ereignis	Betätigung der Schaltfläche „Highscoreliste anzeigen“ im Hauptmenü
Vorbedingungen	App gestartet
Nachbedingungen	Keine
Performanceklasse	2
Priorisierung	3
Ablauf	Benutzer klickt auf die Schaltfläche „Highscoreliste anzeigen“ im Hauptmenü. App zeigt Highscoreliste an.

Nr.	18
Titel	App beenden
Kurzbeschreibung	Die App beendet sich.
Akteure	Benutzer
Auslösendes Ereignis	Betätigen der „zurück“ Taste des Smartphones im Hauptmenü. Benutzer bestätigt beenden der App.
Vorbedingungen	App gestartet. Hauptmenü angezeigt
Nachbedingungen	App beendet
Performanceklasse	3
Priorisierung	3
Ablauf	Benutzer betätigt die „zurück“ Taste an seinem Smartphone. Die App zeigt die Abfrage ob der Benutzer die App schließen möchte. Bestätigt der Benutzer beendet sich die App.

3. Endgültiges Klassenmodell



4. Priorisierung

Die Priorisierung von Arbeitspaketen geschieht nach folgender Definition:

1. Fertigstellung vor Beginn der App Entwicklung
2. Fertigstellung mit Prototyp 1
3. Fertigstellung mit Prototyp 2
4. Fertigstellung mit finaler Version

5. Testplan

5.1. Testbasis

Die Testbasis enthält die im Pflichtenheft festgelegten Anforderungen des Kunden, die als Basis für die Tests dienen. Die Tabelle TestCase_Übersicht beschreibt diese Punkte und einige weitere für das Testen.

5.2. Teststrategie

Die für das Testen verfügbare Zeit ist durch den Projektplan beschränkt und sollte optimal ausgenutzt werden. Die Kapazität für jeden Testdurchlauf wurde bereits im Projektplan festgelegt. Dies bedeutet, es muss festgelegt werden bei welchem Testdurchlauf welche Priorisierung erfolgen muss. So ist sichergestellt, dass unter optimalem Einsatz der verfügbaren Kapazität und Zeit das beste Testergebnis zu verwirklichen ist.

Die Entwicklung von Mastermind umfasst zwei Prototypen vor der Finalen version. Somit ist vor jedem Prototyp im Projektablauf eine Testphase geplant. Jede Anforderung ist mit Nummern der Testphasen und dessen Priorisierung (1 – 10 wobei 1 die höchste Priorität darstellt) versehen. Aufgrund des kleinen Projektes wurde dieses Konzept gewählt, was an die Produktrisikoprüfung und deren Risikoklassen angelehnt ist.

Der Testplan umfasst folgende Testphasen:

Testart	Ziel
Technischer Integrationstest	Sicherstellen, dass die entwickelte Komponente technisch ablauffähig ist
Anwendungstest	Sicherstellung der Funktionsfähigkeit des Gesamtsystems
Performancetest	Frühzeitiges Feststellen von Performance-Problemen auf Basis von Echtdaten

Während der Entwicklung prüfen die Entwickler intuitiv anhand von White Box Testing neue Funktionen bevor diese in den Test Zyklus weitergegeben werden. Aufgrund von Zeitmangel und Terminbindung ist dies nicht weiter spezifiziert. Bei jedem hochladen neuer Funktionen in die Versionsverwaltung wird ein automatisierter Unittest mit TravisCI durchgeführt.

Generell wird in den Testzyklen nach dem Modell des Black Box Testing vorgegangen. So wird mit der *Funktionsüberdeckung* jede spezifizierte Funktion des definierten Testlauf mindestens einmal in jedem Testzyklus durchlaufen. So wird gewährleistet, dass alle spezifizierten Leistungsanforderungen unter normalen als auch möglichst ungünstigen Bedingungen durchlaufen werden.

Ziel ist es, die gewählten Testziele mit möglichst wenig und möglichst guten Testfällen abzudecken. Testziele werden mit den Techniken der *Äquivalenzklassenbildung*, *Grenzwertanalyse* und *Error Guessing* durchgeführt.

5.2.1. Äquivalenzklassenbildung:

Gleichartige Eingabedaten werden zu Klassen zusammengefasst und aus jeder Klasse wird ein Repräsentant getestet. Die Äquivalenzklassenbildung unterteilt Eingabedaten in Gruppen, so dass man annehmen kann, dass mit jedem Objekt einer Klasse die gleichen Fehler wie mit jedem anderen Objekt dieser Klasse gefunden werden kann.

5.2.2. Grenzwertanalyse:

Die Grenzwertanalyse ist ein Spezialfall der Äquivalenzklassenanalyse. Sie fokussiert sich auf die Grenzen zulässiger Datenbereiche, bei welchen erfahrungsgemäß häufig Fehler auftreten. Es bedarf besondere Aufmerksamkeit um Testfälle für solche Grenzfälle auszuwählen.

5.2.3. Error Guessing:

Basiert auf intuitive Testfallauswahl aufgrund von Erfahrung. Diese Methode ist immer ergänzend zu anderen Teststrategien. Die Qualität hängt immer stark von Erfahrung und Intuition der Tester ab. Darum sollte nach Möglichkeit eine Menge an disjunkten Testern mit unterschiedlichen Herangehensweisen in dieser Teststrategie einbezogen werden.

5.3. Testabschlusskriterien

- Zeit / Kosten

4.1. Die zu investierte Zeit und Kosten pro Testlauf ist in den Arbeitspaketen im SDP festgelegt.

4.2. Wenn alle Teststrategien mit jeweils entsprechend geeigneter Anzahl an Testfällen durchlaufen wurde

Wenn einer dieser Punkte erfüllt ist, ist ein Testabschlusskriterium eingetroffen und der Testzyklus ist beendet.

5.4. Testplan

Generell wird bei jedem neuen Stand der Entwickler automatisiert ein Unittest durchlaufen, welcher das Programm auf Konsistenz und Code Korrektheit in Hinsicht der Syntax prüft. Die drei Testphasen werden anhand der folgenden Tabelle durchlaufen. Immer in der jeweils festgelegten Priorität der entsprechenden Testphase.

Nr.	Qualitätsmerkmal	PERF	L-P	R	Verantw.	Ausführung
1	Appstart	3	2 - 1	A	P. Käufer	
2	Auswahl	1	3 - 3	A	P. Käufer	

Nr.	Qualitätsmerkmal	PERF	L-P	R	Verantw.	Ausführung
	Spielmodus					
3	Öffnen der Einstellungen	1	3 - 3	B	P. Käufer	
4	Speichern der Einstellungen	2	3 - 4	B	P. Käufer	
5	Spiel starten	2	2 - 1	A	P. Käufer	
6	Mastercode eingeben	1	4 - 2	A	P. Käufer	
7	Farbvorschlag eingeben	2	2 - 2	A	P. Käufer	
8	Farbvorschlag auswerten	3	3 - 2	A	P. Käufer	
9	Spiel pausieren	2	3 - 5	C	P. Käufer	
10	Spiel abbrechen	2	2 - 5	C	P. Käufer	
11	Spiel speichern	3	3 - 4	B	P. Käufer	
12	Spiel laden	3	3 - 4	B	P. Käufer	
13	Aktion rückgängig machen	2	4 - 5	C	P. Käufer	
14	Spiel gewonnen	2	2 - 1	A	P. Käufer	
15	Spiel verloren	2	2 - 1	A	P. Käufer	
16	Eintrag in Highscore Liste	2	4 - 5	B	P. Käufer	
17	Anzeigen der Highscore Liste	2	3 - 4	B	P. Käufer	
18	App beenden	3	3 - 3	A	P. Käufer	
19	Ansprechende UI		1 - 3 2 - 5 3 - 8	C	P. Käufer	

Erklärung der vorangegangenen Tabelle:

Kürzel	Beschreibung
L-P	Der Testlauf mit jeweiliger Priorität

Kürzel	Beschreibung
Prüfen	Statische Prüfung / Review der verschiedenen Zwischenprodukte (Anforderungen, funktionales Design, technisches Design)
PERF	Performance Klasse aus dem Pflichtenheft
Ausführ.	Ausführung
Verantw.	Verantwortlich
•	Leichtes dynamisches Testen
••	Durchschnittliches dynamisches Testen
•••	Intensives dynamisches Testen
<leer>	Wenn eine Zelle leer ist, bedeutet das, dass die entsprechende Test- oder Prüfungsstufe das Qualitätsmerkmal ignorieren kann

5.5. Fehlermanagement

Im Fehlerfall wird ein Testprotokoll digital direkt in der Fehlerverwaltung der Versionsverwaltung erstellt. Inhalt ist eine Beschreibung des aufgetretenen Fehlers und dessen Auswirkung. Die Kategorie und Priorität können direkt in der Versionsverwaltung angegeben werden und müssen nicht in dem Protokoll selber angegeben werden.

6. Code Example

6.1. Klassen:

Alle Klassen befinden sich im von uns erstellten Java Package „de.dhbw.stuttgart.mastermind“. Dieses stellt gleichzeitig den Namensraum für die Klassen zur Verfügung.

Das Beispiel besitzt die Eigenschaften „id“, „name“ und „saveGame“, den Konstruktor dem die Werte der Eigenschaften übergeben werden und eine Funktion „toString()“ die den Wert der Eigenschaft „name“ zurück gibt. Diese Funktion überschreibt die Java Funktion „toString()“, weshalb davor das „@Override“ zur Erhöhung der Lesbarkeit notiert ist.

```
package de.dhbw.stuttgart.mastermind;

public class SavegameItem
{
    public final int id;
    public final String name;
    public final String saveGame;

    public SavegameItem(int id, String name, String saveGame) {
        [...]
    }

    @Override
    public String toString() {
        [...]
    }
}
```

Zeichnung 1: Beispiel einer Java Klasse

6.2. Funktionen:

Die hier gezeigte Funktion zeichnet sich dadurch aus, dass sie keinen Wert zurück gibt. Sie ist „void“.

```
public void HideFarbauswahl()
{
    _farbauswahl.setVisibility(Layout.INVISIBLE);
    _activeField = -1;
}
```

Zeichnung 2: Beispiel Funktion ohne Rückgabewert

Wird das „void“ durch den Namen eines beliebigen Datentyp ersetzt so wäre es möglich ein Element diesen Types aus der Funktion zurückzugeben. Im folgenden Beispiel wird eine Instanz der Klasse ViewHolder zurückgegeben.

```
@Override
public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
    View view = LayoutInflater.from(parent.getContext())
        .inflate(R.layout.fragment_savedgames, parent, false);
    return new ViewHolder(view);
}
```

Zeichnung 3: Beispiel Funktion mit Rückgabewert

Diese Funktionen sind „public“ können also immer über die Instanz ihrer übergeordneten Klasse aufgerufen werden. Sie dienen dazu, den Zugriff auf bestimmte Eigenschaften, wie zum Beispiel die Sichtbarkeit im ersten Beispiel, zu steuern.

6.3. Einstellungen:

Zum speichern der Einstellungen erstellt die App eine XML-Datei in ihrem Ordner. Dort werden im XML-Format die gesetzten Einstellungen hinterlegt. Ist eine Einstellung dort nicht hinterlegt, so verwendet die App einen voreingestellten Wert.

```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
  <int name="NumberOfColors" value="5" />
  <boolean name="initialized" value="false" />
  <int name="color_3" value="2130903043" />
  <int name="BackgroundColor" value="2131165280" />
  <int name="color_4" value="2130903046" />
  <boolean name="HelpHome" value="false" />
  <boolean name="Multiple" value="false" />
  <int name="color_0" value="2130903056" />
  <int name="color_5" value="2130903054" />
  <int name="NumberOfHoles" value="5" />
  <boolean name="Empty" value="false" />
  <int name="color_1" value="2130903054" />
  <int name="color_7" value="2130903060" />
  <int name="color_6" value="2130903058" />
  <int name="NumberOfRounds" value="23" />
  <int name="color_2" value="2130903055" />
</map>
```

Zeichnung 4: Beispiel XML-Datei der Einstellungen

7. Deployment Vorgaben

Auf Grund der Umstände des Projekts werden folgende Vorgaben zum Deployment getroffen:

1. Die App wird nur für den privaten Gebrauch und zum Zwecke der Bewertung als Prüfungsleistung weitergegeben.
2. Es erfolgt dementsprechend keine Verteilung über einen App-Markt (z.B. dem Google Playstore / Amazon Underground oder den F-Droid Store)
3. Der Quellcode kann öffentlich auf Github eingesehen werden und erlaubt die Weiterverwendung in eigenen Projekten. Auch das eigenständige bauen der Anwendung ist darüber möglich und von der Lizenz erlaubt.
4. Die App wird zum Zwecke der Abgabe und der Verteilung einmal mit einem

Entwicklerzertifikat und einmal ohne eines erstellt. Die APK der jeweiligen werden als Release ebenfalls auf dem Github Repository veröffentlicht.

5. Die Installation der App geschieht durch das manuelle herunterladen und installieren der APK durch den Benutzer. Eventuell muss der Benutzer dazu das Installieren von Anwendungen unbekannter Herkunft in den Sicherheitseinstellungen seines Android Betriebssystems aktivieren.