



# Hochschule für Angewandte Wissenschaften Hamburg

*Hamburg University of Applied Sciences*

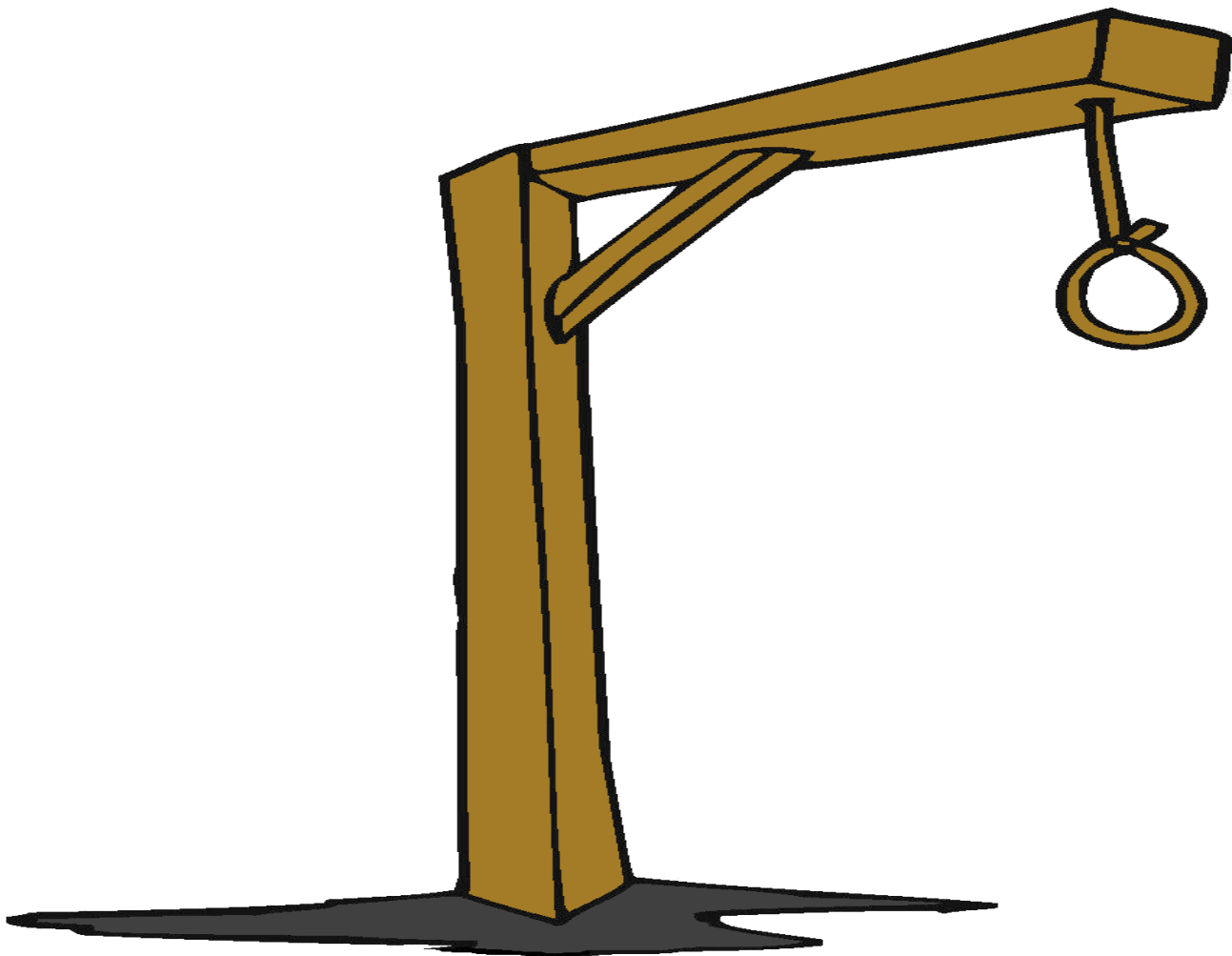
*Fakultät Design, Medien, Information*

*Kunst- und Mediocampus Hamburg*

*Department Medientechnik*

*Studiengang Media Systems*

## ***Galgenmännchen***



**Projektleiter : Herr Prof. Dr. Plaß**

**Mojtaba Asgari Tari (2006879) – Projekt B**

**Shayan Shamseddin (2103763) – Projekt C**

# Inhaltsverzeichnis

1 Projektbeschreibung .....	3
2 Projektplanung .....	3
3 Hardware .....	3
4 Software .....	3
4.1 Android Studio.....	4
4.1.1 Konzept.....	4
4.1.2 Entwurf.....	4
4.2 Phonegap.....	5
4.2.1 Konzept.....	5
4.2.2 Entwurf.....	6
4.3 Web-Version .....	6
4.3 CSS .....	8
5 Umsetzung .....	9
5.1 Analyse und Tests .....	9
5.2 Probleme .....	9
6 Programmierung .....	11
6.1 JAVA .....	11
6.2 JAVA-SCRIPT .....	12
7 Das Logo .....	12
8 So sieht's aus .....	13
9 Weiterentwicklungsmöglichkeiten.....	14
10 Zeitmanagement und Aufgabenverteilung .....	14
11 Fazit .....	15

## 1 Projektbeschreibung

Das Ziel dieses Projekts ist die Entwicklung zwei Galgenmännchen-Spiele, die mit Phonegap(Java-Skript) für Cross-Plattform(Plattformunabhängig), und mit Android(Java) für Android-Geräte geschrieben werden und dementsprechend auf Handys installiert werden können. Das mit Phonegap geschriebenes Spiel wird zu zweit und das mit Android geschriebenes Spiel kann einzeln gespielt werden. Außerdem wollten wir durch dieses Projekt einen Vergleich gewinnen, wie das vorgestellte Spiel-App auf zwei verschiedene Wege realisiert werden kann. Welcher Weg einfacher ist, schneller zum Ziel führt und wann er an seinen Grenzen stößt.

## 2 Projektplanung

Eine möglichst genaue Definition der zu realisierenden Ziele ist wichtig, um das Projekt effektiv umzusetzen. Folgende Meilensteine wurden in der Planung definiert:

**.Entwicklung des Spiels mit Java-Script(Phonegap)**

**.Entwicklung des Spiels mit Java(Android)**

Von Vorteil sind Grundkenntnisse der Java-Script(Mojtaba Asgari Tari), Android-Studio(Shayan Shamseddin), HTML-CSS und objektorientierten Programmierung. Es sollte auch genügend Zeit für Testphasen eingeplant werden, um sich den sicheren Umgang mit den einzelnen Plattformen anzueignen.

## 3 Hardware

Zum Ausprobieren der Apps wurden zwei Android Handys „Samsung-Galaxy-S3 mini“ und „ “ verwendet denn es stand leider kein anderes Gerät zur Verfügung. Deshalb werden die beide Apps in Android-Geräten präsentiert.

## 4 Software

Es werde parallel mit zwei verschiedene Softwares gearbeitet.

-Android Studio für die Android-Version

-Phonegap für Plattformunabhängige Version

Design der Apps werden mit Hilfe von „CSS“ entworfen.

## 4.1 Android Studio

**Android Studio** ist eine freie Integrierte Entwicklungsumgebung von Google und offizielle Entwicklungsumgebung für Android. Android Studio basiert dabei auf IntelliJ IDEA. Es ersetzt unter anderem das Android Developer Tool (ADT) für Eclipse, welches seit Ende 2015 keinen offiziellen Support mehr erhält.

Die App ist mit Android Studio 1.4.1, was am 30.Sep.2015 veröffentlicht wurde und Vector Assets, Theme Editor, Aktualisierte Projekt-Templates, GPU Rendering und Netzwerk Monitor unterstützt für API 17: Android 4.2 (Jelly Bean) geschrieben, was für 73,1% der AndroidUsers umsetzbar ist.

Für die erste Version des Spieles, sind nur drei Java-Klassen mit jeweils passende Layouts programmiert.

Ein Design-Datei(style.xml) ist für alle Klassen geeignet.

### 4.1.1 Konzept

Das Konzept für die erste Version ist im Gegensatz der Cross-Version für Einzelspieler bedacht.

Der Spieler soll so schnell wie möglich mit dem Spiel anfangen können, darum fängt das Spiel nach zwei Klicks an, was bei der Cross-Version auch berücksichtigt wurde.

Werden die Wörtern richtig erraten geht es zur nächsten Runde und Rekord wird bzw. erhöht. Sollte das gesuchte Wort nach fünf falschen Versuche nicht erfasst hat man verloren und wird zur letzten Seite weitergeleitet.

### 4.1.2 Entwurf

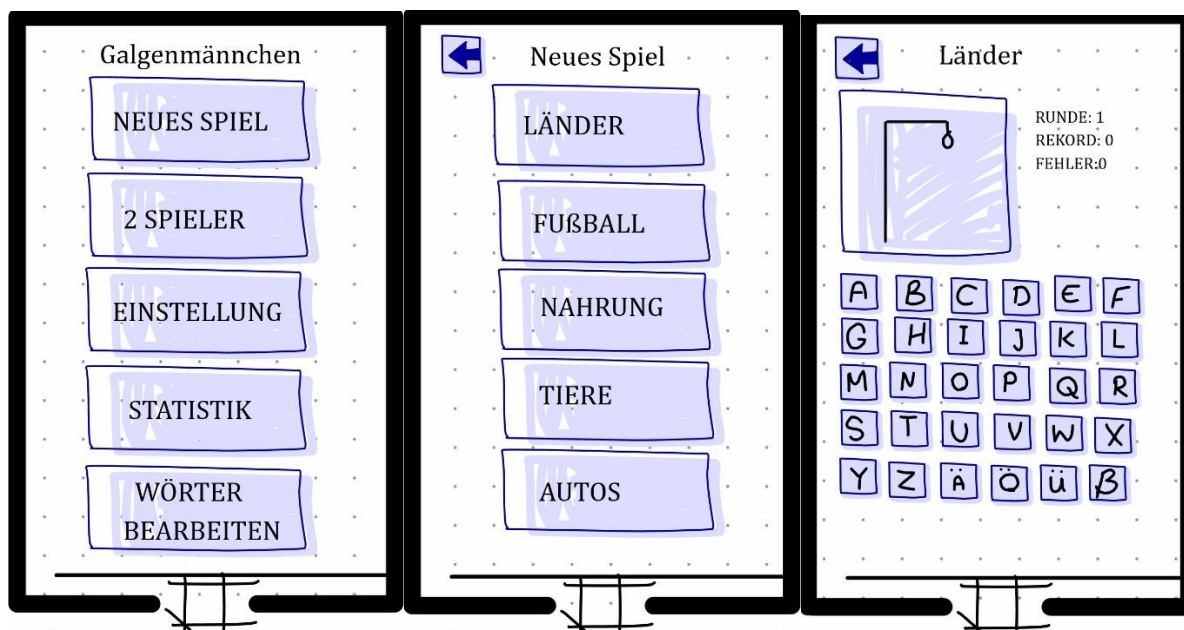


Abbildung 1 (Home)

Abbildung 2 (Thema)

Abbildung 3 (Spielfeld)

Es sollen verschiedenen Kategorien für verschiedene Geschmacks bzw. verschiedene Menschen geben damit die App ernst rüber kommt und nicht direkt langweilig wird.

Für die erste Version wird jedoch von den ersten und zweiten Abbildungen jeweils der erste Knopf programmiert und bei der Präsentation dabei sein.

## 4.2 Phonegap

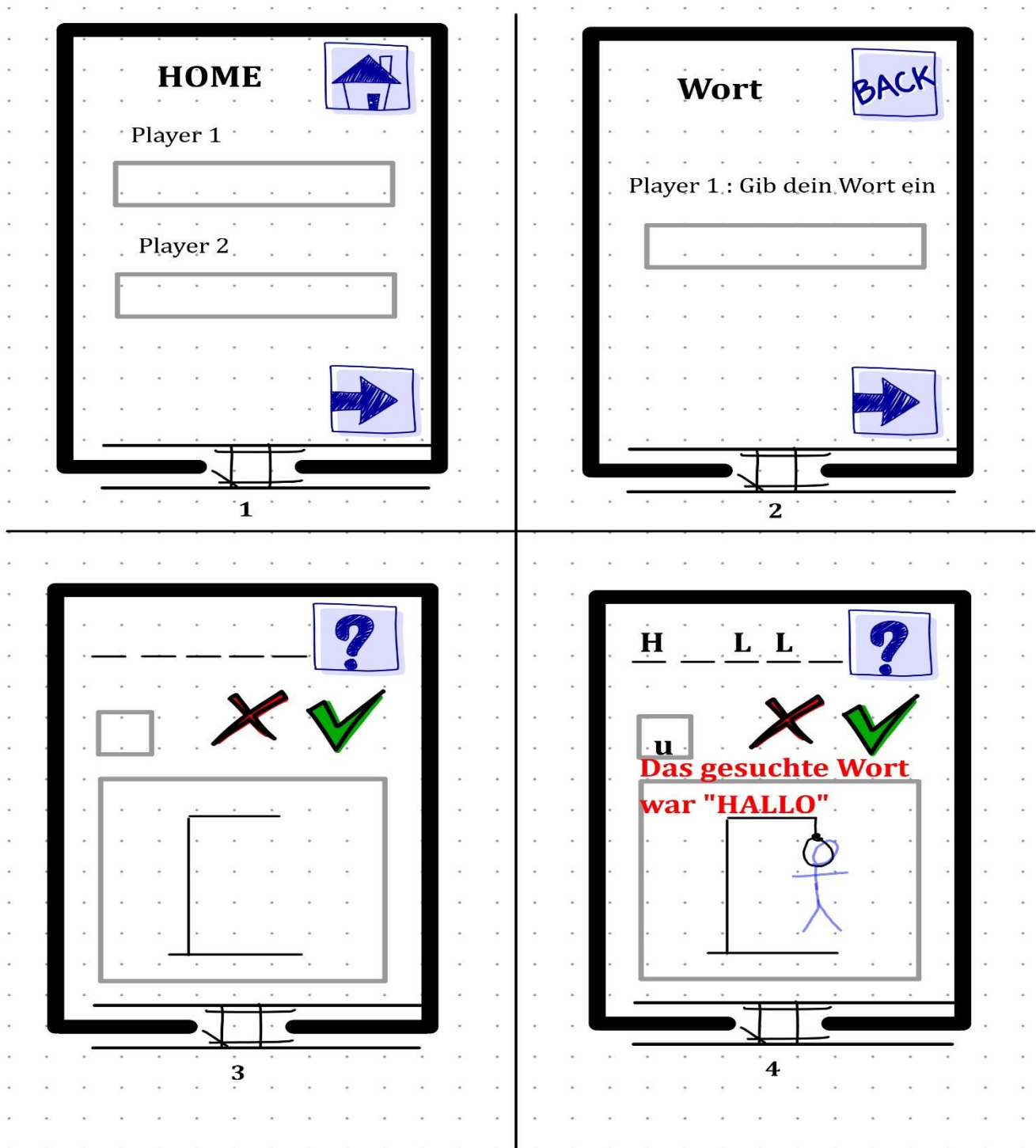
Wer aber eine App-Idee auch mit HTML 5, CSS und JavaScript umsetzen möchte, kann sich die Lernerei der Programmiersprachen der gängigen Mobilgeräte sparen. PhoneGap kann nämlich aus einer Webseite eine native App machen. Für Android, iOS, Windows 8 und noch mehr.

PhoneGap kann eine Web-App in eine native App umwandeln. Diese läuft dann direkt ohne Umweg über den Browser auf dem Smartphone. Zumindest ist für den Anwender kein Browser sichtbar. PhoneGap packt die Web-App also in eine Art Browserhülle. Das Besondere daran: PhoneGap ermöglicht auf diese Art auch Webanwendungen den Zugriff auf die Hardware. So lassen sich via JavaScript zum Beispiel die aktuellen Geokoordinaten auslesen. In unserem Fall werden aber die meisten Sensoren des Handys nicht in Anspruch genommen. Es geht lediglich darum aus einer mobilen Webseite oder einer Web-App eine native App zu erstellen, ohne dafür Java bemühen zu müssen.

### 4.2.1 Konzept

Das Konzept und die Spielweise des Spiels sollte eigentlich jedem bekannt sein und ist mehr oder weniger selbsterklärend. In der Phonegap-Version des Spiels, wird man nach dem Start der App dazu aufgefordert seinen Namen und den von seinem Gegenspieler einzugeben (das Spiel ist für zwei Spieler konzipiert). Danach werden die Spieler abwechselnd dazu aufgefordert ein Wort einzugeben, was der Gegenspieler erraten muss. Wenn das Wort richtig erraten wird, gewinnt der Spieler die Runde und bekommt einen Punkt und wenn nicht, wird die Zeichnung des Galgenmännchens komplett dargestellt und der Spieler verliert die Runde. Das Spiel ist mit Absicht so konzipiert, dass das eingegebene Wort auf jeder Sprache sein kann und außerdem Rechtschreibfehler beinhalten kann, solange die minimale/maximale Länge des Wortes eingehalten wird.

## 4.2.2 Entwurf



Die Phonegap-Version soll genau wie Android-Version mit wenigen Klicks spielbar sein damit der User sich nicht so viel mit Krams beschäftigt und sich dabei ärgert.

## 4.3 Web-Version

Bevor die Phonegap-Version für Handy erstellt wurde, entwarfen wir eine Probe-Web-Version davon.

# Galgenmännchen

Teilnehmer : Mojtaba Asgari-Tari (2006879) & Shayan Shamseddin (2103763)

Mojtaba

Shayan

START!!!

Copyright © 2015

1

# Galgenmännchen

Teilnehmer : Mojtaba Asgari-Tari (2006879) & Shayan Shamseddin (2103763)

Universität

LET'S BEGIN!!!

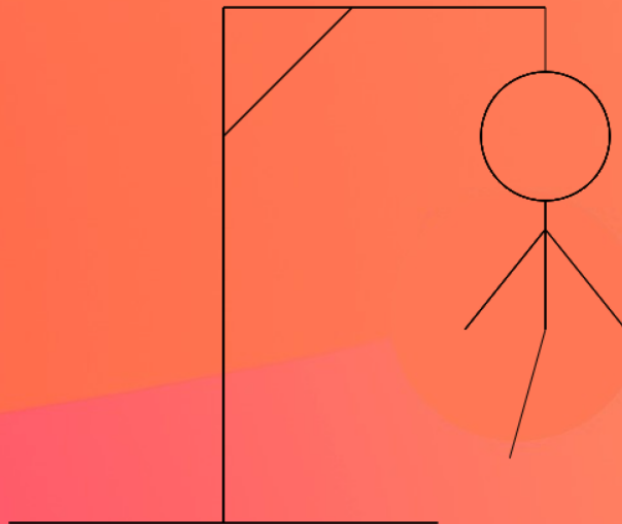
Copyright © 2015

2

Teilnehmer : Mojtaba Asgari-Tari (2006879) & Shayan Shamseddin (2103763)

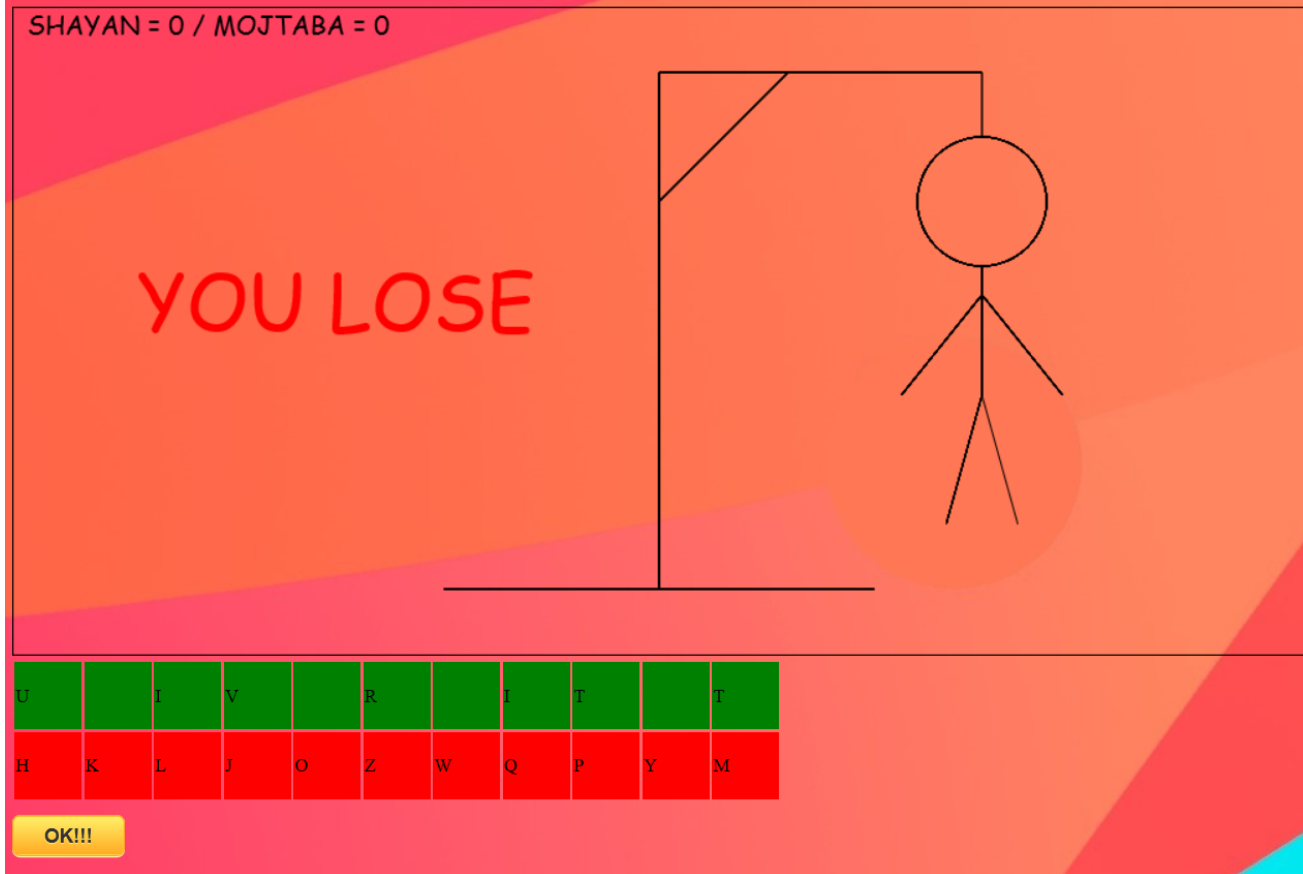
ENTER

SHAYAN = 0 / MOJTABA = 0



U		I	V		R		I	T		T
H	K	L	J	O	Z	W	Q	P	Y	

3



## 4.4 CSS

### Phoneygap (Android) :

Bei der Phoneygap version dient CSS nicht nur das Aufhübschen der Elemente sondern auch die Anpassung und richtige Darstellung des Apps für verschiedene Bildschirme. Nach unsern Tests auf zwei Android Gräte und auch auf dem Laptop Bildschirm ist dies uns auch gelungen.

### Android Studio :

Bei der Android-Version wurde ein „xml“ Datei im Values namens „styles.xml“ gestellt in dem die Styles von den Objekten geschrieben sind.Z.B. : `<style name="ButtonStyle">`

### Java-Script:

Bei der Web-Version wurde das erste CSS-Datei erstellt, das wir bei den anderen Versionen auch verwenden wollten.

Es hat jedoch nicht mit der Implementierung geklappt.Ein Stück von dem Button-Style sieht bei der Web-V. so aus :

```
.myButton {
    -moz-box-shadow: 0px 1px 0px 0px #fff6af;
    -webkit-box-shadow: 0px 1px 0px 0px #fff6af;
    box-shadow: 0px 1px 0px 0px #fff6af;
```



```

background:-webkit-gradient(linear, left top, left bottom, color-stop(0.05, #ffec64), color-
stop(1, #ffab23));
background:-moz-linear-gradient(top, #ffec64 5%, #ffab23 100%);
background:-webkit-linear-gradient(top, #ffec64 5%, #ffab23 100%);
background:-o-linear-gradient(top, #ffec64 5%, #ffab23 100%);
background:-ms-linear-gradient(top, #ffec64 5%, #ffab23 100%);
background:linear-gradient(to bottom, #ffec64 5%, #ffab23 100%);
filter:progid:DXImageTransform.Microsoft.gradient(startColorstr='#ffec64',
endColorstr='#ffab23',GradientType=0);
background-color:#ffec64;
-moz-border-radius:6px;
-webkit-border-radius:6px;
border-radius:6px;
border:1px solid #ffaa22;
display:inline-block;
cursor:pointer;
color:#333333;
margin-top:1%;
font-family:Arial;
font-size:15px;
font-weight:bold;
padding:6px 24px;
text-decoration:none;
text-shadow:0px 1px 0px #ffee66;
}

```

Es wurde ganz viel an dieser CSS-Datei gearbeitet was am Ende für andere Versionen leider nicht zum Einsatz kam

## 5 Umsetzung

Die Realisierung beider Apps sind mit Android-Geräten ermöglicht worden denn es war kein iOS oder WindowsPhone Handy vorhanden. Die Cross-Version ist jedoch für iOS und Windows-Phone mit nur paar Klicks umsetzbar.

### 5.1 Analyse und Tests

Die beide Apps sind auf Samsung Galaxy S3 mini, Galaxy S4, S5 und Sony-xperia-z3 analysiert und getestet worden und bei keinen der genannten Handys trat ein Fehler auf.

### 5.2 Probleme

Bei der Android-App traten immer wieder Fehler wegen der Nutzung der Java-Codes, die beim Recherchieren entweder ganz weggelassen oder dem Android angepasst wurden. Zum Beispiel die Umwandlung von Text zu Char oder String usw.

Bei der Phonegap-Version ...

Die Haupt-Probleme bei der Entwicklung des Apps die leider auch viel Zeit in Anspruch genommen haben, sind bei der Installation des Phonegaps und später bei der Ausführung des Programms aufgetreten. Die Installation von Phonegap ist ziemlich kompliziert und etwas zeitaufwändig. Das liegt hauptsächlich daran dass Phonegap-Installation allein nicht reicht um das Programm zu nutzen. Bevor PhoneGap also installiert werden kann, muss zunächst Node.js heruntergeladen und installiert werden. Da native Android Apps mit der Sprache Java programmiert werden, ist die Installation des Java-SDK notwendig. Phonegap benötigt das SDK zum Kompilieren der Web-Apps in native Anwendungen. Das Android Studio ist die offizielle Entwicklungsumgebung für Android Applikationen von Google. Auch diese Software muss heruntergeladen und installiert werden. Auch Apache Ant muss heruntergeladen und installiert werden. Nun müssen die Umgebungsvariablen hinzugefügt und angepasst werden. Erst jetzt kann Phonegap installiert und ausgeführt werden.

Die Bedienung des Phonegaps erfolgt über die Eingabeaufforderung (Kommandozeile) des Node.js oder die von Windows. Hier sind in unserem Fall bei der Erstellung des Apps Fehlern aufgetreten die wir nicht nachvollziehen und beheben konnten.

Zum Glück gibt es auch einen Webdienst (build.phonegap.com), der aus den hochgeladenen Dateien eine App für Android, IOS und Windows erstellt, die dann auf dem Rechner heruntergeladen werden kann oder mit einem QR-Reader direkt aufs Handy heruntergeladen und danach installiert werden kann.

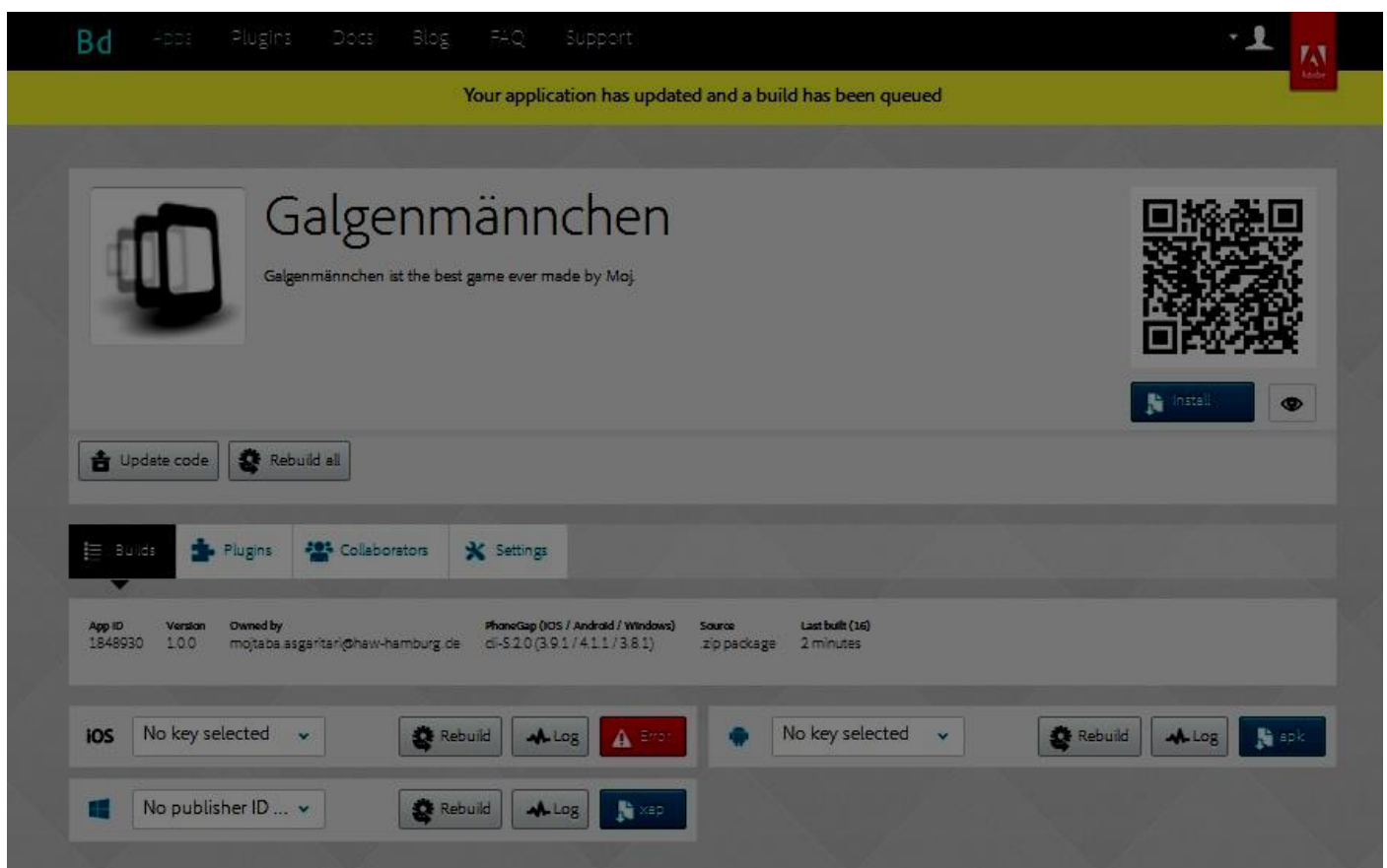


Abbildung: Screenshot von build.phonegap.com

## 6 Programmierung

### 6.1 JAVA

Von dem Home-Screen landet man mit dem onClick-Funktion und durch „intent“ zur zweiten Seite und das Gleiche passiert auf der zweiten Seite mit der Unterschied das diesmal intent-Befehl ein Extra-Befehl mit sich schleppt und zu dem Spiel-Klasse weitergibt. Der Extra-Befehl beinhaltet das Thema, das User auswählt damit das Wort mit Random-Funktion gewählt wird. Nach dem das Wort gewählt wurde wird ein TextView-Feld für die Anzahl der Buchstaben des Wortes die Fragezeichen umsetzen. Damit wird dem User klar wie viele Buchstaben das gesuchte Wort beinhaltet und kann anfangen das Wort durch Drücken auf der eingebauten Tastatur einzugeben.

Durch Drücken der Buchstaben, die mit onClick-Funktion sich mit der Main kommunizieren wird es getestet ob die Char mit einem Char von dem Wort übereinstimmt. Sollte es der Fall sein wird der Char gesetzt und wenn nicht zählt der Fehler-Zähler hoch und lädt das nächste Bild hoch.

Durch Intent-Funktion wird man zu der anderen Klasse weitergeleitet und mit „putExtra“ werden Informationen mitgeliefert.

```
public void firstClick(View view){
    Intent in = new Intent(this, Spiel.class);
    auswahl = "Länder";
    in.putExtra("auswahl", auswahl);
    startActivity(in);
}
```

Damit der User für die Eingabe der Buchstaben nicht jedes Mal die Handy-Tastatur aufmachen muss wurden die Buchstaben als Buttons in der App hinzugefügt die mit setClick-Methoden gerufen werden und durch andere Methoden die Übereinstimmung der einzelnen Buchstaben mit den Buchstaben des gesuchten Wortes testen.

Z.B. die A-Clicked sieht so aus :

```
public void aC(View view){
    if(testClass('A')){
        testTrue();
    }else{
        falscheEingabe();
    }
}
```

In der testClass(char charTest) wird die Übereinstimmung getestet und wenn das stimmt führt testTrue() dazu, dass die eingegebene Buchstabe für „?“ eingesetzt werden und dann wird die winTest(String[] arrayWort) testen ob die Buchstaben komplett eingesetzt sind und wenn ja hat man das Wort richtig erraten und wird zur nächsten Runde geführt.

Sollte es nicht der Fall sein und der User falsche Buchstabe eingegeben hatte, führt die falscheEingabe() dazu, dass der Fehler um ein erhöht und das nächste Bild geladen wird.

Sollte es der letzte Versuch sein wird das gesuchte Wort nach 5 Sekunden Pause (freeze(5000)) angezeigt und der User wird an der ersten Seite (Home) weitergeleitet.

## 6.2 JAVA-SCRIPT

Nach dem das App gestartet ist wird den Nutzer dazu aufgefordert die Namen der Spielern einzugeben. Die Eingaben erfolgen in zwei INPUT-Tags und nach der Aktivierung des START!!! buttons, wird die funktion start() ausgelöst. Die Eingaben werden in zwei Variablen gespeichert. Falls keine Eingabe erfolgen soll, werden die Default Werte: PLAYER1 und PLAYER2 in den Variablen gespeichert. Eine Confirm Nachricht erklärt wie es nun weiter geht.

Die Eingabe des Wortes erfolgt auch über ein INPUT und BUTTON Kombination. Das LETS BEGIN!!! Button löst die Funktion firstClick() aus die, die Funktion validate() aufruft und sorgt dadurch dafür dass die Eingabe keine Zahlen Sonderzeichen und Spaces beinhaltet und mindestens drei Characters lang ist. Außerdem wird das eingegebene Wort innerhalb diese Funktion analysiert, gespeichert und abhängig von der Länge des Wortes passende Anzahl der Zellen für die Tabelle erstellt.

Im nächsten Schritt, wird nach der Eingabe von jedem Buchstabe und Betätigung des Buttons die Funktion nextLevel() aufgerufen. Diese Funktion sorgt für das Vergleich des Richtigkeit der Eingabe. Wenn die Eingabe Richtig ist, wird die in den entsprechende Zellen der Tabelle ausgegeben. Falls die Eingabe kein Buchstabe aus dem Wort ist, wird ein Zeichen-Funktion für das Canvas-Element aufgerufen. Dieser Schritt wiederholt sich solange bis das Wort komplett ist oder die rote Zellen voll und entsprechend die Zeichnung auf der Canvas Vollständig ist.

Gewinnt oder verliert der Spieler erscheint ein OK!!! Button. Wird OK!!! benötigt, wird die Funktion nextRound() aufgerufen die für die Punkte-Vergabe sorgt und den Inhalt des Canvas löscht und das Variable für das Wort zurücksetzt. Danach erscheint die Nachricht dass der andere Spieler dran ist und so kann der nächste Spieler sein Wort eingeben.

Dank die Manipulation des DOM-Tree durch JavaScript und CSS läuft das ganze Spiel auf nur eine Seite. Die Elemente werden abhängig von der ausgelöste Funktion Sichtbar oder verdeckt.

## 7 Das Logo

PhoneGap versieht die App standardmäßig mit dem eigenen Symbol. Klar, hier muss noch ein eigenes Symbol eingefügt werden. Das Symbol für unser App haben wir mit Microsoft Paint zusammengestellt.

Durch die vielen möglich Auflösungen von Android Smartphones, werden vom Icon verschiedene Versionen benötigt. Hierfür gibt es ein kleines Tool namens makeappicon.com. Das erstellte png oder jpg, muss nur noch in die Ablegefläche bei **makeappicon.com** gezogen werden. Der Dienst erstellt dann aus der Datei die passenden Icons für jede Auflösung.

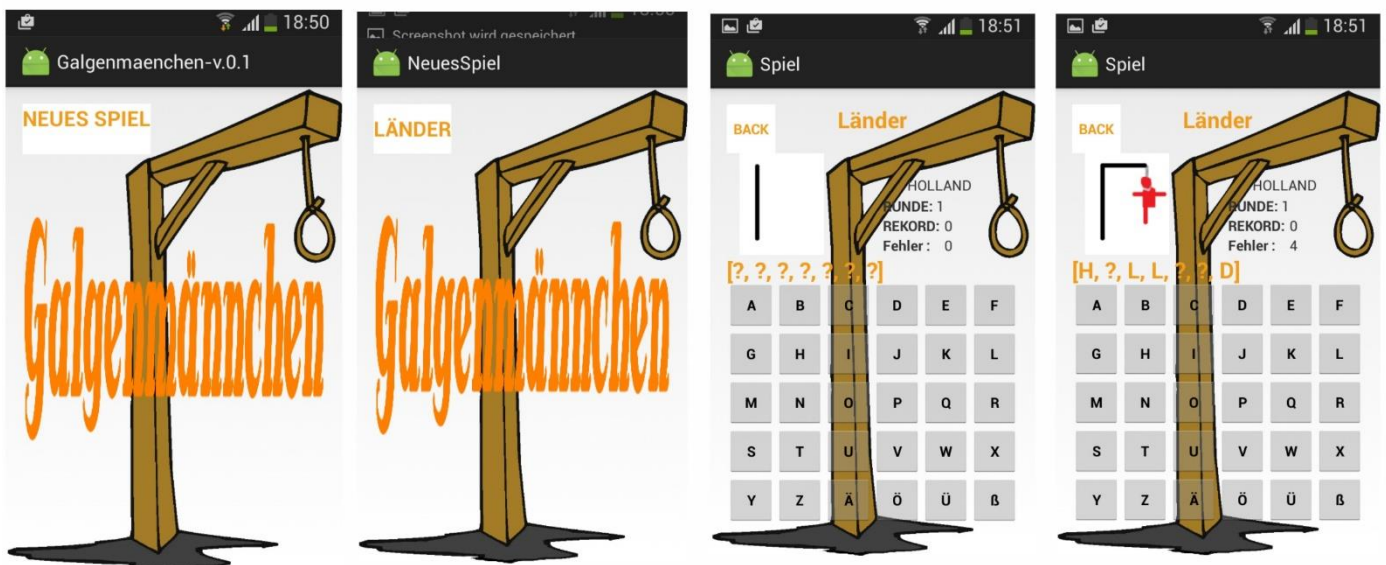
Anhand der Benennung ist auch ersichtlich, welches Icon in welchen Ordner kopiert werden muss.

## 8 So sieht's aus

### Galgenmännchen



### Galgenmännchen V.1



## 9 Weiterentwicklungsmöglichkeiten

Es besteht bei den beiden Versionen große Futures so wie Online-Spielen oder Zeit-Spiel etc.

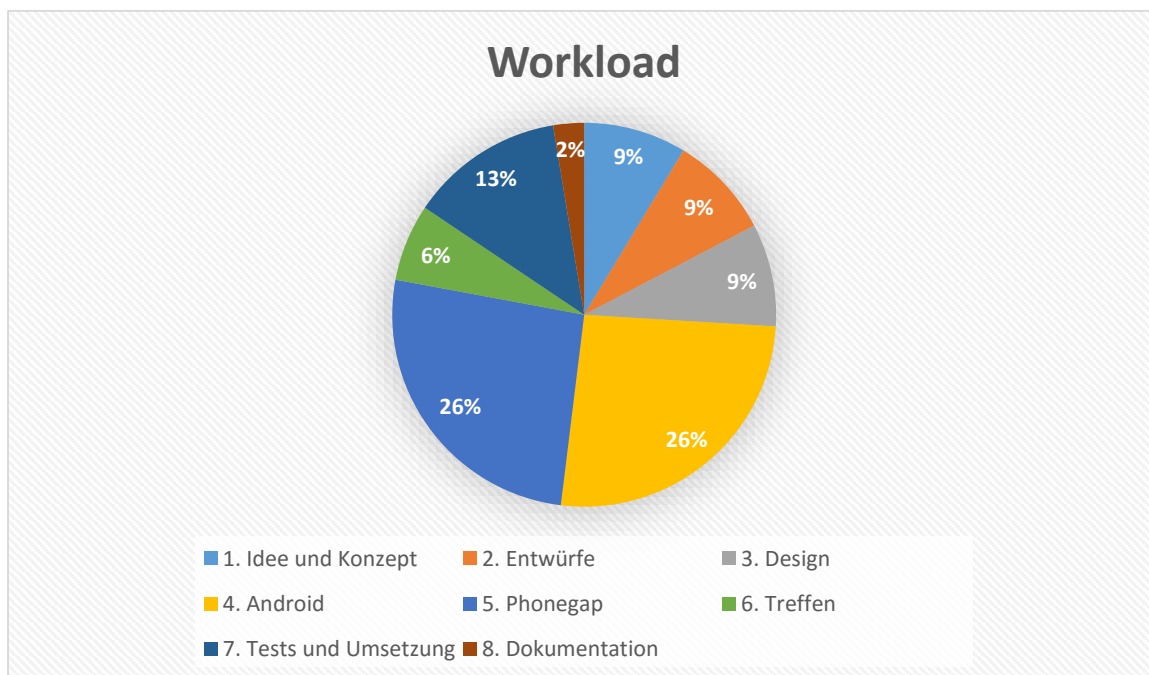
Die Cross-Plattform-Version wird für andere Betriebssysteme vorbereitet und im Store hochgestellt.

Bei der Android-Version wird die Möglichkeit zu zweit zu Spielen programmiert und es werden noch mehrere Themen hinzugefügt. Für die Präsentation sind nun zwei Themen nämlich : Länder und Tiere vorbereitet aber wie es aus dem Entwurf zu sehen ist sollten noch mehrere Themen dazu kommen.

Es soll auch ein freies Thema für den User geben, wo er sich den Name des Themas auswählen darf und dazu die passenden Worten hinzufügen kann. So könnte das Spiel noch mehr Spaß bringen.

Die Web-Version sollte auch noch als ein Online-Spiel weiterentwickelt werden wobei die Usern sich von zu Hause einloggen und gegen einander spielen können. Die weitere Funktionen so wie Themen oder mit der Zeit spielen könnte auch als alternativ hinzugefügt werden.

## 10 Zeitmanagement und Aufgabenverteilung



**Workload : (Die Zahlen sind abgerundet)**

1. Idee und Konzept : 20 Stunden
2. Entwurf : 20 Stunden
3. Design : 20 Stunden
4. Android : 60 Stunden
5. Phonegap : 60 Stunden
6. Treffen : 15 Stunden
7. Tests und Umsetzung : 30 Stunden
8. Dokumentation : ca. 10 Stunden

**Workload insgesamt in Stunden : ca. 235 Stunden**

**Teilnahme :**

Mojtaba Asgari Tari : 1, 2, 3, 5, 6, 7, 8

Shayan Shamseddin : 1, 2, 3, 4, 6, 7, 8

## 11 Fazit

Durch die gründliche Planung des Projektes verlief die Durchführung ohne große Probleme, sodass nur geringe Zeitabweichungen auftraten, wogendessen wir die Abgabe verschieben mussten. Es wurden erst bei der Umsetzung uns klar dass wir auf vieles nicht geachtet haben und um die zu ändern mussten wir vieles am Code ändern denn es gab immer ein Zusammenhang zwischen den Elementen.

Durch das Projekt könnten wir viel über JavaScript dazu lernen. Da wir bis jetzt JavaScript nur über Libraries benutzt haben und für dieses Projekt haben wir uns dafür entschieden mit reine JavaScript zu arbeiten. Außerdem haben wir uns zum ersten mal mit dem Canvas-Element auseinander gesetzt und und damit beschäftigt wie wir durch CSS unser App für verschiedene Bildschirmen passend darstellen.

Wenn man eine Idee für ein App hat und die Meinung ist dass das App mit HTML, CSS und JavaScript realisierbar ist sollte man auf jedenfall Phonegapp in Betracht ziehen. Denn dadurch muss man sich nur mit einer Sprache auskennen um den App für verschiedene Plattformen erstellen zu können und letztendlich gleichzeitig in den verschiedenen Shops anzubieten. So kann man sich jede Menge Zeit und Bemühungen ersparen.