

# **Systopia Companion App - Dokumentation**

Von: Walid Abdul, Bager Bingöl, Alexander Fuhrmann, Sina Sheykholeslami,  
Christoph Thiel, Christopher von Bargaen

Betreuer: Prof. Dr. Andreas Plaß

16. September 21

Projekt B

Studiengang Media Systems (B.Sc.)

---

Hochschule für angewandte Wissenschaften Hamburg /  
Hamburg University of Applied Sciences

Department Medientechnik  
Fakultät Design, Medien und Information

<b>1. Idee .....</b>	<b>3</b>
1.1 Entstehung	3
1.2 Planung	4
<b>2. Aufbau der Software.....</b>	<b>5</b>
2.1 Die Oberfläche	5
2.1.1 Einführungsseiten	5
2.1.2 Hauptseiten	6
2.2 Die Softwarearchitektur	9
<b>3. Projektauswertung .....</b>	<b>13</b>
3.1 Projektablauf	13
3.2 Tatsächlicher Aufwand (Beispiel Bager Bingöl)	15
<b>4. Quellenverzeichnis .....</b>	<b>16</b>
<b>5. Abbildungsverzeichnis .....</b>	<b>17</b>

# 1. Idee

## 1.1 Entstehung

Während des zweiten und dritten Semesters im Fach „Künstlerische Gestaltung 2“ hatten wir uns als gesamte Gruppe die sehr allgemeine Aufgabe gestellt, ein „multimediales Adventure“ auf die Beine zu stellen. Als Studenten einer medientechnischen Fachrichtung sind wir, noch bevor wir uns auf ein Thema oder die Story geeinigt hatten, alle auf die Idee gekommen, dass eine „Companion App“ auf jeden Fall Teil des Projekts sein sollte. Wir waren uns sicher, dass die „Companion App“ nicht nur ein „cooles Gimmick“ wie in vielen anderen Fällen werden sollte, sondern ein zentraler, praktischer Bestandteil des Projekts, welcher viele unserer Pläne und Ideen für das Projekt erst praktikabel machen sollte.

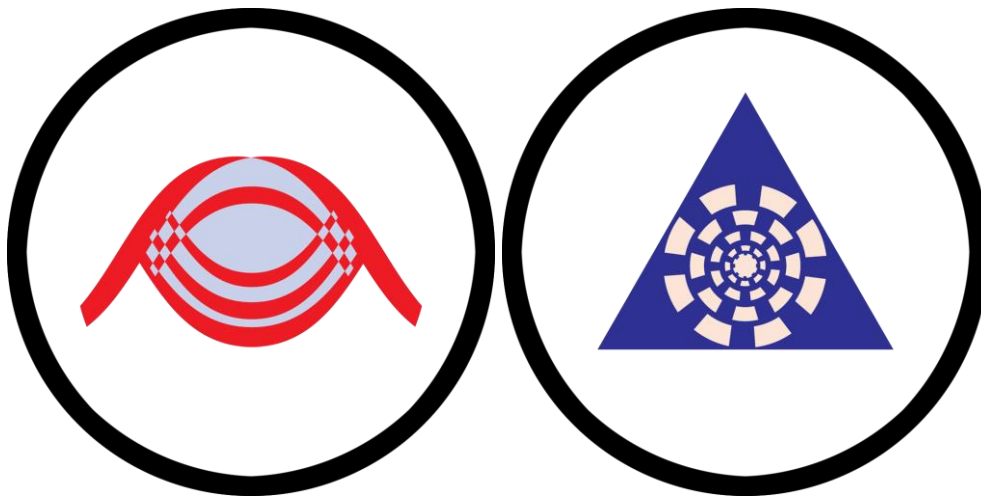


Abbildung 1: Avantgardisten- (links) und Technokratenlogo (rechts)

## 1.2 Planung

Als generelle Inspiration für unsere „Companion App“ schauten wir uns den „Pip-Boy 3000“ aus der Videospielserie „Fallout“ an. Im Spieluniversum ist dies ein am Unterarm angebrachter Computer. Zu seinen Funktionen gehören unter anderem: *Status des Trägers, Karten, Objekteigenschaften, Notizen, ein Inventar...* (Pip-Boy 3000, 2016)



Abbildung 2: Pip-Boy 3000

Als wir mit der Entwicklung der „Companion App“ beginnen wollten, war das Projekt mit dem Arbeitstitel „Taverne“ noch in seiner Anfangsphase, was den Entwicklungsbeginn erschwerte. Um nicht zu viel Zeit an Funktionen zu verschwenden, die möglicherweise nicht in der Endversion auftauchen, hielten wir erstmal einige Besprechungen in denen wir festlegten, was in jedem Fall gebraucht werden würde. Details und spezielle Anforderungen von den einzelnen Gruppen wurden erst eingebracht, als das Projekt etwas reifer war.

## 2. Aufbau der Software

### 2.1 Die Oberfläche

#### 2.1.1 Einführungsseiten

Da „Systopia“ ein „multimediales Rollenspiel“ mit vielen Variablen und optionalem Inhalt ist, brauchten wir natürlich als erstes ein Account-System, um jeden Teilnehmer identifizieren zu können und damit die persönlichen Geschehnisse jedes Spielers auch mit den Informationen in ihrer „Companion App“ korrespondieren. Daher ist die erste Seite auf der man landet ein ganz übliches Registrierungs-/Loginformular. Wird auf „Registrieren“ gedrückt, erscheint ein zweites, selbsterklärendes Passwortfeld. Direkt darauf folgt eine simple „Willkommens-Seite“, auf der es Informationen zur „Companion App“ gibt und die nächsten Schritte erklärt werden.

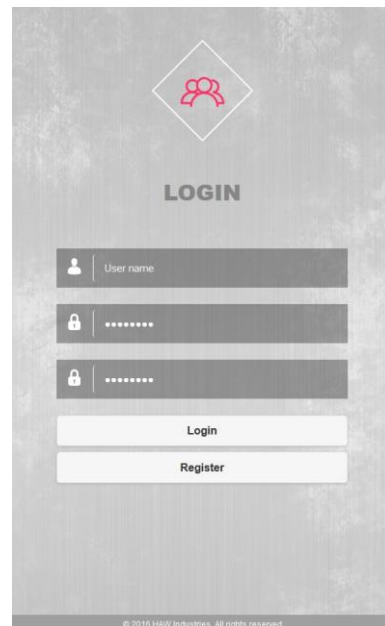


Abbildung 3: Login-Seite



Abbildung 4: Klassenauswahl

Und dieser nächste Schritt ist die Klassenauswahl. Auf diesem neuen Bildschirm kann der Spieler eine der wichtigen Entscheidungen in „Systopia“ treffen, die Einwirkung auf das gesamte restliche Spiel hat. Dem Spieler wird eine Auswahl von 5 Klassen mit jeweils 2 Namen präsentiert. Der endgültige Klassenname wird im späteren Verlauf dadurch entschieden, welcher Fraktion man sich anschließt. Es wird nach Anklicken einer Klasse ein Pop-Up mit Informationen bezüglich dieser angezeigt und nach einer Bestätigung der Auswahl gebeten.

### 2.1.2 Hauptseiten

Nach der Klassenauswahl hat man Zugang zu den drei Hauptseiten unserer „Companion App“. Hierzu gehören die „Inventar“-Seite, die „Quests“-Seite und die „Minimap“-Seite. Am oberen Bildschirmrand ist zu allen Zeiten die Menüleiste sichtbar, welche selbstverständlich zur Navigation dient und darüber hinaus Statusinformationen über den Charakter des Spielers enthält. Da wir die Statusinformationen sehr platzsparend darstellen konnten und Sie im Gegensatz zum Inventar oder der Questliste statisch sind, dachten wir uns, dass Sie keine extra Seite brauchen, sondern in der Menüleiste besser aufgehoben sind. Außerdem gibt es einen Logout-Button, der bei Betätigung selbstverständlich wieder zum Login-/Registrierungsbildschirm führt.

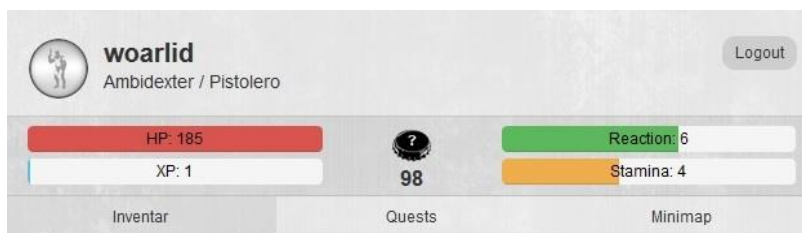


Abbildung 5: Menüleiste

Der **Inventarbildschirm** ist eine denkbar einfache Seite, auf der, wie in jedem normalen Rollenspiel alle gesammelten Waffen, für Quests relevante Gegenstände und auch sonstige Gegenstände aufgelistet werden. Wird auf eines der Gegenstände im Inventar geklickt, erhält man ein Pop-Up mit detaillierteren Informationen zum jeweiligen Item.

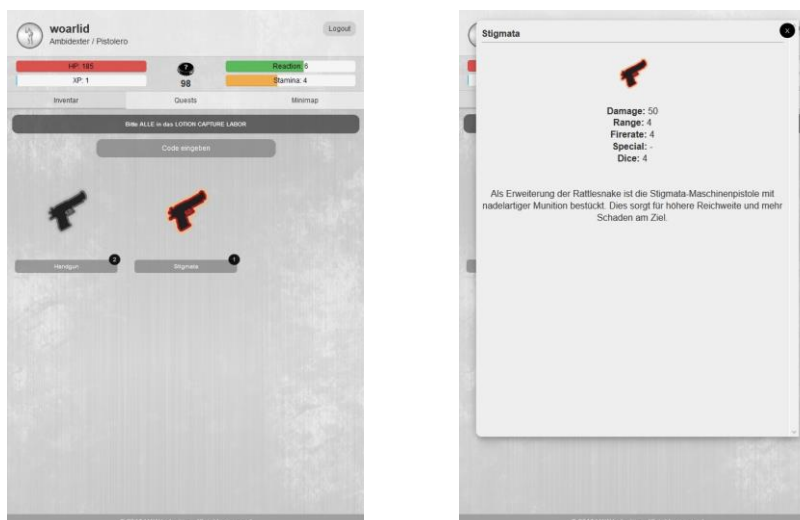


Abbildung 6: Inventarbildschirm (rechts mit Item-Beschreibung)

Auf dem **Questbildschirm** findet man, ähnlich wie beim Inventarbildschirm, eine Sammlung aller aktiven und abgeschlossenen Quests mit Questbeschreibung und kurzen Tipps zur Erledigung dieser. Fertige Quests werden abgehakt und nach unten sortiert.

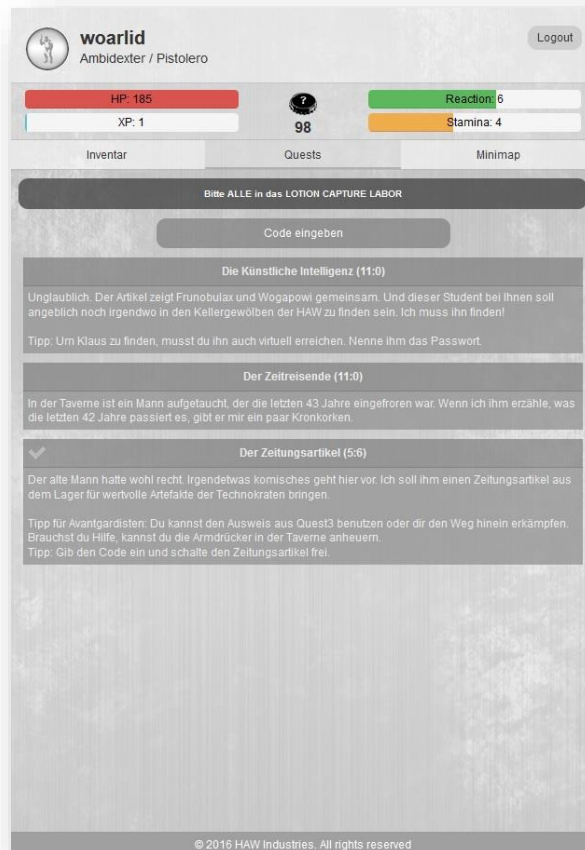


Abbildung 7: Questbildschirm

Als letztes haben wir den „**Minimap**“-Bildschirm. Wie der Name verrät, findet man hier eine Karte, die einem bei der Orientierung in der Spielwelt helfen soll. Den verschiedenen Stationen wurden in der Legende zu den Fraktionsfarben passende, farbige Zeichen zugeordnet. Je nachdem, ob und welcher Fraktion man angehört, kriegt man eine andere Minimap angezeigt. Die Minimap ist eine überarbeitete Version des Raumplans aus dem Studienführer für Media Systems.

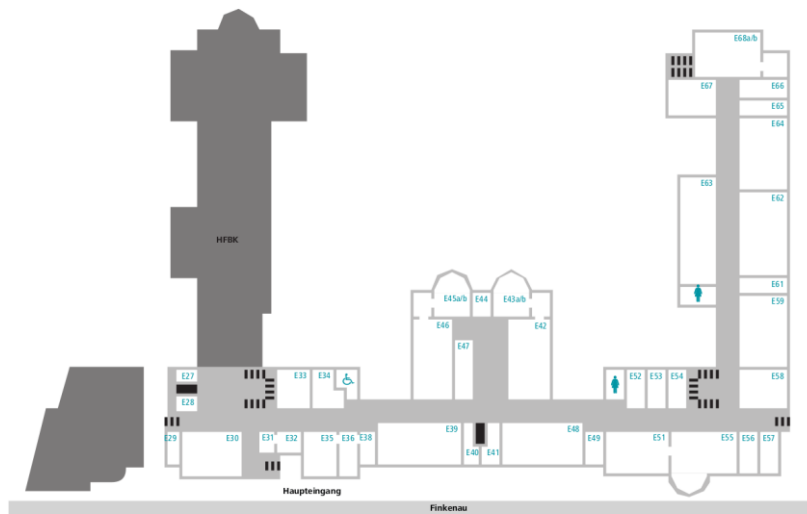


Abbildung 8: Raumplan Studienführer Media Systems (Studienführer, 2017, S. 9)

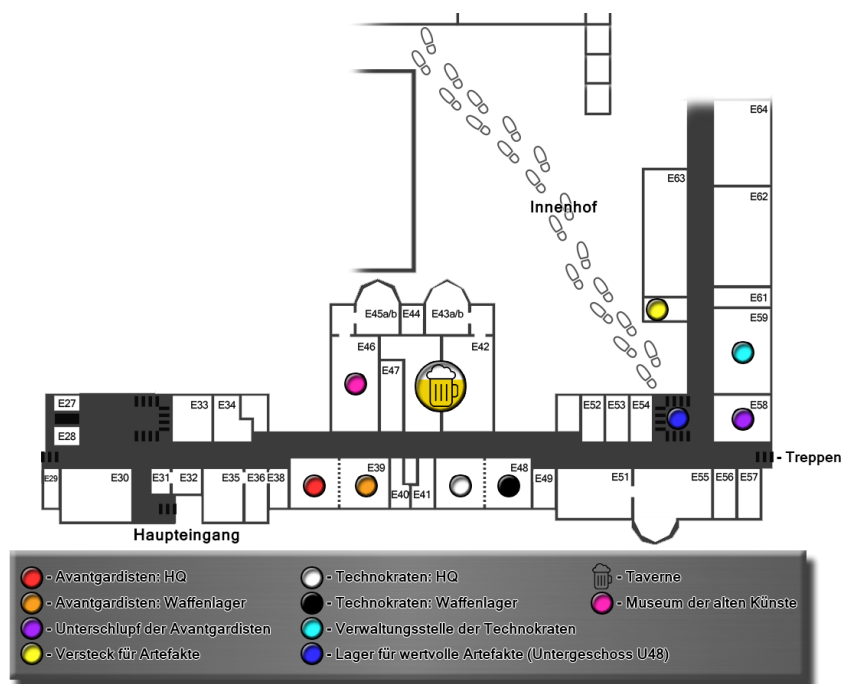
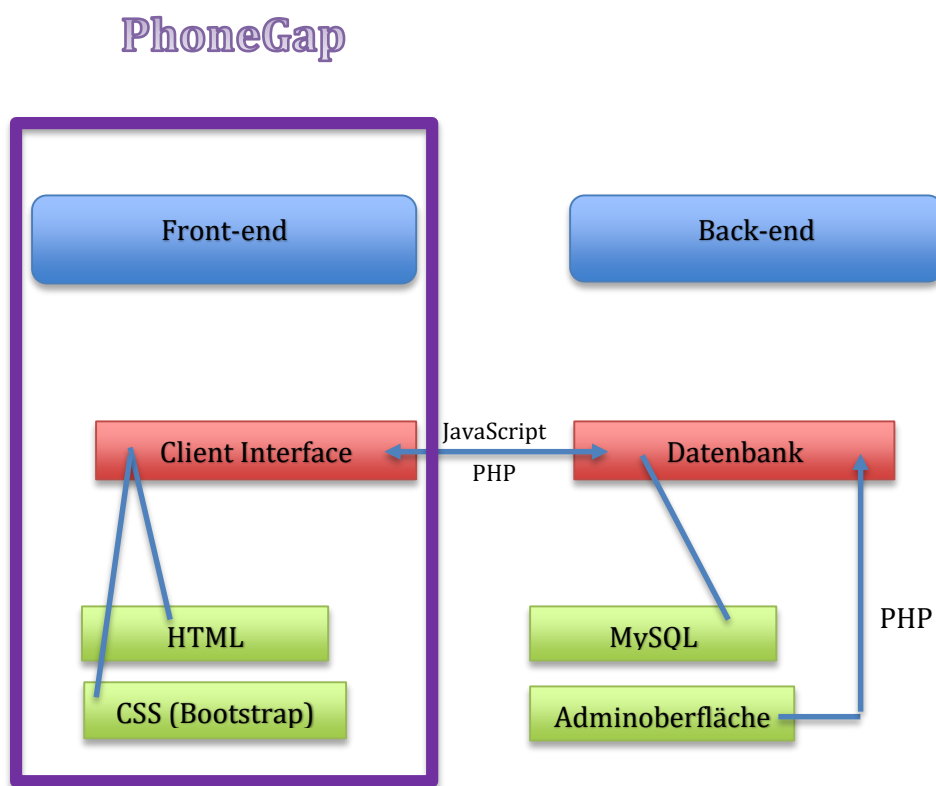


Abbildung 9: Minimap (Gesamtmap)



## 2.2 Die Softwarearchitektur

Im folgenden Teil wird beschrieben aus welchen Komponenten unsere „Companion App“ technisch betrachtet besteht und wie diese zusammen funktionieren. Die Mitglieder, die hauptsächlich für die Softwareprogrammierung zuständig waren, kommen hier zu Wort.



### **Alexander Fuhrmann:**

Meine Priorität lag darin, die grundlegenden Funktionen für die Datenbankanbindung zu schreiben. Da wir per JS Request aus der App auf den Server zugreifen mussten, mussten alle Funktionen in PHP geschrieben werden und mit der MySQL Datenbank kommunizieren. Dafür habe ich mir zuerst eine Datenbankstruktur ausgedacht, und dann eine PHP Klasse geschrieben mit den essentiellen Funktionen wie „Login“, „Registrieren“, „Klasse wählen“ und so weiter, um die Möglichkeit zu haben mit Usern umzugehen. Nachdem klarer wurde welchen Umfang wir anbieten wollen folgten Funktionen die für das Inventar wichtig waren: „Items speichern“, „Items versenden“, „Items zuweisen“, „Items löschen“ und kurz darauf die analogen Funktionen für die Quests. Bei den Items und den Quests war es wichtig immer zu selektieren welchem Charakter und welchem Team der User zugeordnet ist, also habe ich zentrale interne Funktionen eingebaut um immer vorher selektieren zu können ob der User für die jeweiligen Items / Quests in Frage kommt.

Die Items waren unterteilt in die Klassen „weapon“, „item“, „money“, das genügte um die jeweiligen Funktionen den Items zuzuordnen.

Die Quests waren nach den Teams unterteilt, wobei alle Teams die Quests der „Verschwörungstheoretiker“ annehmen konnten, allerdings für die Quests der Technokraten und Avantgardisten das jeweilige Team zugeordnet sein musste.

Zum Schluss implementierte ich ein Code-System mit dem man 4-stellige Codes an den Server senden konnte und daraufhin eine von 4 Aktionen auslöst:

- Item in das Inventar aufnehmen
- Quest annehmen,
- Quest ablehnen,
- Team beitreten

Jeder Code war pro User nur einmal ausführbar. Mit diesen Codes konnten die Spielleiter das Spielgeschehen steuern und für den User greifbar machen.

Da nun alle Funktionen vorhanden waren, brauchten wir noch eine Administratoroberfläche, damit man schnell die Items, Quests und User bearbeiten, hinzufügen und löschen konnte. Diese schrieb ich als schnelle HTML Seite mit PHP-Scripts zur Datenbankbindung und verlinkte die jeweiligen Quests und Inventaritems mit den Usern und fügte noch die Code-erstellung hinzu mit der man schnell neue Codes für alle wichtigen Aktionen generieren konnte. So konnten die Personen die verantwortlich für beispielsweise Quests oder die Waffen waren, ihren Input über diese Oberfläche direkt in die App einfügen verändern und updaten.

### **Christopher von Bargaen:**

Das Back-end der App wurde mit JavaScript und mit Hilfe von „jQuery“ programmiert. Es beinhaltet sämtliche Funktionalitäten und realisiert die Kommunikation mit der Datenbank. Da keine Vorkenntnisse in Bezug auf JavaScript vorhanden waren, wurde eine umfassende Einarbeitung in die Programmiersprache notwendig. Insgesamt ist JavaScript und „jQuery“ aber relativ leicht zu erlernen. Die Hauptaufgaben der Scripts bestehen darin, Interaktionen zu ermöglichen, also auf Eingaben der Benutzer zu reagieren, diese zu verarbeiten und gegebenenfalls an die Datenbank weiterzuleiten. Ebenso werden umgekehrt auch Ausgaben der Datenbank ans Front-end übermittelt.

Wählt ein Nutzer beispielsweise eine Charakterklasse oder gibt einen Quest Code ein, muss dies der Datenbank mitgeteilt werden um die dortigen Einträge zu aktualisieren. Wird einem Spieler zum Beispiel Schaden zugefügt oder Geld abgenommen, werden die Werte (von einem Administrator) zunächst in der Datenbank aktualisiert und dann via JavaScript in der App aktualisiert.

### **Bager Bingöl und Christoph Thiel:**

Die Oberfläche der „Companion App“ wurde mithilfe von HTML & CSS realisiert. Als Hilfe kam ebenfalls das Bootstrap Framework zum Einsatz. Es gab mehrere Designziele bei der Entwicklung:

- Übersichtlichkeit
- Gute Bedienbarkeit
- Anpassung an verschiedene Display Größen
- Minimalistische Präsentation

Am Anfang des Prozesses stand die Einarbeitung in verschiedenste Templates und die Untersuchung auf praktische Anwendungsmöglichkeiten für die oben genannten Ziele.

Nach mehreren Anläufen entschieden wir uns für ein Template welches uns zusätzliche erweiterte Funktionalität bot, um unsere dynamischen Elemente bezüglich des Spielcharakters flexibel darstellen zu können.

Die Farbgebung und die genaue Anordnung der Elemente wurden in gemeinsamer Absprache mit dem Systopia-Team eingebunden und abgestimmt.

Der letzte Feinschliff und Optimierungen wurden ebenfalls in Absprache mit dem restlichen Team gefunden.

### 3. Projektauswertung

#### 3.1 Projektablauf

Für den Großteil des Semesters war dieses Vorhaben eines der größten, wenn nicht sogar das umfangreichste Projekt, an dem sie bisher mitgewirkt haben. Dasselbe gilt für uns als App-Gruppe. Selbstverständlich gab es, besonders anfangs, als der wesentliche Teil des Projekts noch undefiniert war, Startschwierigkeiten. Aber so gut wie jedes Problem konnte mit ein wenig Zeit und Mühe gelöst werden. In regelmäßigen Intervallen haben wir uns getroffen, um einander über den Stand der jeweiligen bearbeiteten Komponenten zu informieren. Die viele Kommunikation war definitiv nötig, da besonders zum Ende hin, wo die verschiedenen Komponenten zusammenarbeiten mussten, hohe Koordination erforderlich war.

Natürlich hatte jeder von uns bei seiner Arbeit individuelle Probleme, wie Christopher Von Barga zum Beispiel:

So leicht auch der Einstieg in und die Umsetzung mit JavaScript waren, umso schwieriger war der Umgang mit PhoneGap-Plug-Ins. PhoneGap bietet die Möglichkeit, von anderen Programmierern entwickelte Plug-Ins, also kleine funktionelle Bausteine, in jedes Projekt zu integrieren. Für unsere App war von Anfang an ein QR-Reader gewünscht, der über ein solches Plug-In realisiert werden sollte. In der Theorie sollte der fertige Programmcode in das dafür vorgesehenen Plug-In-Verzeichnis kopiert werden und dann über wenige Codezeilen an den benötigten Stellen in die App integriert werden. In der Praxis hat dies allerdings nicht funktioniert. Da die Dokumentation der Plug-In-Funktion von PhoneGap keine besondere Hilfe darstellte, hat sich die Fehlersuche als schwierig herausgestellt und war nicht erfolgreich. Nach etwa 16 Stunden inklusive Einarbeitung in Plug-Ins wurde die Integration des QR-Readers aufgegeben. Die Realisierung eines QR-Readers ohne die Plug-In-Funktion zu nutzen wurde ebenfalls versucht, aber aus Zeitmangel zum Ende des Projekts nicht mehr umgesetzt. Um den fehlenden QR-Reader, der zum Beispiel

auch für den Quest-Abschluss benötigt wurde, zu ersetzen, wurden die Quest-Codes eingeführt. Statt eines QR-Codes einzulesen mussten die Spieler also einen vierstelligen Zahlencode eingeben.

Aus diesem Projekt nehmen wir in die Zukunft die gesammelten Erfahrungen, Probleme und Ihre Lösungen, Ideen und neue Kenntnisse mit. Es ist unwahrscheinlich, dass wir am Quellcode selbst noch weiter arbeiten, da das Projekt „Systopia“ wohl eine einmalige Sache war. Da in der Zukunft aber wahrscheinlich noch einige ähnliche Projekte auf uns zukommen werden, sind diese Erfahrungen sehr wertvoll.

### 3.2 Tatsächlicher Aufwand (Beispiel Bager Bingöl)

Datum	Aufgabe	Stunden	€
04.04.2016	Teambesprechung	4	60 €
08.04.2016	Einarbeiten in HTML	8	120 €
09.04.2016	Einarbeiten in HTML	8	120 €
10.04.2016	Anfang der index.html	4	60 €
11.04.2016	Einarbeiten in GIT(hub)	4	60 €
13.04.2016	Fertigstellung der index.html und Anfang der faction.html	8	120 €
15.04.2016	Fertigstellung der faction.html	6	90 €
18.04.2016	Teambesprechung	4	60 €
22.04.2016	Neuanfang der index.html mit Bootstrap	10	150 €
25.04.2016	Fertigstellung der index.html mit Bootstrap	10	150 €
29.04.2016	bisherigen Stand der App vorstellen	3	45 €
03.05.2016	Ausprobieren verschiedener Designs für die index.html	8	120 €
09.05.2016	Festlegung des Designs für die index.html	4	60 €
12.05.2016	Bugfixing der index.html	4	60 €
13.05.2016	Neuanfang der index.html mit HTML5	8	120 €
16.05.2016	Fertigstellung index.html und Neuanfang classes.html	4	60 €
17.05.2016	Fertigstellung classes.html und faction.html.	10	120 €
18.05.2016	Erstellung der Intro-Seiten und sonstigen Seiten	10	120 €
19.05.2016	Erstellung der Intro-Seiten und sonstigen Seiten	14	210 €
20.05.2016	Neuanfang der character.html	12	180 €
21.05.2016	Bugfixing	10	150 €
23.05.2016	Inventarfunktion hinzugefügt	10	150 €
24.05.2016	Avantgardisten- und Technokratenbeschreibung hinzugefügt	10	150 €
26.05.2016	Fertigstellung von allen Seiten und Bugfixing	10	150 €
		183	2.685 €

#### **4. Quellenverzeichnis**

*Pip-Boy 3000*. (2016). Abgerufen am 28. 1 2017 von Wikia.com:

[http://de.fallout.wikia.com/wiki/Pip-Boy\\_3000](http://de.fallout.wikia.com/wiki/Pip-Boy_3000)

*Studienführer*. (2017). Abgerufen am 2017 von haw-hamburg.de:

[https://www.haw-hamburg.de/fileadmin/user\\_upload/DMI-Mt/Studium/stud\\_fuehMT.pdf](https://www.haw-hamburg.de/fileadmin/user_upload/DMI-Mt/Studium/stud_fuehMT.pdf)



## 5. Abbildungsverzeichnis

Abbildung 1: Avantgardisten- (links) und Technokratenlogo (rechts)	3
Abbildung 2: Pip-Boy 3000	4
Abbildung 3: Login-Seite	5
Abbildung 4: Klassenauswahl	5
Abbildung 5: Menüleiste	6
Abbildung 6: Inventarbildschirm (rechts mit Item-Beschreibung)	6
Abbildung 7: Questbildschirm	7
Abbildung 8: Raumplan Studienführer Media Systems (Studienführer, 2017, S. 9)	8
Abbildung 9: Minimap (Gesamtmap)	8