**Abschlussprüfung Sommer 2016**

Fachinformatiker für Anwendungsentwicklung  
****Dokumentation zu betrieblichen Projektarbeit

**Webseite zur Diablo 3-Datenabfrage**

Entwicklung einer Webseite zur Abfrage   
von Diablo 3 Spiele-Daten

Abgabetermin: 09.09.2016

**Ausbildungsbetrieb:**  
Berufsbildungswerk Neckargemünd  
Im Spitzerfeld 25  
69151 Neckargemünd

**Prüfungsbewerber**  
Tobias Scholl  
Paul-Hindemith-Ring 7  
76669 Bad Schönborn

**Inhaltsverzeichnis**

[Abkürzungs- und Stichwortverzeichnis 3](#_Toc461025317)

[Begriffserklärungen 3](#_Toc461025318)

[1. Einleitung 4](#_Toc461025319)

[1.1. Projektbeschreibung 4](#_Toc461025320)

[1.2. Projektziel 4](#_Toc461025321)

[1.3. Projektumfeld 4](#_Toc461025322)

[1.4. Projektbegründung 4](#_Toc461025323)

[1.5. Projektschnittstellen 5](#_Toc461025324)

[1.6. Projektabgrenzung 5](#_Toc461025325)

[2. Projektplanung 5](#_Toc461025326)

[2.1. Projektphasen 5](#_Toc461025327)

[2.2. Projektkosten 6](#_Toc461025328)

[3. Projektablauf 6](#_Toc461025329)

# Abkürzungs- und Stichwortverzeichnis

API Application Programming Interface

GitHub Desktop Desktop-Version des

SASS/SCSS Syntactically Awesome Stylesheet (CSS Erweiterung)

Prepros Preprocessor (um SASS in gewöhnliches CSS zu wandeln)

UX User Experience (Nutzererfahrung, nutzerfreundliche Gestaltung)

# Begriffserklärungen

|  |  |
| --- | --- |
| **Begriff** | **Erklärung** |
| Charakter | die erstellte Spielfigur |
| Hardcore Charakter | spezieller Spielmodus, bei dem der Charakter nach einem Tod nicht wiederbelebt werden kann. |
| Klasse | eine Klasse ist mit einem Beruf zu vergleichen und setzt fest, welche Fertigkeiten und Waffenklassen der Charakter tragen kann. |
| Item / Ausrüstung | Ein Gegenstand im Spiel. Kann zum Beispiel eine Waffe, ein Ausrüstungsteil oder ein Edelstein sein. |
| Paragonstufe | Ähnlich des Charakterlevels, Wird erhalten, sobald der Charakter die Maximalstufe (70) erreicht hat. |
| Fertigkeiten | Fertigkeiten sind Kräfte, die der Charakter bei bestimmten Levelstufen freischaltet. |
| Edelsteine / Sockelplätze | Edelsteine sind Items, das in eine Waffe oder Rüstungsgegenstand mit einem Sockelplatz eingesetzt werden kann. Ein Sockelplatz ist wie eine Edelsteinfassung. |
| Saisons | Ein bestimmter Zeitraum (~ 3 Monate) in dem ein neuer Charakter erstellt kann, um sich mit anderen Saison-Spielern zu messen (ähnlich einer Fußball-Saison).  Ein Saison-Charakter fängt komplett neu an. |

# Einleitung

Die folgende Projektdokumentation schildert den Ablauf des HK-Abschlussprojekts, welches der Autor im Rahmen seiner Ausbildung zum Fachinformatiker für Anwendungsentwicklung durchgeführt hat. Ausbildungsbetrieb ist die Berufsbildungswerk Neckargemünd GmbH (BBW). Die IT-Abteilung des Berufsbildungswerks umfasst über 100 Mitarbeiter und bildet unter anderem Informatikkaufleute, Fachinformatiker für Anwendungsentwicklung und Fachinformatiker für Systemintegration aus.

# Projektbeschreibung

Die Firma XY ist ein Unternehmen, das eine Spieleplattform-Webseite betreibt. Die Webseite veröffentlicht Inhalte wie News und Anleitungen zu verschiedenen Computerspielen. Um die Benutzerzahlen auf der Webseite zu erhöhen, möchte man ein weiteres Spiel hinzufügen.

# Projektziel

Der Spieleplattform-Webseite soll das Computerspiel „Diablo 3“ hinzugefügt werden, jedoch in größerem Umfang als die gewöhnlichen News und Anleitungen. Die Nutzer sollen mit ihrem Account-Namen (so genanntem BattleTag) bestimmte Daten ihrer Charaktere abfragen sowie statistische Werte wie Schaden, Zähigkeit, Erholung und Leben mit anderen Charakteren ihres Accounts vergleichen können.

# Projektumfeld

Die Hardware für das Projekt wird von der Firma zur Verfügung gestellt. Es wird ein Windows 7 64Bit PC mit der Serverumgebung XAMMP zur Verfügung gestellt. Zusätzlich genutzte Programme sind die kostenlosen Entwicklungstools Atom (Editor), Prepros (SASS Preprocessor) und GitHub.

# Projektbegründung

Das Projekt soll mehr Nutzer auf die Webseite locken, um dadurch höhere Nutzerzahlen und mehr Umsatz durch geschaltete Werbung zu erwirtschaften.

# Projektschnittstellen

Die Anwendung wird über die Blizzard-API mit den Servern von Blizzard Entertainment ® (der Entwicklungsfirma von Diablo 3) Daten abfragen.

# Projektabgrenzung

Im Projekt wird **keine** komplette Spieleplattform-Webseite, sondern lediglich eine Webseite mit API-Abfrage und –Ausgabe der Account- und Charakterdaten entwickelt und designt.

Ebenfalls wird die Webseite nicht ausgiebig auf Browserkompatibilität getestet und es wird kein Responsive Design erstellt.

# Projektplanung

# Projektphasen

Das Projekt wird in den zwei Arbeitswochen vom 29. August bis zum 09. September 2016 realisiert. Die Projektzeit beträgt 70 Stunden inklusive Dokumentation, Kundeneinweisung und Abnahme.

|  |  |  |
| --- | --- | --- |
| **Projektphase** | **Aufgabenbeschreibung** | **Soll-Stunden** |
| **Vorbereitung** | Informieren über den Aufbau und die Nutzung der Blizzard-API für Diablo 3 | 6 Stunden |
| **Umsetzung** | Grundgerüst der Oberfläche in HTM planen und erstellen | 4 Stunden |
| Planen und Designen der Oberfläche mit CSS | 8 Stunden |
| Implementieren der API-Abfragen mit PHP | 7 Stunden |
| Entwickeln der Charakter-Übersicht mit PHP | 10 Stunden |
| Erstellen des Charakter-Vergleichs mit PHP | 7 Stunden |
| **Abschluss** | Testen der Webseitenkompatibilität in verschiedenen Browsern | 4 Stunden |
| Erstellung der Dokumentation | 24 Stunden |
| **Gesamtzeit** | | **70 Stunden** |

Tabelle 2: Phasen der Projektarbeit

# Projektkosten

Die Projektkosten werden mit Pauschalsätzen berechnet, die für Personal- und Ressourcenkosten anfallen. Der Stundensatz eines Auszubildenden im 3. Lehrjahr beträgt 10 €, der Stundensatz der Ressourcen (z.B. Strom, Hard- und Softwarenutzung) wird auf 15€ festgesetzt. Als Entwicklungsressourcen wurden ein PC mit Windows 7 64Bit und die Entwicklungstools Git Desktop, XAMMP, Prepros und Atom Text Editor verwendet.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Vorgang** | **Zeit** | **Personalkosten** | **Ressourcenkosten** | **Gesamtkosten** |
| Entwicklung | 70 h | 700,00 € | 1050,00 € | 1750,00 € |

# Projektablauf

Tabelle 1: geschätzte Projektkosten

# Vorbereitung

Um den Aufbau der API von Blizzard zu verstehen, habe ich zuerst mithilfe einer Demo-Webseite von Blizzard grundlegende Abfragen zu Account-, Charakter- und Item-Daten getätigt. Somit habe ich mir einen Überblick über die nötigen Methoden wie „file\_get\_contents“ (Inhalt einer Webseite kopieren) und „json\_decode“ (Text in ein JSON-Array speichern) in PHP verschafft. Für diese Vorbereitung wurde weniger Zeit als geplant benötigt, da die Demo-Webseite inklusive Dokumentation seitens Blizzard sehr verständlich ist (mit Ausnahme einer Sache, dazu aber später mehr).

# Landingpage

Als zweiten Schritt habe ich eine Startseite (sog. Landing Page) mit HTML und CSS entwickelt. Diese Seite soll veranschaulichen, welche Funktionen die Webseite beinhaltet. Diese Idee hatte ich erst nachträglich und wurde mit der gewonnenen Zeit aus der Vorbereitung (ca. eine Stunde) realisiert.

# Header und Footer

Der dritte Schritt war nun, ein Grundgerüst für Header und Footer in HTML und CSS zu schreiben. Dieses bleibt auf jeder weiteren Seite identisch, sodass die Nutzererfahrung (UX) ebenfalls gleich bleibt.

Das CSS für den Header und Footer wurden mithilfe SCSS in eigene Dateien geschrieben, sodass diese einfacher gewartet und geändert werden können. Diese Dateien werden dann in einer SCSS-Hauptdatei importiert und mithilfe von Prepros in standardisiertes CSS „übersetzt“.   
Da ich in einem kurz zuvor absolvierten Praktikum sehr viel über HTML5, CSS3 und deren konforme Umsetzung gelernt habe, wurde auch an diesem Projektpunkt weniger Zeit (ungefähr 2 Stunden) als geplant benötigt.

# erste API-Abfrage

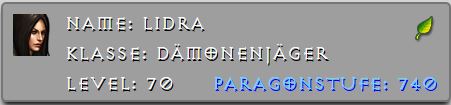
Nachdem das Grundgerüst stand habe ich begonnen die ersten API-Abfragen für den Account zu erstellen. Diese Request-URLs setzen sich aus mehreren Teilen zusammen: der **statische battle.net Link**, der **BattleTag** (ausgelesen aus dem Input-Feld), die **Lokalisierung** und der **API-Key**. Beispiel:  
**https://eu.api.battle.net/d3/profile/Hurric4ne-2268/?locale=de\_DE&apikey= 27hd5khrafctn6dv8g9egtbt25qbzcqz**Teile der von der API gesendeten Daten werden dann dynamisch mit HTML und CSS in Charakter-Containern untergebracht.

Bild 1: Charakter-Container

Der Charakter-Container ist anklickbar, um weiter auf die Charakter-Übersicht zu gelangen. Hier bin ich auf das erste Problem gestoßen, da die Links des Containers dynamisch sein müssen.

**Problem:** Um auf der Charakter-Übersicht die richtigen Daten zu erhalten und anzuzeigen, muss die Request-URL des richtigen Charakters auf die nächste Seite übertragen werden.

**Erste Lösungsidee:** Die Request-URL wird komplett an die vorhandene PHP-URL angefügt. Allerdings wird so die URL sehr lange, da die URL dann aus zwei Links besteht. Ein größeres Problem ist allerdings, dass durch diese Lösung der API-Key im Klartext in der URL steht. Dieser Key ist zwar nicht wie ein Passwort, Fremde könnten allerdings damit ebenfalls API-Requests durchführen.

**Zweite Lösungsidee:** Nur die Charakter-ID wird an die PHP-URL angefügt. Die Request-URLs aller Charaktere werden in ein Session-Array gespeichert, bei dem der Array-Index der Charakter-ID entspricht. So lässt sich mit $\_SERVER[“REQUEST\_URI“] die Charakter-ID auf der Folgeseite wieder auslesen und als Index für das Session-Array genutzt werden.

1. **Item-Links mit richtigen Werten für Mouseover**

**Problem:** Aufgrund ungenauer Beschreibung der Blizzard API-Dokumentation war unklar, wie genau ein Link für ein Item auszusehen hat, der von der Tooltips.js (sorgt für Mouseover) richtig interpretiert wird. Standartwerte von Items werden mit dem Item-Namen generiert, z.B.: „http://eu.battle.net/d3/de/item/**Item-Name**“

**Lösung:** Damit die Tooltips.js die Werte des Items als Mouseover richtig erstellt, muss anstatt des Item-Namens der Item-Hashwert an die URL angefügt werden, z.B.: „http://eu.battle.net/d3/de/item/**Item-Hashwert**“

1. **Positionierung von Item-Icons aufgrund verschiedener Abmessungen**

**Problem:** Verschiedene Item-Icons von Brustteilen sind unterschiedlich groß, sodass eine generelle Positionierung des IMG-Tags zuerst nicht möglich war. **Lösung:** Änderungen an der Positionierung des IMG-Tags in Kombination mit der CSS-Eigenschaft „overflow:hidden“ hat dieses Problem gelöst.

1. **Leere Links für Skills**

**Problem:** Wenn ein Charakter einen Slot in der Fertigkeiten-Leiste noch nicht freigeschaltet hat oder keine Fertigkeit eingesetzt hat, entstehen leere IMG- und Anker-Tags.

**Lösung:** Mit einem Isset-If wird zuerst abgefragt, ob eine Fertigkeit gesetzt ist. Wenn Ja, wird der API-Request ausgeführt und der Anker- und IMG-Tag erstellt und mit den Request-Daten (Link und Bild-URL) gefüllt, ansonsten wird der Vorgang übersprungen und es wird zur nächsten Fertigkeit gesprungen.

**Problem:**

Set-Waffen und Nebenhand-Items haben keine Sockel im JSON-Code.

**Keine Lösung!**

Bildquelle:

http://www.diablo-3.net/wp-content/uploads/2014/07/diablo3-reaper-of-souls-logo\_seite.png