

Dokumentation

der Projektarbeit *löffel-abgeben.de* im Modul Einführung in Multimedia

Projektteam:

Carolina Track 571265

Aaron Jacob 570648

Susann Schöbel 571657

Dozierende:

Prof. Dr.-Ing. Johann Habakuk Israel

Katrin Lang

Inhaltsverzeichnis

0 Einleitung	3
1 Ideenfindung und Konzept	3
2 Verworfen und umgesetzte Features	6
3 Schwierigkeiten und Lösungsansätze	9
4 Prototypen	10
5 Evaluationen	11
6 Zusammenfassung und Ausblick	16
7 Quellenangaben	17

0 Einleitung

Im Rahmen der Projektarbeit im Modul *Einführung in Multimedia* wurde eine interaktive multimediale Anwendung unter dem Namen *löffel-abgeben.de* erstellt, welche Rezepte im txt-Format einliest und daraus unter Verwendung einer eigenen Datenbank Kochvideos im mp4-Format generiert. Der Anwendungscode sowie weitere Projektdokumente sind frei einsehbar unter <https://github.com/CarolinaTrack/MuMe/>.

Dieses Dokument dient der Dokumentation des Arbeitsprozesses und des Arbeitsergebnisses.

Es hält den Verlauf der Ideenfindung und die Evolution des Konzeptes fest, die im Projektzeitraum umgesetzten und verworfenen Funktionen, die dabei aufgetretenen Schwierigkeiten und Probleme sowie deren Lösungsansätze, die angefertigten Prototypen und die durchgeführten Evaluationen. Am Ende des Dokumentes werden diese Punkte nochmals zusammengefasst.

1 Ideenfindung und Konzept

Ideenfindung

In der Gruppenfindungsphase gab es zwei verschiedene Ideen ein interaktives eBook zu erstellen: ein interaktives Kochbuch und ein interaktives Kinderbuch.

1. Kinderbuch

Grundidee: Ein elektronisches, illustriertes Kinderbuch mit Text. Der Text wird beim "Aufschlagen" der Seite laut vorgelesen oder kann übersprungen werden. Nachdem der Text der Seite vorgelesen wurde, kann das Kind frei auf Symbole oder Gegenstände auf der Seite klicken, wodurch Reaktionen ausgelöst werden.

2. Kochbuch

Grundidee: Ein möglichst barrierearmes Kochbuch mit multimedialen Anleitungen und Veranschaulichungen. Jedem Rezept/Kapitel vorweg steht eine kleine Einleitung mit wissenswerten Informationen zu dem Gericht, den benötigten Geräten und/oder den Zutaten, die verwendet werden. Bei den Rezepten soll besonders auf Kleinschrittigkeit

und Verständlichkeit in der Darstellung geachtet werden. Die Arbeitsschritte werden in Textform dargestellt, die man auf Wunsch laut vorlesen lassen kann und komplizierte Schritte werden durch begleitende Videos erläutert.

Nachdem sich die Gruppe zusammengefunden hatte, einigten wir uns auf die zweite Idee.

Prototyp-Phase

Nach der Einigung auf eine Idee erstellten wir unabhängig voneinander unsere ersten Papierprototypen. Anhand dieser besprachen wir alle Features und welche davon wir umsetzen wollten.

Ein Rezept im eBook sollte folgendes umfassen:

- Trivia/"Wussten Sie schon?"-Bereich
- ggf. Bilder der Zutaten/Küchenutensilien
- Umrechnen der Maßeinheiten
- alternative Zutaten
- Farblegende für einzelne Funktionalitäten
- Timer
- Miniaturansicht der Features/Ausklappfunktion
- Möglichkeit, ein Lesezeichen setzen

Nach interner Besprechung des Prototypen nahmen wir am 11.06. an der Sprechstunde mit Herr Prof. Israel teil und stellten unsere bisherige Arbeit vor. Es kristallisierte sich heraus, dass wir unsere Idee noch einmal komplett überarbeiten müssten.

Nach reiflicher Überlegung entschieden wir uns dafür die eBook-Idee zu verwerfen und Kochvideos zu erstellen. Erneut erstellten wir jeweils eigene Prototypen und besprachen diese in der Gruppe.

Als Ergebnis entstand folgende Idee:

3. Kochvideo nach Erstellen eines Rezeptes

Grundidee: Der/die NutzerIn kann auf einer Website sein Rezept einlesen. Dazu müssen die Zutaten und Kochanweisungen einzeln eingegeben werden. Nachdem alles eingegeben wurde sucht das Programm im Hintergrund nach entsprechenden Videos in einer Datenbank. Aus diesen Videos kann der/die NutzerIn für sich passende Videoerklärungen zu seinem Rezept ein Kochvideo erstellen.

Wenn es kein passendes Video in der Datenbank gibt kann der/die NutzerIn auswählen, dass die Anweisung in Textform in das Video eingebunden wird.

Nach Erstellen des Videos kann der Nutzer dies herunterladen und abspielen.

Am 15.06. hatten wir die Möglichkeit, unsere Idee im Rahmen einer Projektbesprechung erneut vorzustellen. Aus der eBook-Idee wurde nun endgültig das Projekt zur Erstellung von Kochvideos.

Implementierung

Bei Beginn dieser Phase fiel uns recht schnell auf, dass wir uns in der verbleibenden Zeit nur auf die Implementierung der Kernfeatures konzentrieren können. Wir entschlossen uns, den Algorithmus zur Videoerstellung, das Auslesen des Textdokuments, sowie den Suchalgorithmus zur Schlagwortsuche in der Datenbank, sowie die Datenbank selbst zu implementieren.

Da unser Algorithmus die einzelnen Kochanweisungen aus einem Textdokument nach und nach auf Schlagworte untersucht und diese mit der Datenbank abgleicht, hielten wir die Fehleranfälligkeit für zu hoch, als dass der/die NutzerIn die Rezepte selbst eintragen könnte. Wir kamen zu dem Schluss, dass die Rezepte schon auf der Seite verfügbar sein müssen. Der/die NutzerIn hat dennoch die Möglichkeit, eigene Rezepte einzureichen. Diese müssen vor dem Umwandeln in ein Kochvideo in eine für den Algorithmus passende Formatierung überführt werden.

Wir entschieden uns dazu, unser Projekt ein letztes Mal zu überarbeiten.

4. Kochvideos auf Basis von Bildern und Videos

Grundidee: Über eine Website lassen sich Kochvideos erstellen. Die Rezepte können von der Community vorgeschlagen werden. Damit diese für den Algorithmus nutzbar sind, müssen die Vorschläge unserer Vorlage entsprechend nachbereitet werden. Anschließend werden diese hochgeladen und sind in der Rezeptausswahl für die ganze Community verfügbar.

Bereits hochgeladene Rezepte lassen sich auswählen und die Kochvideos dazu können heruntergeladen werden. Im Hintergrund wird zum Erstellen des Videos die Datenbank mit Schlagwörtern aus den einzelnen Kochanweisungen durchsucht. Wenn keine Datei in der Datenbank gefunden wird, wird die Kochanweisung in Textform im Video eingeblendet.

Bei den Bildern und Videos kann die Community sich aktiv einbringen, sodass nach und nach mehr Bilder und Videos in die Datenbank eingebunden werden können.

2 Verworfenne und umgesetzte Features

Verworfenne Features

Leider musste zeitbedingt auf mehrere Features verzichtet werden. Die wichtigsten Features wurden herauskristallisiert, um trotz Zeitdruck eine lauffähige Anwendung entwickeln zu können. Eines der größten Features, für das die Zeit nicht gereicht hat, ist die Webseite, die nicht implementiert werden konnte. Stattdessen besteht das Produkt der ursprünglichen Anforderung entsprechend als Processing-Anwendung.

Website

Eine Website auf der zum einen Rezepte hochgeladen und zum anderen die automatisiert erstellten Kochvideos heruntergeladen werden können. Hierzu gehört ein simpler Upload- und Download-Button, der Kommunikation, sowie Austausch von Daten zwischen UserInnen und Server möglich macht.

Umgesetzte Features:

Alle Features wurden mit Hilfe der auf Java basierenden Programmiersprache Processing (Version 3.5.4) umgesetzt.

Folgende Begrifflichkeiten müssen an dieser Stelle verständnishalber geklärt werden:

Array: Ein Array ist eine Aneinanderreihung von Elementen eines festen Datentyps.

Datentyp: Der Datentyp gibt an, von welcher Art die Daten sind, die mit ihm beschrieben werden, und welche Operationen auf diesen ausgeführt werden können.

Auslesen des Textdokuments

Das erste Feature, das umgesetzt wurde, stellt das Auslesen des Rezeptes in Form eines Textdokumentes mit der Endung .txt dar. Hierfür wird der Inhalt des Dokuments zeilenweise in ein String-Array geschrieben, d.h. die Zeilen des Rezeptes werden einzeln extrahiert und zur weiteren Bearbeitung als String Datentyp zwischengespeichert [vgl. WWW1]. Anschließend wird nach dem Schlüsselwort "Zubereitung" gesucht, da die darauffolgenden

Kochanweisungen eingelesen werden sollen [vgl. WWW2]. Um dies zu bewerkstelligen, wird eine einheitliche Formatierung benötigt. Eine Kochanweisung darf maximal eine Zeile umfassen und ab dem fünften Zeichen der Zeile beginnen. Ist dies der Fall, können nun die ersten zwei Wörter jeder Zeile, Zeichen für Zeichen, als Suchbegriffe in einem neuen String-Array gesammelt werden. Dieses neue String-Array ist nun die Basis für alle weiteren Operationen.

Datenbank

Im Rahmen der Projektarbeit wurde mit einer PostgreSQL-Datenbank, installiert auf einer Virtual Machine mit dem Betriebssystem Ubuntu (Version 1804.2019.521.0), gearbeitet. Kurze Videos von einzelnen Kochanweisungen sollen der Datenbank mit Hilfe von JDBC-Standards hinzugefügt werden. JDBC ist eine Datenbankschnittstelle der Java-Plattform, die eine einheitliche Schnittstelle zu Datenbanken verschiedener Hersteller bietet und speziell auf relationale Datenbanken ausgerichtet ist. Um in Processing mit JDBC arbeiten zu können, müssen folgende Bibliotheken importiert werden:

- `java.sql.Connection;`
- `java.sql.DriverManager;`
- `java.sql.ResultSet;`
- `java.sql.Statement;`
- `java.sql.PreparedStatement;`
- `java.sql.SQLException;`
- `javax.swing.filechooser.FileSystemView;`
- `processing.video.*;`
- `java.io.*;`

Nach dem Herstellen einer Verbindung zur Datenbank, kann ein Kochvideo im mp4-Format über einen File-Input-Stream in einen Binary Stream umgewandelt werden. Der Binary bzw. Byte-Strom wird benötigt, da die Daten erst in dieser Form in der Datenbank gespeichert werden können. In PostgreSQL wird der Datentyp Bytea für große Objekte wie Videos oder Fotos genutzt und kommt auch hier zum Einsatz. Der Bytestream wird also in einer Tabelle mit einer Spalte vom Datentyp Bytea abgespeichert und erhält in einer weiteren Spalte derselben Zeile noch einen Dateinamen [vgl. WWW3]. Der Dateiname besteht genauso wie die gespeicherten Suchbegriffe aus zwei Wörtern.

Suchalgorithmus zur Schlagwortsuche in der Datenbank:

Die gespeicherten Suchbegriffe sollen genutzt werden, um passende Kochvideos in der Datenbank zu finden und zu extrahieren. Hierfür werden folgende Bibliotheken benötigt:

- `java.sql.Connection;`
- `java.sql.DriverManager;`
- `java.sql.ResultSet;`
- `java.sql.Statement;`
- `java.sql.PreparedStatement;`
- `java.sql.SQLException;`
- `javax.swing.filechooser.FileSystemView;`
- `java.io.*;`

Sind die Bibliotheken importiert, kann die Tabelle mit den Kochvideos nach den Suchbegriffen durchsucht werden [vgl. WWW4]. Hierfür werden Queries, dh. Anfragen an die Datenbank in der Sprache SQL formuliert und mit Hilfe von JDBC in Processing eingebettet. Wird ein Video gefunden, wird der Byte-Strom in ein Byte-Array geschrieben, welches zu einem Byte-Array-Input-Stream umgewandelt wird. Anschließend wird eine neue Datei erstellt, welche den Zielort für eine temporäre Datei bildet. Mit Hilfe dieser Datei kann der Input-Stream lokal (hier auf der Virtual Machine) gespeichert werden. Jetzt kann mit dem in der Datei gespeicherten Stream ein Processing eigenes Movie-Objekt erstellt werden, welches ermöglicht, das extrahierte Video im Sketch, dem graphischen Ausgabefenster von Processing, anzusehen und weiter zu bearbeiten. Zuletzt wird das Movie-Objekt in einem Array gespeichert.

Algorithmus zur Videoerstellung

Um die einzelnen extrahierten Kochvideos zusammen fügen zu können, werden folgende Bibliotheken gebraucht:

- `processing.core.PApplet;`
- `com.hamoid.*;`
- `processing.video.*;`

Bei der Suche nach den Videos in der Datenbank wurden zwei Arrays erstellt. Das eine beinhaltet die gefundenen Videos und das andere die Suchbegriffe zu denen kein Video gefunden wurde. Die beiden Arrays sind so sortiert, das die Reihenfolge der Kochanweisungen immer noch beachtet wird. Jetzt können die Videos aus dem Array nacheinander abgespielt werden [vgl. WWW5]. Für jedes Video wird abhängig von seiner

Länge eine Zeitspanne definiert. Wird unter einem Index des Video-Arrays kein Video gefunden, bedeutet das, dass zuvor unter einem bestimmten Suchbegriff kein entsprechendes Video in der Datenbank gefunden wurde. Jetzt wird stattdessen mit Hilfe dieses Suchbegriffes wieder auf die ursprüngliche Kochanweisung zugegriffen. Um die Kochanweisung an Stelle eines Kochvideos im Sketch auszugeben, wurden vorher Schriftart, -größe und -platzierung, sowie eine Zeitspanne festgelegt, in der der Text dargestellt werden soll.

Zuletzt wird ein Video-Export-Objekt erstellt und als mp4-Datei abgespeichert. In das Objekt werden alle Frames von Beginn bis Ende des Sketches geschrieben, sodass alle Videos und Kochanweisungen vollständig und in richtiger Reihenfolge vorhanden sind.

3 Schwierigkeiten und Lösungsansätze

Eine der grundlegenden Herausforderungen war, dass wir in diesem Semester keine Möglichkeit hatten, uns während der Lehrveranstaltungen oder, wie sonst üblich, in den Pausen danach in der HTW zu treffen. Die Lehre auf Distanz und Social Distancing ließen uns nur die Möglichkeit über verschiedene Online-Tools zusammen zu arbeiten. So nutzten wir hauptsächlich Discord für regelmäßige längere Besprechungen sowie WhatsApp zum kurzen Klären unkomplizierter Angelegenheiten. GitHub war unser Tool der Wahl für die gemeinsame Arbeit am Code und Google Docs das zum gemeinsamen Arbeiten an dieser Dokumentation. Nichtsdestotrotz war es nötig, uns zum besseren Verständnis und gemeinsamen Brainstormen zu treffen. So entsprangen zwei intensiven Working Sessions die ersten Zeilen Code und weitere Ideen zur Implementierung.

Durch ein Missverständnis in der Kommunikation und der Aufgabenstellung zum Projekt haben wir in den ersten Phasen des Projektes an Ideen gearbeitet, welche dann verworfen wurden, um diese mehr der Aufgabenstellung anzupassen. Nachdem wir unsere erste Idee verworfen haben, haben wir uns häufiger Feedback abgeholt. Teilweise haben wir auch schon mit der weiteren Entwicklung begonnen, während wir noch auf eine Rückmeldung warteten.

Bedingt durch einen Denkfehler beim Arbeiten mit PDFs war es uns unmöglich, das Einlesen solcher Dateien zu implementieren, sodass wir uns dafür entschieden, auf txt-Dateien umzusteigen. Mithilfe dieser konnten wir dann weiterarbeiten und das Auslesen von Schlagwörtern für das Durchsuchen der Datenbank ermöglichen.

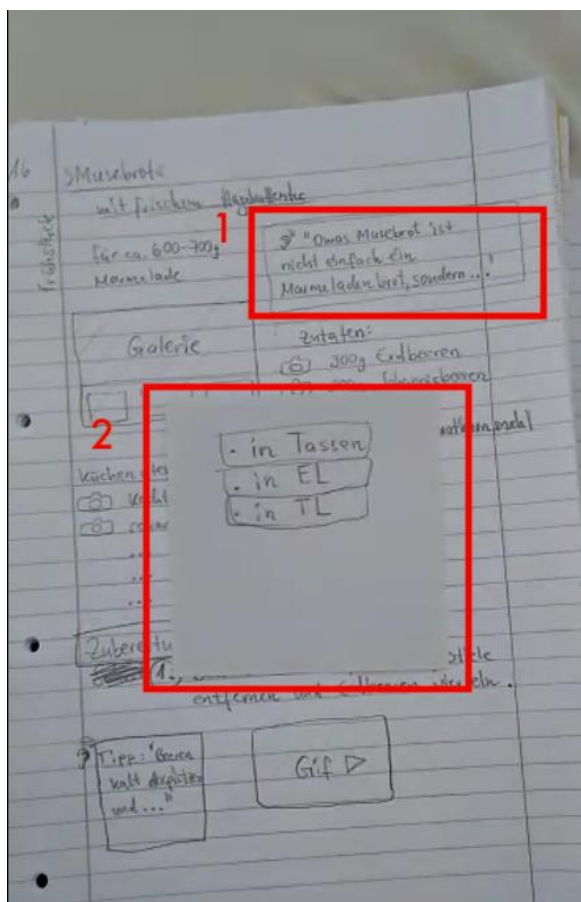
Leider ist die Abspielgeschwindigkeit der Videos noch nicht optimiert, sodass die mp4-Dateien derzeit in einer zu schnellen Geschwindigkeit gespeichert werden.

4 Prototypen

Im Laufe des Projektes wurden mehrere Prototypen angefertigt. Sie dienen der einfachen und vergleichsweise schnellen Veranschaulichung möglicher Nutzungsoberflächendesigns. Ein entscheidender Vorteil von Prototypen ist, dass sie billig produziert, bearbeitet und verworfen werden können.

Mit einer Ausnahme handelt es sich dabei um, wie in der Lehrveranstaltung vorgegeben, Papierprototypen. Für den Usability-Test mit einer Testperson wurde ein neuer Prototyp in digitaler Form angefertigt. Dieser wurde mit der Anwendung Keynote als Anwendung erstellt, d.h. als Präsentation mit Funktionalitäten. Sämtliche Prototypen sind als Videos in einem gesonderten Ordner in dem *Github-Repository* einsehbar, auf das zu Beginn der Dokumentation verlinkt wurde.

In diesem Abschnitt sind im Hinblick auf die Dokumentenlänge nur einige ausgewählte Ausschnitte der Prototypen als Screenshots abgebildet und erläutert.



- 1: Trivia/"Wussten Sie schon?"-Bereich
- 2: Umrechnen der Maßeinheiten

Abbildung 1: Auszug aus einem Papierprototypen einer früheren Entwicklungsphase mit 2 geplanten Features.

5 Evaluationen

Um sowohl unsere Prototypen als auch unsere Endanwendung auf Bedienbarkeit und Nutzerfreundlichkeit zu testen, haben wir verschiedene Evaluationen durchgeführt.

Peer-Review

Die Prototypen dienten dabei bereits als eine Möglichkeit zur kontinuierlichen Verbesserung, da jeder Prototyp von uns im Team durch Peer-Review beurteilt und anhand des Feedbacks nachgebessert wurde. Während dieses Prozesses ist eine Vielzahl von Papier-Prototypen entstanden.

Bei der Peer-Review hat jedes Teammitglied die Prototypen nach Aspekten beurteilt, die es introspektiv für sich selbst festgelegt hatte. Dies bedeutet, dass jede/r die Prototypen danach bewertet hat, ob er oder sie die Funktionen für sinnvoll, verständlich, und unter Umständen auch visuell ansprechend eingebracht hält, und darauf basierend konstruktives Feedback gegeben hat. Dies führte zu einem finalen Papier-Prototypen und einem finalen digitalen Prototypen, welche mithilfe weiterer Evaluationsmethoden untersucht wurden.

Usability Test

Eine dieser verwendeten Evaluationsmethoden ist der sogenannte Usability-Test. Dabei bekommt ein/e TesterIn Anweisungen vorgegeben, die er/sie ohne vorherige Einführung in das zu testende Produkt ausführen soll. Ein/e oder mehrere PrüferInnen beobachten den/die TesterIn dabei und machen sich Notizen über das Verhalten und dabei möglicherweise auftretende Probleme.

Auf diese Art testeten wir den digitalen Prototypen. Unsere hierfür ausgewählte Testperson ist männlich, Mitte 20, und verfügt über gute technische Kenntnisse. Die zu testende Anwendung war der Testperson vor dem Test gänzlich unbekannt. Eine Live-Aufnahme des Tests ist in dem in dieser Dokumentation angegebenen Projekt-Repository auf Github zu finden.

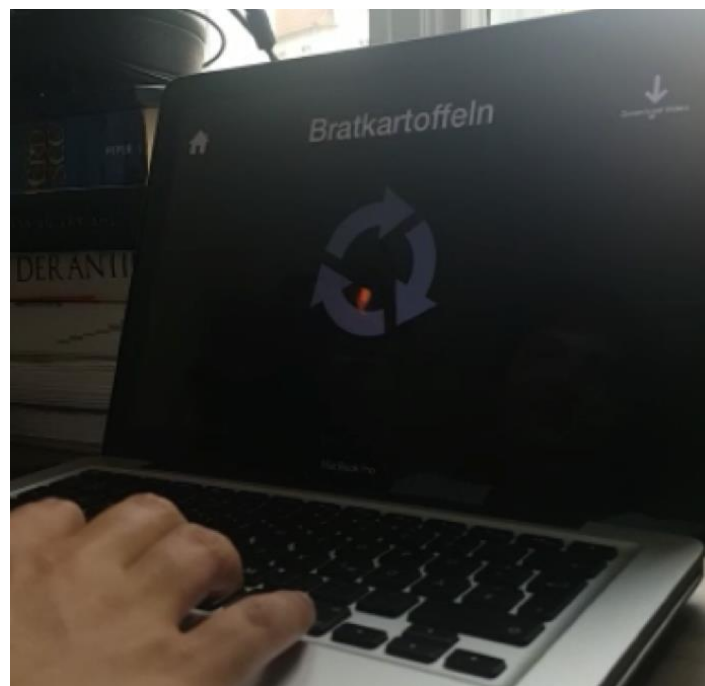
Bei diesem Testdurchlauf wurde festgestellt, dass die Landing-Page sehr unglücklich designet war. Der Text war zu lang und unaufregend, sodass der Tester damit sichtlich überfordert war und ihn nicht lesen wollte. Er stellte fest, dass der Text durch das Design entmutigend wirkte und sich negativ auf die Freude, die Anwendung zu nutzen, auswirkte.

Um dieses Problem zu lösen, wurde beschlossen, den NutzerInnen der zukünftigen Anwendung den Infotext, der auf der Landing-Page angezeigt wurde, als optionalen Menüpunkt bereitzustellen. So ist es jedem Nutzer und jeder Nutzerin selbst überlassen, ob er oder sie den Infotext angezeigt bekommt, und sie werden nicht von einem langen Text auf der Webseite begrüßt.

Der Unterschied zwischen den Anweisungen “Zurück gehen” und “auf die Startseite gehen” wurde dem Tester nicht klar, sodass es zu einer Verwechslung kam, und der Tester bei der Aufforderung, “zurück zu gehen” auf den Home-Button klickte. Diese Anweisungen wurden bewusst so gewählt, um diesen Punkt zu testen.

Da die Anwendung beziehungsweise Webseite keinen tiefen Verlinkungsbaum hat, wurde besprochen, ob die “Zurück”-Schaltfläche aus dem Prototyp herausgenommen werden sollte, da sie in fast allen Fällen den selben Effekt hat wie der Home-Button. Alternativ wurde auch ein sogenanntes Burger-Menü vorgeschlagen, also ein Menü, das auf Knopfdruck ein- und wieder ausgeblendet wird, und von dem aus man auf jeden Menüpunkt der Anwendung zugreifen kann.

Zuletzt wurde festgestellt, dass Schaltflächen, die nicht aus einem klassischen Viereck bestehen, zu klein für den Tester waren. Aus Gewohnheit klickte er nicht genau auf das Symbol, sondern lediglich in die Nähe, was zu Problemen in der Anwendung führte. Bei diesem Problem handelt es sich um einen Implementierungsfehler, der schnell behoben



werden konnte, in dem die klickbare Schaltfläche um das Symbol herum vergrößert wurde.

Abbildung 2: Live-Usability-Test des digitalen Prototypen.

AttrakDiff-Onlinetool

Bei AttrakDiff handelt es sich um ein Online-Tool, welches EntwicklerInnen und DesignerInnen dabei hilft, die Usability ihrer Anwendungen einzuschätzen. Dabei wird ein umfassender Fragebogen an die Testpersonen ausgegeben, der Fragen über die subjektive Einschätzung der Tester von Bedienbarkeit und Aussehen der Anwendung beinhaltet. Beim AttrakDiff-Evaluationsverfahren wird sowohl die pragmatische als auch die hedonische Qualität und Attraktivität der Anwendung untersucht [vgl. WWW6].

Der Evaluationsbogen besteht aus 28 Wortpaaren, die gegenübergestellt werden. Die TeilnehmerInnen geben mittels einer siebenstufigen Skala an, welcher Begriff ihr Interaktionserlebnis besser beschreibt.

Den AttrakDiff-Fragebogen haben wir acht Testpersonen aus unserem Umfeld zukommen lassen, welche unseren digitalen Prototyp beurteilt haben.

Unter den TeilnehmerInnen befanden sich 5 Personen zwischen 20 und 40, eine Person zwischen 40 und 60, sowie zwei Personen über 60. Vier der TeilnehmerInnen waren männlich und vier weiblich. Zu den beruflichen Tätigkeiten haben leider die wenigsten eine Angabe gemacht, aber da uns die TeilnehmerInnen bekannt waren, kann gesagt werden, dass die Hälfte der TeilnehmerInnen über keine nennenswerten technischen Vorkenntnisse verfügte.

Bei der Auswertung der Ergebnisse fiel auf, dass ältere TeilnehmerInnen dazu tendierten, andere und extremere Angaben zu machen, als jüngere. So gaben die TeilnehmerInnen aus den höheren Altersgruppen an, die Anwendung eher "fesselnd" als "lahm" empfunden zu haben, und deutlich mehr "originell" als "konventionell", während jüngere TeilnehmerInnen in diesen Punkten widersprachen. Mit beiden Extremen ergab sich folgendes Bild.

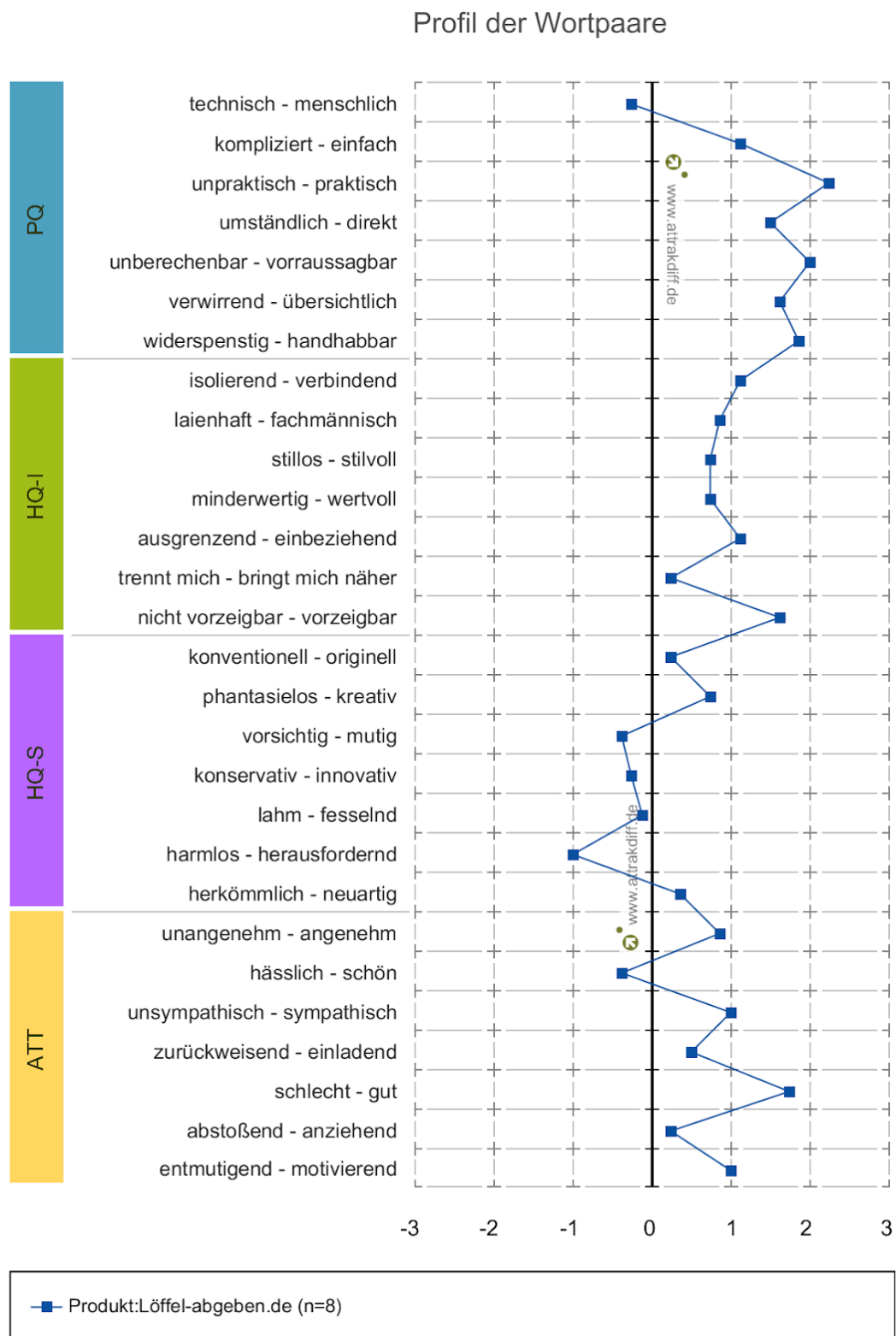


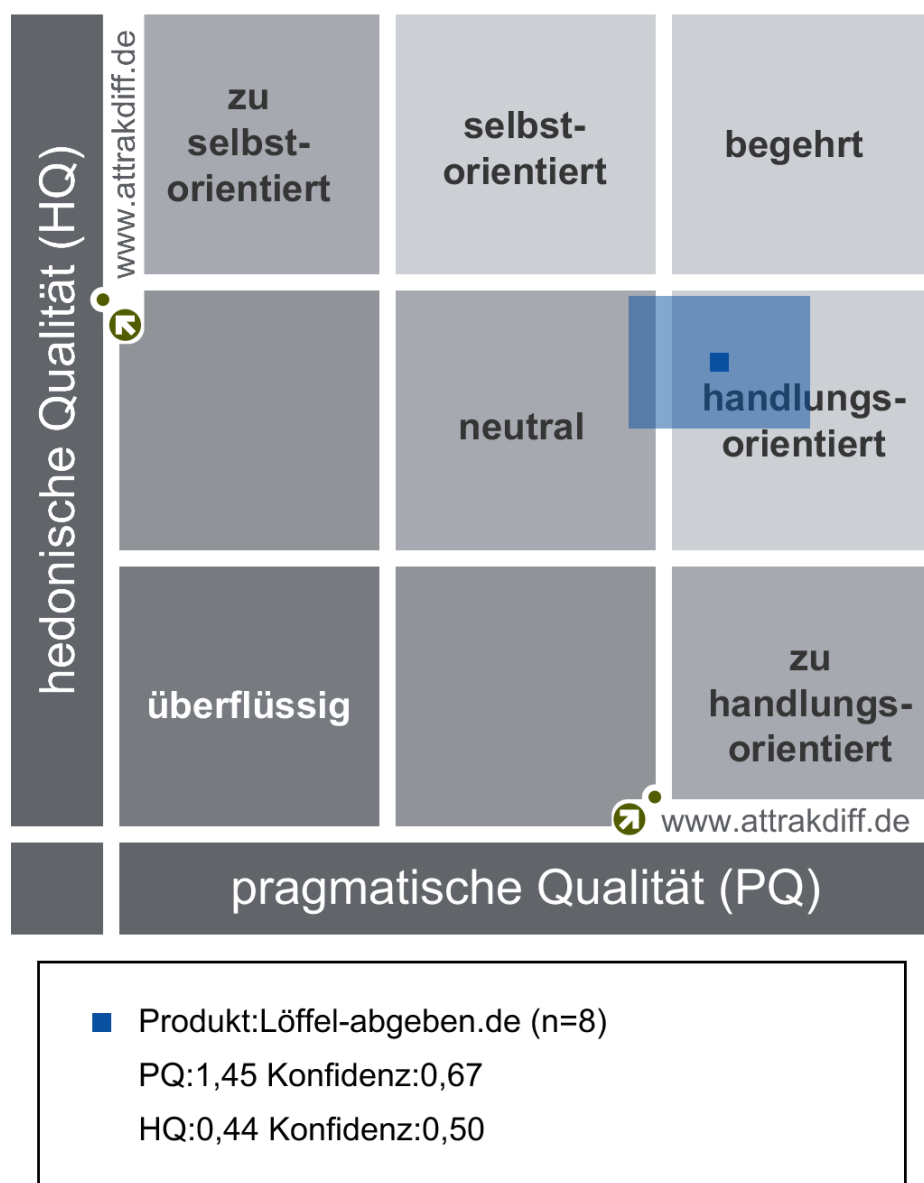
Abbildung 3: Profil der Wortpaare.

Das Online-Tool bietet außerdem die Möglichkeit, die Auswertungsergebnisse in einer Matrix anzeigen zu lassen, die auf der X-Achse die pragmatische Qualität anzeigt, also für wie handlungs- und sachbezogen die Anwendung befunden wird, und auf der Y-Achse die hedonische Qualität der Anwendung, also wie angenehm sie von NutzerInnen empfunden

wird. Der bewertete Prototyp liegt im Bereich “handlungsorientiert”, die von den TesterInnen wahrgenommene hedonische Qualität liegt im Mittelfeld, während die pragmatische Qualität etwas ausgeprägter ist.

Das kleine dunkelblaue Rechteck repräsentiert die durchschnittlichen Auswertungsergebnisse aller UmfrageteilnehmerInnen, das etwas hellere Rechteck, welches das kleinere umgibt, ist das sogenannte Konfidenz-Rechteck. Je größer dieses ausfällt, desto uneiniger waren sich die TeilnehmerInnen bei den Testfragen. Auf diese

Portfolio-Darstellung



Diskrepanz wurde im Vorherigen bereits eingegangen.

Abbildung 4: Portfolio-Darstellung der Auswertung.

Die Auswertung der AttrakDiff-Fragebögen hat ergeben, dass die Anwendung bzw. der digitale Prototyp zwar nicht als ästhetisch herausragend und die Interaktion als eher unoriginell beurteilt wurde, jedoch von den TesterInnen als unkompliziert und direkt empfunden wurde. Letztere Aspekte entsprechen unserer Intention, eine leicht verständliche und praktische Anwendung zu entwickeln, die von allen Altersstufen und ohne nennenswerte technische Vorkenntnisse verwendet werden kann. Bevor das Produkt einer breiten Öffentlichkeit zugänglich gemacht wird, muss allerdings an der Grafik gefeilt werden. Weitere Versionen sollten wie zuvor von mehreren Benutzergruppen bewertet werden, um ein optimales Ergebnis zu erzielen.

6 Zusammenfassung und Ausblick

Die im Projektrahmen erstellte Anwendung unter dem Namen *löffel-abgeben.de* liest Rezepte im txt-Format aus, erkennt Schlagwörter als Kochanweisungen, durchsucht eine von dem Projektteam und einer fiktiven Koch-Community gepflegte Datenbank nach Einträgen zu diesen Anweisungen, und generiert daraus ein Kochvideo. Je mehr Videos in die Datenbank eingepflegt wurden, desto vollständiger das Video; Anweisungen ohne hinterlegten Videoclip werden als Textsegment im Video ausgegeben. Das generierte Video kann heruntergeladen und lokal auf dem eigenen Rechner abgespielt werden. Der Weg zu diesem Endprodukt war definitiv nicht leicht. Viele Hürden wurden überwunden, letztendlich litt das Projekt dennoch in erster Linie unter dem Zeitdruck und der durch die aktuelle Pandemie-Situation stark erschwerten Kommunikation. Das finale Konzept konnte in der wenigen verbliebenen Zeit nicht vollständig den Prototypen entsprechend umgesetzt werden, jedoch wurden alle Kernfeatures implementiert und die Anwendung läuft stabil. Verschiedene Evaluationen haben ergeben, dass an dem Produkt besonders in Hinsicht auf eine ansprechende Nutzungsoberfläche nachgearbeitet werden muss, aber durch Übersichtlichkeit und gute Verständlichkeit punktet. Eine den universitären Rahmen übersteigende Weiterarbeit an dem Projekt ist geplant, eine Webseite wird momentan aufgesetzt.

7 Quellenangaben

Online-Quellen:

WWW1: Ben Fry, Casey Reas: *Processing loadStrings()*, unter

https://processing.org/reference/loadStrings_.html, zuletzt aufgerufen am 23.07.20

WWW2: Ben Fry, Casey Reas: *Processing Data*, unter

<https://processing.org/tutorials/data/>, zuletzt aufgerufen am 23.07.20

WWW3: *How to store videos in a PostgreSQL database?*, unter

<https://stackoverflow.com/questions/8970636/how-to-store-videos-in-a-postgresql-database>, zuletzt aufgerufen am 23.07.20

WWW4: The PostgreSQL Global Development Group: *PostgreSQL 7.4.30 Documentation 31.7. Storing Binary Data*, unter <https://www.postgresql.org/docs/7.4/jdbc-binary-data.html>, zuletzt aufgerufen am 23.07.20

WWW5: Ben Fry, Casey Reas: *Processing Video*, unter

<https://processing.org/reference/libraries/video/index.html>, zuletzt aufgerufen am 23.07.20

WWW6: User Interface Design GmbH (UID): *Nutzen/Einsatzgebiete*, unter

<http://www.attrakdiff.de/#nutzen>, zuletzt aufgerufen am 23.07.20

Abbildungen:

Abbildung 1: Auszug aus einem Papierprototypen einer früheren Entwicklungsphase mit 2 geplanten Features. Bearbeiteter Screenshot aus eigener Aufnahme

Abbildung 2: Live-Usability-Test des digitalen Prototypen. Screenshot aus eigener Aufnahme

Abbildung 3: Profil der Wortpaare. Erstellt mittels Online-Tool unter <http://www.attrakdiff.de/>

Abbildung 4: Portfolio-Darstellung der Auswertungsergebnisse. Erstellt mittels Online-Tool unter <http://www.attrakdiff.de/>