

Julius-Maximilian-Universität Würzburg
Neuphilosophisches Institut – Digital Humanities

Projektarbeit

Stephan Moser, Dr.

SoSe 18

**Projektarbeit –
Digitalisierung zweisprachiger Rezepte**

Gundlach Paulina

Theresienstraße 15

97070 Würzburg

paulina.gundlach@stud-mail.uni-wuerzburg.de

1974193

Anglistik/Amerikanistik und Digital Humanities 85-BA

Semester 8

Inhaltsverzeichnis

1. Einleitung	1
2. Arbeitsschritte	1
2.1 Scannen	2
2.2 Transkribieren	3
2.3 Kodieren in XML	4
2.4 XSLT-Stylesheet	5
2.5 Webpage und Gestaltung	6
3. Fazit	9
4. Referenzen	10

1. Einleitung

Im Zuge meines Bachelors habe ich verschiedene Einblicke in Auszeichnungs- und Programmiersprachen bekommen. Primär werden diese Kenntnisse im Zusammenhang mit der Digitalisierung und der Erhaltung bzw. Aufbereitung von Daten oder Material, wie z.B. Handschriften oder Textsammlungen gelehrt. Die Projektarbeit bietet sich an, um die erworbenen Grundkenntnisse in die Praxis umzusetzen. Ich entschloss mich, meine Grundkenntnisse in *xml* und *html* zu stärken, indem ich das zweisprachige Kochbuch meiner Großmutter digitalisiere und auf eine Webpage stelle. Zweisprachig ist die Rezeptsammlung deshalb, weil meine Großmutter im zweiten Weltkrieg aus ihrer schlesischen Heimat nach Kolumbien floh und dort bis zu ihrem Tod blieb. Ein weiterer Hintergedanke zu dem Projekt war, dass ich die Rezepte gerne sowohl für meine spanischsprachige, als auch für die deutschsprachige Verwandtschaft zugänglich machen wollte. Ich suchte mir also dreizehn verschiedene Kuchenrezepte heraus, welche nun auf der fertiggestellten Webpage zu sehen sind. Die einzelnen Schritte der Digitalisierung und der Umsetzung werden in den folgenden Absätzen beschrieben, wobei ebenfalls auf Probleme, die während der Erarbeitung auftraten, und auf deren Lösungen eingegangen wird.

2. Arbeitsschritte

Die dreizehn ausgewählten Kuchenrezepte habe ich aus der Rezeptsammlung meiner Großmutter, die ich nach ihrem Tod erhielt. Die meisten Rezepte wurden handschriftlich verfasst, jedoch habe ich auch eine Auswahl an Rezepten ausgesucht, die mit der Schreibmaschine geschrieben sind. Die einzelnen Arbeitsschritte werden in den folgenden Unterpunkten mit Beispielen und Auszügen erläutert.

2.1 Scannen

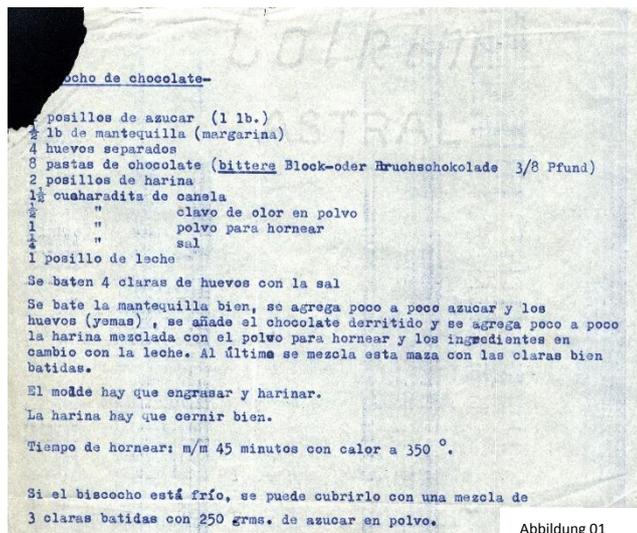


Abbildung 01

Das ausgewählte Material scannte ich an der Scan-Station der Universitätsbibliothek als tif-Datei mit einer Auflösung von 300dpi ein. Dies ist vor allem bei großen Datensätzen vorteilhaft, da es qualitativ gute Digitalisierung mit geringem Speicherverbrauch werden. Die Schreibmaschinen geschriebenen Texte ließ ich beim

Scannen direkt über OCR (Texterkennung) laufen. Die zweisprachig vermischten Rezepte ließ ich jeweils einmal mit Deutscher und einmal mit Spanischer Texterkennung über OCR laufen. Die Fehlerquote war dann geringer, wenn die Sprache der Texterkennung mit der im Text am meisten vorkommenden Sprache übereinstimmt. Typischer Fehler sind beispielsweise die Verwechslung von u zu /i, ein zu un, oder auch ß/z. Da ich nicht nur die überarbeiteten Rezepte, sondern auch die Digitalisate der originalen Rezepte auf die Webpage stellen wollte, habe ich die tif-Dateien im Bildbearbeitungsprogramm (z.B.: Photoshop) in jpg-Dateien transformiert und abgespeichert.

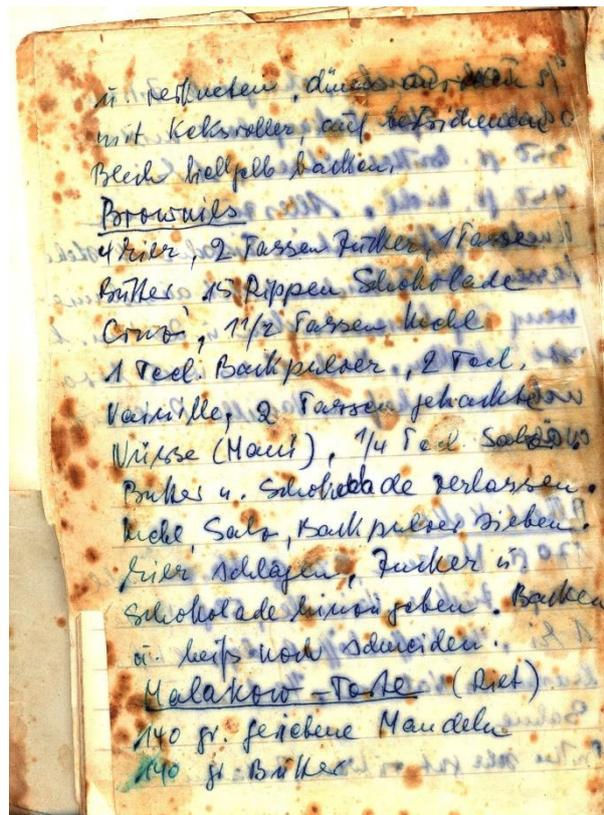


Abbildung 02

2.2 Transkribieren

Die Transkription der einzelnen Rezepte im Texteditor nahm einige Zeit in Anspruch, denn war mir zu Beginn noch nicht ganz klar, wie ich diese gerne am Ende darstellen möchte. Eine Option war, die Rezepte so nah am Original zu lassen. Dazu gehören also Rechtschreibfehler, falsche Bezeichnungen, das Vorkommen beider Sprachen in einem Rezept, oder unvollständige bzw. das Fehlen einer Zutatenliste. Die zweite Option war die Verbesserung der Texte und eine komplette Überarbeitung in ein einheitliches Schema (Zutatenliste, Vorgangsbeschreibung, nur Deutsch/Spanisch im Text). Ich entschloss mich, die Transkriptionen erst einmal recht getreu dem Original zu transkribieren und nur Kleinigkeiten auszubessern, wie z.B. offensichtliche Rechtschreibfehler (Biscocho anstatt Bizcocho). Auch fügte ich ein paar Kommentare bzw. Informationen hinzu (siehe Abbildung 03). Anschließend übersetzte ich die Rezepte jeweils ins Deutsche bzw. Spanische.

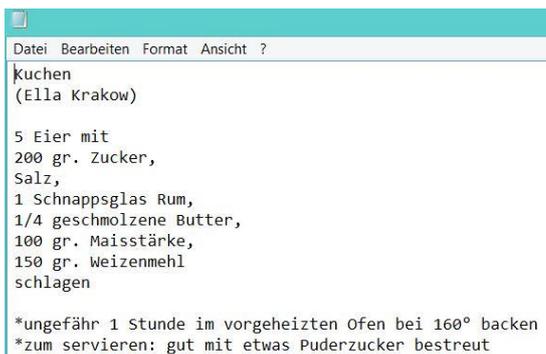


Abbildung 03

In folgender Abbildung 03 kann man außerdem sehen, dass es keine eigene Zutatenliste bzw. Vorgangsbeschreibung gibt. Hier wurde beides in einem kombiniert, was ich für die Transkription übernommen habe. Auffällig waren in den Kuchenrezepten meiner Großmutter vor allem die Vermischung beider Sprachen (siehe Abbildung 04), oder die Verwendung bestimmter Bezeichnungen wie z.B. „Champagners“ bzw. „Biskotten“ oder „Eisschrank“. Außerdem verwendete sie als Mengeneinheit oftmals Pfund und/oder Tassen (siehe Abbildung 05).

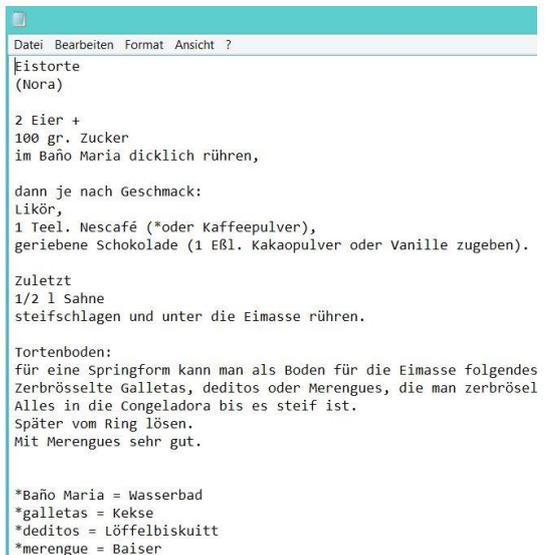


Abbildung 04

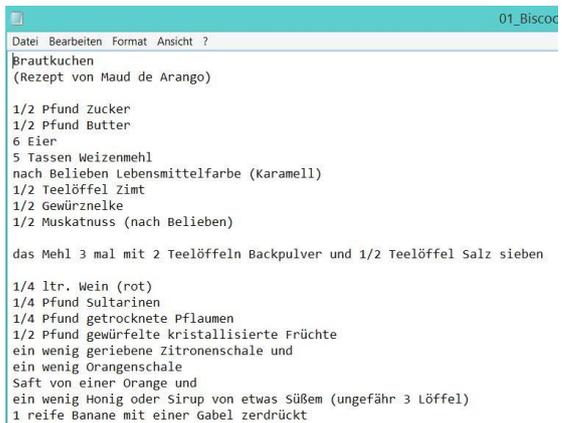


Abbildung 05

2.3 Kodieren in XML

Nachdem ich alle ausgewählten Rezepte fertig transkribiert und übersetzt habe, überlegte ich mir, wie ich die Transkriptionen kodieren wollte, bzw. was relevant für die Umsetzung meines Projekts sei. Ich entschied mich die Rezepte in ein einheitliches Schema zu bringen, das heißt es sollte für jedes Rezept eine vollständige Zutatenliste und eine Vorgangsbeschreibung geben. Ziel war nämlich, die Rezepte benutzerfreundlich darzustellen. Außerdem hatte ich die Idee, die Zutaten der Zutatenliste mit einer eindeutigen ID zu vermerken, und jede Zutat mit Tochter-Tags

```
<zutatenliste id="r04d.li01">
  <zutat id="r04d.z01"><menge>5 </menge>ganze <substanz>Eier </substanz></zutat>
  <zutat id="r04d.z02"><menge>200 </menge><einheit>gr. </einheit><substanz>Zucker</substanz></zutat>
  <zutat id="r04d.z03"><substanz>Salz</substanz></zutat>
  <zutat id="r04d.z04"><menge>1 </menge><einheit>Schnappsglas </einheit><substanz>Rum</substanz></zutat>
  <zutat id="r04d.z05"><menge>1/4 </menge>geschmolzene <substanz>Butter</substanz></zutat>
  <zutat id="r04d.z06"><menge>100 </menge><einheit>gr. </einheit><substanz>Maisstärke</substanz></zutat>
  <zutat id="r04d.z07"><menge>150 </menge><einheit>gr. </einheit><substanz>Weizenmehl</substanz></zutat>
</zutatenliste>
```

Abbildung 06

zu versehen (<menge>, <einheit> und <substanz>, siehe Abbildung 06). Zusätzlich verknüpfte ich die Zutaten aus der Vorgangsbeschreibung mit denen aus der Zutatenliste (idref="#"). So konnte ich mir die Option offen zu lassen, später (z.B. in meine Bachelorarbeit) mit *hover* (in HTML) ein Feld, das die Menge und die Einheit der Zutat angibt, anzeigen zu lassen, wenn man auf der Webpage mit der Maus über

```
<vorgangsbeschreibung id="r10d.vg01">
  Eismasse:
  <zutat idref="#r10d.z01">Eier</zutat> + <zutat idref="#r10d.z02">Zucker</zutat> im <kommentar idre
  dann je nach Geschmack:
  <zutat idref="#r10d.z03">Likör</zutat>, <zutat id="r10d.z04">Nescafé <kommentar idref="#r10d.k01">
  <zutat idref="#r10d.z05">geriebene Schokolade</zutat> (<zutat idref="#r10d.z06">Kakaopulver oder V
  Zuletzt <zutat idref="#r10d.z07">Sahne</zutat> steifschlagen und unter die Eismasse rühren.
```

Abbildung 07

die Zutat in der Vorgangsbeschreibung geht (siehe Abbildung 07). Das ist gerade bei langen Rezepten vorteilhaft, denn man erspart sich das hochscrollen zur Zutatenliste während dem Backen. Diese Überlegung habe ich ebenfalls auf die Kommentare (<kommentar>), wie auch auf die zubereitung (<zubereitung>) übertragen (s. Abb. 07). In Abbildung 08 kann man außerdem sehen, dass ich den einzelnen Rezepten eine eindeutige ID nach folgendem Schema vergeben habe: *r* für Rezept, Nummerierung des Rezepts (01-13) und *d* für Deutsch bzw. *e* für Spanisch. Wichtig für das Projekt selbst sind jedoch nur bestimmte Tags, welche im nächsten Absatz genannt werden.

```
<kochbuch>
  <rezepte>
    <kuchenrezept>
      <rezept id="r10d">
        <titel>Eistorte</titel>
        <untertitel>(Nora)</untertitel>
        <zutatenliste id="r10d.li01">
```

Abbildung 08

2.4 XSLT-Stylesheet

```
<xsl:template match="/">
  <html>
    <head>
    </head>

    <body>
      <xsl:apply-templates/>
    </body>
  </html>
</xsl:template>

<xsl:template match="rezept">
  <div>
    <xsl:apply-templates/>
  </div>
</xsl:template>

<xsl:template match="titel">
  <h3>
    <xsl:apply-templates/>
  </h3>
</xsl:template>

<xsl:template match="untertitel">
  <h5 style="margin-top:0.5em;">
    <xsl:apply-templates/>
  </h5>
</xsl:template>
```

Abbildung 09

liefern, was gebraucht und erwartet wird.

Ich änderte die Kodierungen so um, dass jedes Rezept bereits im XML-Format eine vollständige Zutatenliste enthielt. Das Stylesheet änderte und überarbeitete ich auch ein wenig parallel zur Erstellung der Webpage um. In Abbildung 09 wird gesagt, dass er eine HTML-Datei mit `<head>` und `<body>` erstellen soll. Im `<body>` sollen folgende Angaben aus XML vorliegen: aus der Rubrik `<rezept>`, soll

der Text, der als `<titel>` bezeichnet ist in `<h3>` dargestellt werden. Der Text, der als `<untertitel>` kodiert ist, soll in einem gewissen Abstand zur oberen Unterschrift in `<h5>` dargestellt werden. Aus Abb. 10 liest man heraus, dass die `<zutaten>` der `<zutatenliste>` aus XML in einer `` (unordered list) aufgezählt (``) werden sollen. Die Zutaten in der Vorgangsbeschreibung sollen fett und farbig (teal) markiert werden. Außerdem soll die gesamte Vorgangsbeschreibung einen gewissen Abstand zur Zutatenliste haben. Sobald das Stylesheet festgelegt ist und die Kodierungen valide und vollständig sind, kann man die XML-Dateien über das Stylesheet laufen lassen

Um die XML-kodierten Transkriptionen in die Webpage einzuschleusen, braucht es ein XSLT-Stylesheet, in dem definiert wird, welche Daten aus der XML-Datei in eine html-Datei gepackt werden soll. Da ich selbst nie wirklich mit XSLT gearbeitet habe, zeigte mir Herr Dr. Moser wie es funktioniert. Als wir besprachen, was genau wichtig und relevant für die Webpage ist, wurde mir bewusst, wie die Kodierungen genau auszusehen haben. Also bearbeitete ich daraufhin die Kodierungen noch einmal so, dass sie in Kombination mit dem XSLT-Stylesheet das

```
<xsl:template match="zutatenliste">
  <ul style="margin-top:1.5em;">
    <xsl:apply-templates/>
  </ul>
</xsl:template>

<xsl:template match="zutatenliste//zutat">
  <li>
    <xsl:apply-templates/>
  </li>
</xsl:template>

<xsl:template match="vorgangsbeschreibung//zutat">
  <b style="color:teal;">
    <xsl:apply-templates/>
  </b>
</xsl:template>

<xsl:template match="vorgangsbeschreibung">
  <div style="margin-top:1.5em;">
    <xsl:apply-templates/>
  </div>
</xsl:template>
```

`</xsl:transform>`

Abbildung 10

und man erhält die zugehörigen HTML-Codes (s. Abb. 11). Öffnet man diese im Browser, sieht es wie auf Abb. 12 aus.

```

<body>
  <div>
    <h2>Kuchen</h2>
    <h4 style="margin-top:0.5em;">(Ella Krakow)</h4>
    <ul style="margin-top:1.5em;">
      <li>5 ganze Eier </li>
      <li>200 gr. Zucker</li>
      <li>Salz</li>
      <li>1 Schnapsglas Rum</li>
      <li>1/4 geschmolzene Butter</li>
      <li>100 gr. Maisstärke</li>
      <li>150 gr. Weizenmehl</li>
    </ul>
    <div style="margin-top:1.5em;">
      Alle Zutaten zu einem T
      *ungefähr 1
      *zum servieren: gut mit
    </div>
  </div>
</body>

```

Abbildung 11

Kuchen

(Ella Krakow)

- 5 ganze Eier
- 200 gr. Zucker
- Salz
- 1 Schnapsglas Rum
- 1/4 geschmolzene Butter
- 100 gr. Maisstärke
- 150 gr. Weizenmehl

Alle Zutaten zu einem Teig verrühren. *ungefähr 1

Abbildung 12

2.5 Webpage und Gestaltung

Die Webpage und ihre Gestaltung sind parallel zur Kodierung und der Überführung der XML-Dateien in HTML entstanden. Ich entschied mich mit Bootstrap zu arbeiten, da Bootstrap ein schönes Layout anbieten und ich so keine eigenen CSS-Dateien anlegen musste. Anfangs war ich mir noch unschlüssig, ob ich die Rezepte auf einzelne Seiten, oder alle auf einer Seite verpacken wollte. Außerdem war am Anfang auch noch unklar, ob die Deutschen und Spanischen Rezepte in eine Tabelle gesetzt werden sollten, oder in Spalten. Ich probierte verschieden

```

<thead>
  <tr>
    <th>Brownies</th> <th>Brownies</th>
  </tr>
</thead>
<tbody>
  <tr>
    <td>
      <ul>
        <li>4 Eier</li>
        <li>2 Tassen Zucker</li>
        <li>1 Tasse Butter</li>
        <li>15 Rippen Schokolade</li>
        <li>1 1/2 Tassen Mehl</li>
        <li>1 Teelöffel Backpulver</li>
        <li>2 Teelöffel Vanillenessenz</li>
        <li>2 Tassen gehackte Nüsse (Maní) oder Mandeln</li>
        <li>1/4 Teelöffel Salz</li>
      </ul>
    </td>
    <td>
      <ul>
        <li>4 huevos</li>
        <li>2 tazas de azúcar</li>
        <li>1 taza de mantequilla</li>
      </ul>
    </td>
  </tr>
</tbody>

```

Abbildung 13

Möglichkeiten aus und kam recht schnell zu dem Entschluss, die Rezepte nicht in Tabellen einzufügen, denn so hätte ich die einzelnen Rezepte aufteilen und manuell ohne Verwendung meines angefertigten Stylesheets in den HTML-Code einsetzen müssen. (Das liegt in erster Linie

daran, dass man für eine Tabelle erst <th>, <tr> und dann <td> definiert, s. Abb. 13).

Die Rezepte wollte ich zusätzlich in Container (<div>) einfügen, also richtete ich mich demzufolge nach dem Stylesheet und fügte die transformierten XML-Codes (Rezepte) in Spalten (<div class="col-sm-4">: um 3 Spalten hinein zu packen, links für ein Foto und die Verlinkung zu dem zugehörigen Original-Scan, in der Mitte das Deutsche, und rechts das Spanisch Rezept). Bei Rezepten ohne Foto reichte links col-sm-1 und für die Rezepte jeweils col-sm-5. Wie bereits erwähnt, war es mir wichtig für jedes Rezept einen Link zum Originalen Scan hinzuzufügen. Da ich die Scans jedoch alle auf eine Seite setzen wollte, überlegt ich mir, wie ich diese mit den Rezepten so verlinken konnte, sodass man direkt zum richtigen Scan auf der Seite gelangt. Das löste ich, indem ich die Scans ebenfalls in eindeutig definierte Container („id=r001“ bzw. „id=s001“, *r* für Rezept und *s* für Scan) einschleuste und so den jeweils kompletten Container mit <div href="#s001"> verknüpfte. Das gleiche tat ich verkehrt herum auch, sodass man von den Original Scans direkt zu dem dazu gehörigen Rezept (href="#r001“) gelangt (s. Abb. 14 und 15).

```

<div class="container" id="r001">
  <div class="panel panel-default">
    <div class="panel-heading">
      Brautkuchen
    </div>
    <div class="col-sm-1" style="margin-top:1.0em;">
      <a href="scans_orig.html#s001"><b style="color:teal;">Scans</b></a>
    </div>
  </div>
</div>

```

Abbildung 14

```

<div class="container">
  <div class="panel panel-default" id="s001">
    <div class="panel-heading">
      <a href="kuchenrezepte_alle.html#r001" style="colour: navy;"><b style="color:navy;">Biscocho de Novia</b></a>
    </div>
  </div>
</div>

```

Abbildung 15

Als weitere kleine Hürde stellte sich die Navigationsleiste dar, denn diese wollte ich mit einem Drop-Down Menü versehen, sodass man auf einen Blick eine Übersicht über alle Rezepte bekommt und direkt zum ausgewählten Rezept bzw. Scan gelangt. Das Drop-Down-Menü beinhaltet ebenfalls href-Verknüpfungen zu den jeweiligen Rezepten bzw. Scans. Außerdem gibt es die Option, auf die gesamte Kuchenrezeptseite bzw. Scans-seite zu gelangen, welche an oberster Stelle des Drop-Down-Menüs dick und farbig (<b style="color:teal;">) markiert sind. In Abbildung 16 ist der komplette Befehl für die Navigationsleiste mit Dropdowns zu sehen. Beim durchscrollen fiel mir auf, dass es ebenfalls nützlich wäre, die Menü-leiste immer an oberster Stelle des Bildschirms/des Browsers zur Verfügung zu haben, sodass ich die Navigationsleiste mit dem Attribut „fixed“ kodierte (<nav class="navbar navbar-

inverse navbar-fixed-top">). *Navbar-inverse* bedeutet, dass die Farbe der Navigationsleiste invertiert (anstatt weiß, schwarz) wird.

```

<nav class="navbar navbar-inverse navbar-fixed-top">
  <div class="container-fluid">
    <div id="menu" class="navbar-header">
      <a class="navbar-brand" href="#">Oma Irene's Kuchenrezepte</a>
    </div>
    <ul class="nav navbar-nav">
      <li><a href="projekt.html">Projekt</a></li>
      <li class="dropdown">
        <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false">Rezepte<span class="caret"></span></a>
        <ul class="dropdown-menu">
          <li><a href="kuchenrezepte_alle.html"><b style="color:teal;">alle Rezepte</b></a></li>
          <li><a href="kuchenrezepte_alle.html#001">Biscocho de Novia</a>
          <li><a href="kuchenrezepte_alle.html#002">Nußtorte</a></li>
          <li><a href="kuchenrezepte_alle.html#003">Sachertorte</a></li>
          <li><a href="kuchenrezepte_alle.html#004">Kuchen (Ella Krakow)</a></li>
          <li><a href="kuchenrezepte_alle.html#005">Brownies</a></li>
          <li><a href="kuchenrezepte_alle.html#006">Malakow-Torte</a></li>
          <li><a href="kuchenrezepte_alle.html#007">Pflirsich-Marillen-Gateau</a></li>
          <li><a href="kuchenrezepte_alle.html#008">Russischer Honigkuchen</a></li>
          <li><a href="kuchenrezepte_alle.html#009">Quarktorte mit Himbeeren</a></li>
          <li><a href="kuchenrezepte_alle.html#010">Eistorte (Nora)</a></li>
          <li><a href="kuchenrezepte_alle.html#011">Biscocho de chocolate</a></li>
          <li><a href="kuchenrezepte_alle.html#012">Schokoladenkuchen</a></li>
          <li><a href="kuchenrezepte_alle.html#013">Pfefferkuchen auf dem Blech</a></li>
        </ul>
      </li>
      <li class="dropdown">
        <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false">Scans<span class="caret"></span></a>
        <ul class="dropdown-menu">
          <li><a href="scans_orig.html"><b style="color:teal;">alle Scans</b></a></li>
          <li><a href="scans_orig.html#001">Biscocho de Novia</a>
          <li><a href="scans_orig.html#002">Nußtorte</a></li>
          <li><a href="scans_orig.html#003">Sachertorte</a></li>
          <li><a href="scans_orig.html#004">Kuchen (Ella Krakow)</a></li>
          <li><a href="scans_orig.html#005">Brownies</a></li>
          <li><a href="scans_orig.html#006">Malakow-Torte</a></li>
          <li><a href="scans_orig.html#007">Pflirsich-Marillen-Gateau</a></li>
          <li><a href="scans_orig.html#008">Russischer Honigkuchen</a></li>
          <li><a href="scans_orig.html#009">Quarktorte mit Himbeeren</a></li>
          <li><a href="scans_orig.html#010">Eistorte (Nora)</a></li>
          <li><a href="scans_orig.html#011">Biscocho de chocolate</a></li>
          <li><a href="scans_orig.html#012">Schokoladenkuchen</a></li>
          <li><a href="scans_orig.html#013">Pfefferkuchen auf dem Blech</a></li>
        </ul>
      </li>
    </ul>
  </div>
</nav>

```

Abbildung 16

Das Ganze sieht auf dem Browser folgendermaßen aus (s. Abb. 17-19):



Abbildung 17

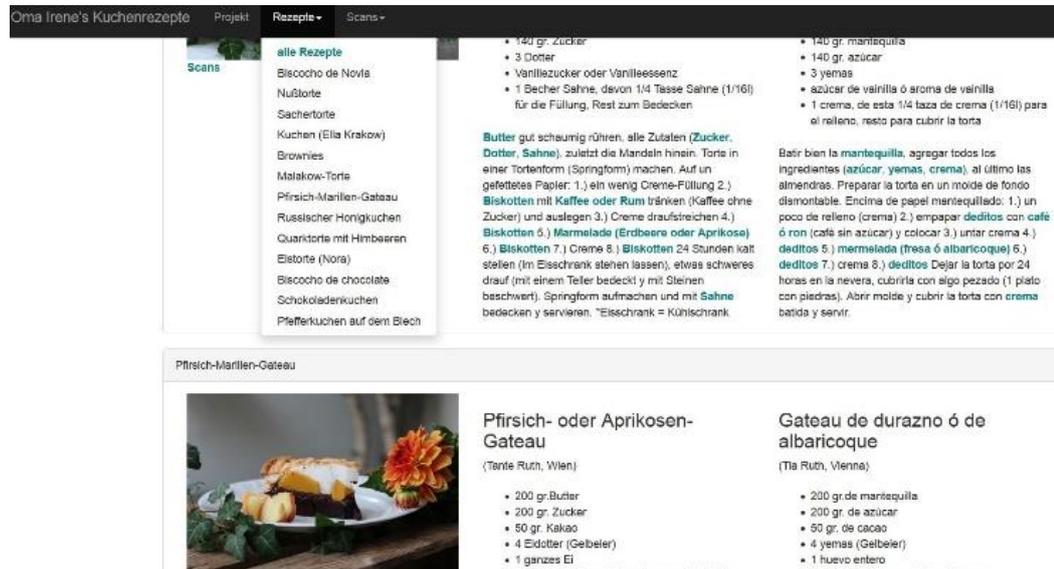


Abbildung 18

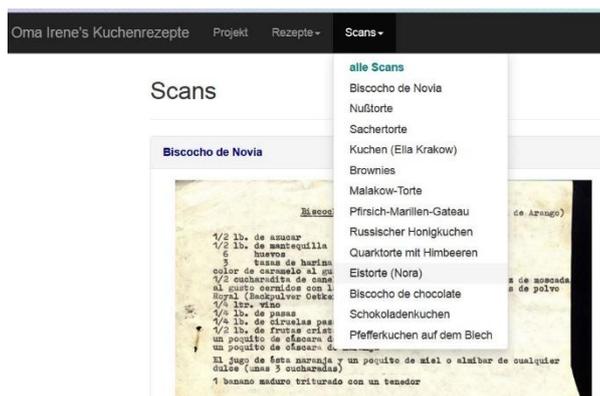


Abbildung 19

Zusätzlich zu der Seite mit den Kuchenrezepten und der Seite mit den Scans erstellte ich eine Art Startseite, auf welcher ich ein Foto meiner Großmutter und eine kurze Beschreibung zu meinem Projekt und der Idee dahinter setzte (s. Abb. 20). Das habe ich ebenfalls in einen

Container verpackt und mit der gleichen Navigationsleiste versehen.

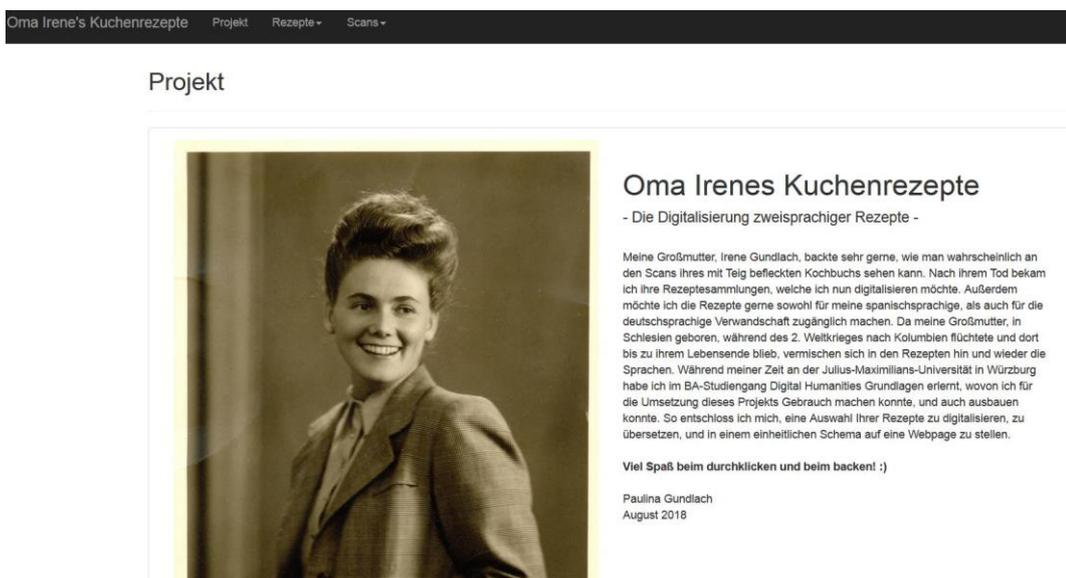


Abbildung 20

3. Fazit

Die Projektarbeit hat sehr viel Spaß gemacht, zwar habe ich mir eine sehr zeitaufwändige Idee ausgesucht, es war mir jedoch wegen dem persönlichen Bezug und Hintergedanken mehr als wert. Außerdem habe ich dadurch all das erlernte Wissen aus den vergangenen Semestern praktisch ein- und umsetzen können. Wahrscheinlich hätte ich mir viel Zeit sparen können, wenn ich von Anfang an ein konkreteres Bild oder eine konkretere Vorstellung gehabt hätte, wie das Ganze am Ende genau aussehen soll, bzw. welche Funktionen das fertige Projekt beinhalten soll. Sehr hilfreich für eine konkrete Vorstellung hat mir die Erstellung des XSLT-Stylesheets, denn erst dann wurde mir bewusst, wie genau ich vorgehen will bzw. muss, um die Ideen umzusetzen. Trotz, dass ich einige Arbeitsschritte hätte schneller erledigen können, bin ich dennoch überzeugt, dass gerade Probleme und deren Lösungsfindung zum Lernprozess dazu

gehören. Deshalb ist der Spruch „Übung macht den Meister“ sehr passend zur Beschreibung meines Studiengangs Digital Humanities. Das Projekt kann ich nun aber mit der erlernten praktischen Erfahrung für meine Bachelorarbeit ausbauen. Dazu gehören zum Beispiel die in Paragraph 2.3 beschriebene Idee: die Verknüpfung der Zutaten aus der Zutatenliste, sodass sich bei der Zutat in der Vorgangsbeschreibung ein *hover*-Feld mit genauen Angaben zur Menge und Einheit der Substanz öffnet. Außerdem habe ich mir überlegt, eine Suchleiste hinzuzufügen, um durch das Eingeben einer Zutat bzw. einer Substanz (mit Mengenangabe) alle Rezepte angezeigt zu bekommen, welche mit genau dieser Zutat bzw. Menge zu backen möglich sind. Das wäre vor allem dann vorteilhaft, wenn man beispielsweise nur noch wenige Eier oder andere Zutaten zur Verfügung hat und die Geschäfte bereits geschlossen sind. Das kommt gerade bei Studenten ab und an mal vor...

Das Projekt könnte man außerdem auch in die sprachwissenschaftliche Richtung ausbauen, indem man bei der Digitalisierung und bei der Transkription auf die Sprache eingeht. Dies ist gerade bei großen Datensätzen interessant und aussagekräftiger. Bei den Rezepten meiner Großmutter konnte man jedoch bereits bei einer kleinen Anzahl an Rezepten einige sich wiederholende Muster erkennen: dazu gehören wie bereits in Punkt 2.2 beschrieben, die Verwendung bestimmter Begriffe (gerade bei den Deutschen Rezepten werden bestimmte Spanische Begriffe wiederholt verwendet), Rechtschreibfehler (bizcocho anstatt biscocho), oder Bezeichnungen, die man heutzutage nicht mehr so verwendet (z.B. Eisschrank oder Vanillenzucker).

Ich bin gespannt wie sich das Projekt weiter entwickeln wird, bin jedoch mit dem ersten Ergebnis sehr zufrieden, denn ich habe alles soweit erreicht, was mir für diese Arbeit zum Ziel gesetzt habe: Eine Webpage mit direkten Verlinkungen zu den Rezepten bzw. zu den Scans, die farbliche Hervorhebung der Zutaten in der Vorgangsbeschreibung und eine fixierte Navigationsleiste mit Dropdown-Menüs.

4. Referenzen

Für die Erstellung der Webpage orientierte ich mich an [w3schools.com](https://www.w3schools.com) und an getbootstrap.com.