



HÖHERE TECHNISCHE BUNDESLEHRANSTALT Wien 3, Rennweg  
IT & Mechatronik

HTL Rennweg :: Rennweg 89b  
A-1030 Wien :: Tel +43 1 24215-10 :: Fax DW 18

# Diplomarbeit

## Capentory Digitalisierung der Schulinventur

ausgeführt an der  
Höheren Abteilung für Informationstechnologie/Ausbildungsschwerpunkt  
Netzwerktechnik  
der Höheren Technischen Lehranstalt Wien 3 Rennweg

im Schuljahr 2019/2020

durch

**Josip Domazet**  
**Mathias Möller**  
**Hannes Weiss**

unter der Anleitung von

DI Clemens Kussbach  
DI August Hörandl

Wien, 3. Januar 2020



# Kurzfassung

Darum geht es.

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.



# Abstract

Thats why – the translated text „Kurzfassung“ (this should be a translation).

Im englischen Abstract sollte inhaltlich das Gleiche stehen wie in der deutschen Kurzfassung. Versuchen Sie daher, die Kurzfassung präzise umzusetzen, ohne aber dabei Wort für Wort zu übersetzen. Beachten Sie bei der Übersetzung, dass gewisse Redewendungen aus dem Deutschen im Englischen kein Pendant haben oder völlig anders formuliert werden müssen und dass die Satzstellung im Englischen sich (bekanntlich) vom Deutschen stark unterscheidet. Es empfiehlt sich übrigens – auch bei höchstem Vertrauen in die persönlichen Englischkenntnisse – eine kundige Person für das „proof reading“ zu engagieren. Die richtige Übersetzung für „Diplomarbeit“ ist übrigens schlicht thesis, allenfalls „diploma thesis“ oder „Master’s thesis“, auf keinen Fall aber „diploma work“ oder gar „dissertation“[15].

Wichtig ist wegen des Abteilens ein `\begin{english}` bzw. `\selectlanguage{ngerman}`.



# Ehrenwörtliche Erklärung

Ich erkläre an Eides statt, dass ich die individuelle Themenstellung selbstständig und ohne fremde Hilfe verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche erkenntlich gemacht habe.

Wien, am 3. Januar 2020

---

Josip Domazet

---

Mathias Möller

---

Hannes Weiss





# Inhaltsverzeichnis

<b>Tabellenverzeichnis</b>	<b>xi</b>
<b>Abbildungsverzeichnis</b>	<b>xiii</b>
<b>1 Ziele</b>	<b>1</b>
<b>2 Formatierung</b>	<b>3</b>
2.1 Vorlagen . . . . .	3
2.1.1 Abweichungen . . . . .	3
2.1.2 Formatvorlagen . . . . .	3
2.1.3 Schriften und Absätze . . . . .	3
2.1.4 Bilder . . . . .	5
2.1.5 Tabellen . . . . .	5
2.1.6 Formel . . . . .	5
2.1.7 Sourcecode . . . . .	6
2.1.8 Fachbegriffe . . . . .	7
2.1.9 Zitieren . . . . .	7
2.2 Inhalt . . . . .	9
2.2.1 Aussagen . . . . .	9
2.2.2 Bad Practice . . . . .	9
2.3 Details zu Formatierung . . . . .	11
2.3.1 Schriftarten . . . . .	11
2.4 Beispiele . . . . .	11
2.4.1 Zitieren mit Latex . . . . .	11
2.4.2 Direkte Formatierungen – sollte man vermeiden . . . . .	12
2.5 Technische Umsetzung: Server . . . . .	13
2.5.1 Django und Ralph . . . . .	14
2.5.2 Kurzfassung der Funktionsweise von Django und Rlaph . . . . .	15
<b>3 Planung</b>	<b>21</b>
<b>A Anhang 1</b>	<b>23</b>
<b>Literaturverzeichnis</b>	<b>27</b>



# Tabellenverzeichnis

2.1 So eine tolle Tabelle . . . . .	5
-------------------------------------	---

kann  
entfal-  
len falls  
(fast) leer



# Abbildungsverzeichnis

2.1 Ein Bild . . . . .	5
------------------------	---



# 1 Ziele

Das erste Kapitel stellt die Ziele der DA (inkl. individuelle Ziele aller Mitarbeiter) dar.

viel Text  
schreiben

Mögliche Gliederung (nach [32])

- Einleitung
- Zielsetzung und Aufgabenstellung des Gesamtprojekts
- individuelle Zielsetzung und Aufgabenstellung mit Terminplan der einzelnen Teammitglieder
- Grundlagen und Methoden (Ist-Situation, Lösungsansätze, konkrete Vorgehensweise)
- Bearbeitung der Aufgabenstellung (technische Beschreibungen, Berechnungen)
- Ergebnisse (Ergebnisdarstellung, kritische Gegenüberstellung mit der Zielsetzung und der gewählten Vorgehensweise)





## 2 Formatierung

### 2.1 Vorlagen

In diesen Kapitel gibt es einige Muster für Dinge die oft vorkommen.

#### 2.1.1 Abweichungen

Die in diesem Dokument getroffenen Einstellungen bzw. das resultierende PDF ist die Referenz für die Diplomarbeiten der Abteilung IT.

#### 2.1.2 Formatvorlagen

Alle Formatierungen sollten mit Formatvorlagen vorgenommen werden. Spätestens bei der Konvertierung in ein Ebook rächen sich diese „Sünden“: Ebooks sind HTML Dokumente mit einer Formatierung mittels CSS.

Auch bei der Umwandlung in interaktive PDFs ist eine konsequente Formatierung wichtig.

#### 2.1.3 Schriften und Absätze

Hier findet man eine Beschreibung des Layouts – Details folgen weiter unten.

**Schrift:** dieses  $\text{\LaTeX}$ -Dokument verwendet die Standardschriften. Die Schriftgröße soll 12 pt betragen.

**Absatz:** entweder verwendet man wie in  $\text{\LaTeX}$  einen etwas größeren Seitenrand oder einen größeren Zeilenabstand. Beides sorgt für bessere Lesbarkeit. Zwischen den Abätzen ist ein Abstand. Alternative: die erste Zeile eines Absatzes wird etwas eingerückt (nicht die erste Zeile nach einer Überschrift, nach einem Bild etc.)

und bzw. oder es gibt einen Abstand zwischen den Absätzen. Am Ende und Anfang einer Seite sollten mindestens zwei Zeilen eines Absatzes sein (keine Schusterjungen<sup>1</sup> und Hurenkinder<sup>2</sup>).

**Blocksatz:** Alle Texte werden im Blocksatz gesetzt. Die Silbentrennung ist dann obligatorisch.

**Kapitelüberschriften:** Überschriften erster Ordnung sollten auf rechten Seiten beginnen. Über jeder Überschrift sollte ein Abstand sein. Alle Überschriften müssen mit dem nächsten Absatz „zusammengehalten“ werden – keine einsamen Überschriften am Ende einer Seite.

**Inhaltsverzeichnis:** das Inhaltsverzeichnis sollte möglichst kompakt sein. Als Gliederung dienen fette Hauptüberschriften und etwas Abstand über den Zeilen.

**Seitenformat:** der Ausdruck erfolgt zweiseitig, ein entsprechender Bundsteg ist zu berücksichtigen<sup>3</sup>. Nach Rücksprache mit dem Betreuer kann auch eine einseitige Variante gewählt werden. Bei Bedarf könne auch einzelne Seiten im Querformat gesetzt werden.

**Kopfzeile:** die Kopfzeile sollte dieser Vorlage entsprechen. Falls, nach Rücksprache mit dem Betreuer, der Ausdruck nur in Schwarz-weiß erfolgt, kann das Logo entfallen.

**Fußzeile:** hier ist Platz für den Autor des Kapitels und die Seitennummer. Wie bei technischen Publikationen üblich ist die Einleitung und die Verzeichnisse mit römischen Seitennummern versehen. Das eigentliche Dokument wird mit arabischen Ziffern nummeriert. Beide Nummerierungen sind unabhängig voneinander und beginnen jeweils bei 1.

**Autor:** Jedes Kapitel muss auch einem Autor haben. Das sieht man in der Fußzeile oder als Textbox in der Nähe der Überschrift. Alternativ kann es im Anhang eine Liste geben. Das ist besonders wichtig wenn es viele Beilagen, z.B. Handbücher ohne direkte Angabe des Autors, gibt.

**PDF:** Die PDF Metainformation sollten richtig sein (Autor etc.) – siehe Datei/Eigenschaften. Links auf Webseiten, Verweise innerhalb des Dokuments, das Inhaltsverzeichnis, die Fußnoten usw. sollten „klickbar“ sein.

---

<sup>1</sup> siehe <http://www.typolexikon.de/s/schusterjunge.html>

<sup>2</sup> siehe <http://www.typolexikon.de/h/hurenkind.html>

<sup>3</sup> Die Einstellung der Seitenränder ist keinesfalls beliebig. Sie sollte bewährten Regeln folgen, [...]. Die häufige Zielvorgabe „Den Platz auf dem Papier möglichst gut ausnutzen“ ist keine typografische sondern eine extrem laienhafte Regel. aus [18]

### 2.1.4 Bilder

Das Bild als Gleitobjekt ist genau hier, oder oben auf der Seite, oder unten, aber immer zentriert mit Nummer und Beschreibung – wenn es sinnvoll ist auch mit Querverweis (siehe Abbildung 2.1). Durch Gleitobjekte, d. h. Bilder oben oder unten auf der Seite statt „genau hier“, werden halbleere Seiten durch besonders große Bilder vermieden.

Wichtig: alle Bilder oder andere Medien z. B. Screenshots, Audio oder Video für EBooks und interaktive PDFs sollten mit einen entsprechenden Quellennachweis versehen sein.



Abbildung 2.1: Ein Bild

Eingefügte Screenshots sind im Ausdruck meistens unscharf. Ursache

- zu niedrige Auflösung
- als JPEG mit verlustbehafteter Komprimierung gespeichert

Abhilfe: z.B. <http://meyerweb.com/eric/thoughts/2015/10/22/firefoxs-screenshot-command>

### 2.1.5 Tabellen

In der folgenden Tabelle sieht man: es gibt immer eine Nummer und eine Beschreibung. Besonders längere Tabellen sollten eventuell als Gleitobjekt am Ende oder Anfang einer Seite positioniert werden. Geht die Tabelle über mehrere Seiten so ist die Überschrift zu wiederholen.

Überschrift	Wert	noch einer
1	abc	Hallo
2	def	Latex

Tabelle 2.1: So eine tolle Tabelle

### 2.1.6 Formel

Etwas Text als Überleitung zu einer Formel:

$$f(x) = \begin{cases} \log_8 x & x > 0 \\ 0 & x = 0 \\ \sum_{i=1}^5 \alpha_i + \sqrt{-\frac{1}{x}} & x < 0 \end{cases}$$

Wenn man sehr viele Formeln hat sollte man diese auch nummerieren. Besonders bei Verweisen ist das sehr sinnvoll.

Hinweis: auch für Werte wie 100 MiB gibt es ein eigenes Paket – siunitx.

### 2.1.7 Sourcecode

Wichtig: keine endlos langen Listings ohne Erklärung. Besser sind kurze, im Text erläuterte Ausschnitte des Codes, ohne Code-Kommentare. Man kann z. B. auch die Fehlerbehandlung entfernen und stattdessen auf den Quelltext verweisen.

Sourcecode sollte in einer Schrift mit fixer Breite sein. Der Zeilenabstand sollte möglichst gering sein. Falls man Verweise braucht sollte man die Listings auch nummerieren.

```
1 #include <stdio.h>
3 int main()
{
5     printf("Hello world\n");
}
```

Die genaue Formatierung ist freigestellt: Einstellungen wie bunt bzw. fett, Markierung von Leerzeichen und Zeilennummerierung kann an den Bedarf der Diplomarbeit angepasst werden.

Beispiel Java mit anderen Einstellungen – nur als Beispiel, in der Diplomarbeit sollte man sich an eine einheitliches Format halten. Bei längeren Listings muss man eventuell mit Umbrüchen rechnen, oder man verwendet einen Rahmen der frei angeordnet werden kann (siehe Kapitel 2.1.4, Seite 5).

Achtung  $\text{\LaTeX}$ -User: Listing kann keine Umlaute, aber unter [19] gibt es eine Lösung.

```
import java.awt.*;
import java.awt.event.*;

public class AL extends Frame
    implements WindowListener, ActionListener {
    TextField text = new TextField(20);
```

```
Button b;  
private int numClicks = 0;  
  
public static void main(String[] args) {  
    AL myWindow = new AL("My first window");  
    myWindow.setSize(350,100);  
    myWindow.setVisible(true);  
}  
}
```

Listing 2.1: Java Beispiel

Hinweise:

- Pandoc erzeugt automatisch bunte Listings
- und es gibt die Pakete listings und minted
- aber man sollte die drei Varianten nicht mischen!

## 2.1.8 Fachbegriffe

Fachbegriffe in einer Fremdsprache oder Kommandos sollten einheitlich gekennzeichnet werden. Bei Latex verwendet man dazu „logisches Markup“, bei Word oder Open/Libre-office wird all diesen Wörtern wird eine Vorlage zugewiesen, das Aussehen wird dann an einer Stelle zentral festgelegt.

Als Beispiel soll *Text to Speech* dienen. Solche Wörter sollte natürlich in ein Glossar aufgenommen werden.

Oder der Befehl **dir** für die Kommandozeile. Die Angabe von Dateinamen sollte auch einheitlich sein: entweder */etc/passwd* oder **C:\system32**.

## 2.1.9 Zitieren

Die Quellenangabe kann in Form eines Vollbelegs in der Fußnote<sup>4</sup>(bei technischen Dokumenten eher unüblich) oder als Kurzbeleg am Schluss der gesamten Arbeit aufgeführt werden. Beim Kurzbeleg sind dabei verschiedene Formen üblich. Der platzsparendste,

<sup>4</sup> aus Zitat — Wikipedia, Die freie Enzyklopädie, <http://de.wikipedia.org/w/index.php?title=Zitat>, Abgerufen 2014-09-14

aber am wenigsten aussagekräftige Zitierstil ist die fortlaufende Nummerierung aller zitierten Quellen [123] – diese wird auch in dieser Vorlage verwendet.

Alternativen – nur in begründeten Ausnahmefällen: Insbesondere in der Informatik üblich ist eine Kombination der ersten drei Buchstaben des Autorennamens und der letzten beiden Ziffern des Erscheinungsjahres (z. B. „The04“ für Theisen 2004). Wohl am weitesten verbreitet ist der vollständige Verfassername mit Erscheinungsjahr, wobei mehrere Quellen desselben Autors innerhalb eines Jahres durch fortlaufende Buchstaben kenntlich gemacht werden (z. B. „Theisen 2004c“). Weniger üblich, aber am aussagekräftigsten ist die Quellenangabe unter Hinzufügung eines Schlagwortes, das den mit der Materie vertrauten Leser zumeist bereits die zitierte Quelle erkennen lässt, z. B. in der Form „Theisen (Wissenschaftliches Arbeiten, 2004)“.

Obwohl mehrere Zitierstile bzw. Zitiertechniken zur Verfügung stehen, werden in einem Dokument üblicherweise nicht mehrere verwendet; ein ausgewählter Zitierstil wird im gesamten Dokument konsequent beibehalten. Ein gute Übersicht bietet [37].

Auch unter <http://www.diplomarbeiten-bbs.at/zitation-plagiate> gibt es gute Informationen zum Thema.

### 2.1.9.1 Quellenverzeichnis

Unter L<sup>A</sup>T<sub>E</sub>X kann das Programm BibT<sub>E</sub>X zur Erstellung von Literaturangaben verwendet werden.

- Links auf Wikipedia sollten vermieden werden.
- Jeder Link sollte mit einem Abfragedatum versehen sein.
- Das Literaturverzeichnis kommt an das Ende des Dokuments.

Viele Details dazu findet man bei [37].

### 2.1.9.2 Rechtliches zum Zitieren

Achtung: nicht gekennzeichnete Zitate (Plagiate) führen zu einer negativen Beurteilung der Diplomarbeit. Nach [36]:

§ 57 des österreichischen Urheberrechtsgesetzes[33] enthält detaillierte Vorschriften über die Quellenangabe, unter anderem: Werden Stellen oder Teile von Sprachwerken nach § 46 vervielfältigt, so sind sie in der Quellenangabe so genau zu bezeichnen, dass sie in dem benutzten Werke leicht aufgefunden werden können. In den Erläuterungen (ErlRV)

heißt es: Bei Entlehnungen aus umfangreichen Werken muss also in der Quellenangabe auch die Seite, der Abschnitt, das Kapitel oder der Akt, wo sich die entlehnte Stelle befindet, angeführt werden (Dillenz, Materialien zum österreichischen Urheberrecht, 134, zitiert nach [34], S. 621)

2002 nahm der österreichische OGH zur Frage der Quellenangabe in der Entscheidung Riven Rock Stellung: Nach § 57 Abs 4 UrhG bedarf die Unterlassung einer Quellenangabe der Rechtfertigung durch die im redlichen Verkehr geltenden Gewohnheiten und Gebräuche. Bei Auslegung dieser Bestimmung ist eine Abwägung der Interessen des Urhebers mit jenen des zur freien Werknutzung Berechtigten nach dem Verständnis loyaler, den Belangen des Urhebers mit Verständnis gegenüberstehenden, billig und gerecht denkenden Benutzern (Vinck aaO § 63 Rz 2) geboten und danach zu beurteilen, ob dem freien Werknutzer neben der Nennung des Autors/Verlags auch die Nennung des Namens des Übersetzers von in einer Rundfunksendung verlesenen Roman-Zitaten zumutbar ist.

## 2.2 Inhalt

### 2.2.1 Aussagen

Alles im Buch sollte man in folgende drei Typen einteilen:

1. Stand der Technik – Quelle notwendig
2. selbst gemacht – Verweis wie und wo
3. Meinung – als eigene Meinung bzw. Einschätzung kennzeichnen

### 2.2.2 Bad Practice

Was man vermeiden sollte – diese Dinge führen zur Mehrarbeit und verursachen zusätzlichen Stress in der hektischen Zeit knapp vor dem Abgabetermin.

#### Stil

- Extrem lange, geschachtelte Sätze und/oder endlose Textpassagen ohne Gliederung durch Absätze.

- Vielleicht bzw. sinnvollerweise lassen Sie den Text auch von einer „außenstehenden Person“ lesen.
- Aufzählungen im Text statt Listen. Wie man hier sieht dürfen bei Listen auch mehrere Sätze stehen.
- Unterstreichen ist ein Relikt aus „Schreibmaschinen-Zeiten“.
- Eine Diplomarbeit ist keine Erzählung. Natürlich kann man als „ich“ oder „wir“ auf „unsere Probleme“ eingehen, aber im Allgemeinen ist ein formaler, beschreibender und technischer Stil einzuhalten.
- Eine Diplomarbeit ist auch keine Email oder SMS: Schreiben Sie ganze Sätze ohne kryptische Abkürzungen und Smileys.
- Ein Mindestmaß an Interpunktion wird vorausgesetzt. Eventuell lassen Sie den Text durch eine kundige Person Ihres Vertrauens korrigieren.
- Es gibt viele verschiedene Striche, und alle sehen verschieden aus: Gedankenstriche, Bindestriche und Minus kommen in einer Diplomarbeit häufig vor.
- Weitere Wörter die Ihren Betreuer verzweifeln lassen – natürlich nur bei übermäßiger Verwendung
  - Welcher/Welches, Hierbei

## Technik

- (viele) händische Formatierungen statt Formatvorlagen.
- zusätzliche manuelle Seitenumbrüche oder Leerzeilen für ein „schöneres“ Layout. Es gibt bei den Absatzformatierungen tolle Möglichkeiten für Abstände vor und nach einem Absatz bzw. zum Beeinflussen des Textflusses. Mittels z. B. `\needspace{2cm}` kann man für *genügend* Platz sorgen.
- Arbeiten Sie mit dem Programm statt gegen das Programm:
  - Verweise als fixer Text. Nutzen Sie die Möglichkeiten der Textverarbeitung.
  - Dinge die „kompliziert“ einzugeben sind, sind meist falsch – richtige Lösungen sind in allen Programmen auch „leicht“ zu erreichen<sup>5</sup>.

---

<sup>5</sup> Oder Sie verwenden ein für Ihre Zwecke schlecht geeignetes Programm.



- Kontrollieren Sie beim fertigen PDF die Angaben unter Datei / Eigenschaften – dort sollten sinnvolle Dinge stehen. Bei Latex wird dazu das Paket `hyperref` verwendet.

## 2.3 Details zu Formatierung

### 2.3.1 Schriftarten

Die Word- und Libreoffice-vorlage verwenden etwas andere Schriften als das Latex Dokument.

## 2.4 Beispiele

### 2.4.1 Zitieren mit Latex

Am Beispiel der URL <http://de.wikibooks.org/wiki/LaTeX-Kompendium> und des Buches „`LaTeX`: Einführung“.

Man braucht eine `.bib` Datei mit den notwendigen Informationen:

```

@book{kopka1991latex ,
  title={LaTeX: Einfuehrung},
  author={Kopka, Helmut and Rahtz, Sebastian},
  volume={2},
  year={1991},
  publisher={Addison-Wesley}
}

@online{latexKomp ,
  author = {},
  title = {LaTeX-Kompendium – Wikibooks, Sammlung freier Lehr-,
Sach- und Fachbuecher},
  url = {http://de.wikibooks.org/wiki/LaTeX-Kompendium},
  lastchecked = {2014.09.14},
}

```

Diese Einträge werden dann im Text verwendet: `\cite{kopka1991latex}` und das Ergebnis sieht dann so [35] und so [17] aus. Gleichzeitig erscheinen diese Einträge auch im Literaturverzeichnis am Ende des Dokuments.

### 2.4.2 Direkte Formatierungen – sollte man vermeiden

Der Autor kann natürlich auch eingreifen und zum Beispiel Zeilenumbrüche erzwingen und

Leerzeilen – aber das sollte man nicht machen.

Ein Seitenumbruch kostet mich auch nicht mehr als einen müden Lacher, ist aber noch seltener wirklich sinnvoll.

Schriftgrößen ändern:

- tiny
- scriptsize
- footnotesize
- small
- normalsize
- large
- Large
- LARGE
- huge
- Huge

ein Strich unten drunter oder sogar zwei.

## 2.5 Technische Umsetzung: Server

Die Inventur- sowie Gegenstandsdaten der HTL Rennweg sollen an einem zentralen Ort verwaltet und geführt werden. Der für diesen Zweck entwickelte Server muss also folgende Anforderungen erfüllen:

- eine einfache Datenbankverwaltung und -verbindung
- das Führen einer Historie aller Zustände der Inventar-Gegenstände
- eine Grundlage für eine Web-Administrationsoberfläche
- die Möglichkeit für Datenimport und -export, etwa als *.xlsx* Datei
- eine Grundlage für die Kommunikation mit der Client-Applikation
- hohe Stabilität und Verfügbarkeit

Angesichts der Programmiersprachen, in die das Projektteam spezialisiert ist, stehen als Backend-Lösung vier *Frameworks* zur öffentlichen Verfügung, zwischen denen gewählt wurde:

- Django [21]
- Pyramid [28]
- Web2Py [29]
- Flask [26]

Alle genannten Alternativen sind *Frameworks* der Programmiersprache Python. Gewählt wurde “Django” aufgrund einer bestehenden und frei verfügbaren Inventarverwaltungsplattform “Ralph” [30], welche auf Django aufbaut und durch die vorliegende Diplomarbeit hinreichend erweitert wird.

### 2.5.1 Django und Ralph

Django ist ein in der Programmiersprache Python geschriebenes Webserver-*Framework*. Ralph ist eine auf dem Django-*Framework* basierende *DCIM* und *CMDB* Softwarelösung. Haupteinsatzgebiet dieser Software sind vor allem Rechenzentren mit hoher Komplexität, die externe Verwaltungsplattformen benötigen. Zusätzlich können aber auch herkömmliche Inventardaten von IT-spezifischen Gegenständen in die Ralph-Plattform aufgenommen werden.

Ralph wurde von der polnischen Softwarefirma “Allegro” entwickelt und ist unter der Apache 2.0 Lizenz öffentlich verfügbar. Dies ermöglicht auch Veränderungen und Erweiterungen. Zu Demonstrationszwecken bietet Allegro eine öffentlich nutzbare Demo-Version [4] von Ralph an.

Die vorliegende Diplomarbeit bietet eine Erweiterung des Ralph-Systems.

#### 2.5.1.1 Begründung der Wahl von Django und Ralph

Django bietet eine weit verbreitete Open-Source Lösung für die Entwicklung von Web-Diensten. Bekannte Webseiten, die auf Django basieren sind u.a. Instagram, Mozilla, Pinterest und Open Stack. [1] Django zeichnet sich besonders durch die sog. “*Batteries included*” Mentalität aus. Das heißt, dass Django bereits die gängigsten *Features* eines Webserver-Backends standardmäßig innehat. Diese sind (im Vergleich zu Alternativen wie etwa “Flask”) u.a.:

- Authentifikation und Autorisierung, sowie eine damit verbundene Benutzerverwaltung

- Schutz vor gängigen Attacken (wie *SQL-Injections* oder *CSRF*[31]), siehe Views

Zusätzlich bietet Ralph bereits einige *Features*, die die grundlegende Führung und Verwaltung eines herkömmlichen Inventars unterstützen (beispielsweise eine Suchfunktion mit automatischer Textvervollständigung).

## 2.5.2 Kurzfassung der Funktionsweise von Django und Ralph

Im folgenden Kapitel wird die Funktionsweise des Django-*Frameworks*, sowie Ralph beschrieben. Ziel dieses Kapitels ist es, eine Wissensbasis für die darauffolgenden Kapitel zu schaffen.

### 2.5.2.1 Datenbank-Verbindung, Pakete und Tabellen-Definition

Folgende Datenbank-Typen werden von Django unterstützt:

- PostgreSQL
- MariaDB
- MySQL
- Oracle
- SQLite

Die Konfiguration der Datenbank-Verbindung geschieht unter Standard-Django in der Datei `settings.py`, unter Ralph in der jeweiligen Datei im Verzeichnis `settings`. Eine detaillierte Anleitung zur Verbindung mit einer Datenbank ist in der offiziellen Django-Dokumentation [7] zu finden.

Die verschiedenen Funktionsbereiche des Servers sind in Pakete bzw. Module gegliedert. Jedes Paket ist ein Ordner, der verschiedene Dateien und Unterordner beinhalten kann. Die Dateinamen-Nomenklatur eines Packets ist normiert.[6] Der Name eines Packets wird fortlaufend “App-Label” genannt. Standardmäßig ist dieser Name erster Bestandteil einer URL zu einer beliebigen grafischen Administrationsoberfläche des Packets. Pakete werden durch einen Eintrag in die Variable `INSTALLED_APPS` innerhalb der o.a. Einstellungsdatei registriert. Beispiele sind die beiden durch die vorliegende Diplomarbeit registrierten Pakete `"ralph.capentory"` und `"ralph.stocktaking"`

Ist ein Python-Paket erfolgreich registriert, können in der Datei `models.py` Datenbank-Tabellen als python Klassen<sup>6</sup> definiert werden. Diese Klassen werden fortlaufend als “Modell” bezeichnet. Tabellenattribute werden als Attribute dieser Klassen definiert

---

<sup>6</sup> erbend von der Superklasse `Model`[8]

und sind jeweils Instanzen der Klasse `Field`[8]<sup>7</sup>. Datenbankinträge können demnach als Instanzen der Modellklassen betrachtet und behandelt werden. Standardmäßig besitzt jedes Modell ein Attribut `id`, welches als *primärer Schlüssel* dient. Der Wert des `id` Attributs ist unter allen Instanzen eines Modells einzigartig. Die Anpassung dieses Attributs wird in der offiziellen Django-Dokumentation genauer behandelt. [8]

Jedes Modell benötigt eine innere Klasse `Meta`. Sie beschreibt die *Metadaten* der Modellklasse. Dazu gehört vor allem der von Benutzern lesbare Name des Modells `verbose_name`. [10]

### 2.5.2.2 Administration über das Webinterface

Um die Administration von Modelldaten über das Webinterface des Servers zu ermöglichen, werden grundsätzlich zwei Ansichten der Daten benötigt: Eine Listenansicht aller Datensätze und eine Detailansicht einzelner Datensätze.

Die Listenansicht aller Datensätze eines Modells wird in der Datei `admin.py` als *Subklasse* von `ModelAdmin`<sup>8</sup> definiert. Attribute dieser Klasse beeinflussen das Aussehen und die Funktionsweise der Weboberfläche. Durch das Setzen von `list_display` werden beispielsweise die in der Liste anzuzeigenden Attribute definiert.

Die Detailansicht einzelner Datensätze wird grundsätzlich durch die `ModelAdmin` Klasse automatisch generiert, kann aber durch Setzen dessen `form` Attributs auf eine eigens definierte *Subklasse* von `ModelForm`<sup>9</sup> angepasst werden. Diese Klassen werden in der Datei `forms.py` definiert und besitzen, ähnlich der `Model` Klasse, auch eine innere Klasse `Meta`.

Um die `ModelAdmin` *Subklassen* über eine URL erreichbar zu machen, müssen diese registriert werden. Dies geschieht durch den `register` *Dekorator*. Dieser Dekorator akzeptiert die zu registrierende Modellklasse, die zu dem `ModelAdmin` gehört, als Parameter. Die Listenansicht einer registrierten `ModelAdmin` Subklasse ist standardmäßig unter der URL

```
/<App-Label>/<Modell-Name>/
```

erreichbar, die Detailansicht einer Modellinstanz unter der URL

```
/<App-Label>/<Modell-Name>/<Modellinstanz-ID>/
```

---

<sup>7</sup> oder davon ererbende Klassen

<sup>8</sup> Unter `Ralph` steht hierfür die Klasse `RalphAdmin` zur Verfügung.[22]

<sup>9</sup> Unter `Ralph` steht hierfür die Klasse `RalphAdminForm` zur Verfügung.

. Somit repräsentiert die URL der Listenansicht gleichzeitig den Pfad, unter der sie definiert wurde.

Die Dokumentation der Administrationsfeatures von Django ist auf der offiziellen Dokumentationswebseite von Django [5] zu finden.

### 2.5.2.3 API und DRF

Um Daten außerhalb der grafischen Administrationsoberfläche zu bearbeiten, wird eine *API* benötigt. Eine besondere und weit verbreitete Form einer API ist eine *REST-API* [16], die unter Django durch das integrierte *DRF* implementiert wird.[27] API Definitionen werden unter Django in einem Paket in der Datei `api.py` getätigt.

Um den API-Zugriff auf ein Modell zu ermöglichen werden üblicherweise eine `APIView`<sup>10</sup> *Subklasse* und eine `Serializer`<sup>11</sup> *Subklasse* definiert. `APIView` Klassen sind zuständig für das Abarbeiten von Anfragen mithilfe einer `Serializer` Klasse, die die Daten aus der Datenbank repräsentiert und in das gewünschte Format konvertiert. Durch `APIView` Klassen werden Berechtigungen und sonstige Attribute definiert, die sich auf das wahrgenommene Erscheinungsbild des Servers auf einen Client auswirken. Beispiel dafür ist die Art der *Paginierung* [27]. Die erstellten `APIView` Klassen können dann mithilfe einer `Router`<sup>12</sup> Instanz registriert werden. Anleitungen zur Erstellung dieser API-Klassen sind auf der offiziellen Webseite des DRF [27] und der offiziellen Ralph-Dokumentationsseite [23] zu finden.

### 2.5.2.4 Views

Schnittstellen, die keiner der beiden o.a. Kategorien zugeordnet werden können, werden in der Datei `views.py` definiert. Bei diesen *generischen* Schnittstellen handelt es sich entweder um *Subklassen* der Klasse `View`<sup>13</sup> [9] oder vereinzelt Methoden mit einem `request`<sup>14</sup> Parameter. [14] Diese Schnittstellen werden fortlaufend Views genannt.

Soll ein View als Antwort auf eine Anfrage HTML-Daten liefern, so sollte dazu ein *Template* verwendet werden. Mithilfe von Templates können Daten, die etwa durch Datenbankabfrage entstehen, zu einer HTML Antwort aufbereitet werden. Besonders ist hierbei die zusätzlich zu HTML verfügbare Django-Template-*Syntax* [12]. Damit können HTML Elemente auf den Input-Daten basierend dynamisch generiert werden. So stehen beispielsweise `if` Statements direkt in der Definition des Templates zur Verfügung.

<sup>10</sup> Unter Ralph steht hierfür die Klasse `RalphAPISerializer` zur Verfügung. [23]

<sup>11</sup> Unter Ralph steht hierfür die Klasse `RalphAPIViewSet` zur Verfügung. [23]

<sup>12</sup> Unter Ralph steht hierfür die globale `RalphRouter` Instanz `router` zur Verfügung. [23]

<sup>13</sup> die ebenfalls Superklasse der Klasse `APIView` ist

<sup>14</sup> zu Deutsch: Anfrage; entspricht dem empfangenen Packet, meist als HTML.

Die Benutzung von Templates schützt standardmäßig gegen gängige Attacken, wie *SQL-Injections* oder *CSRF* [31] und gilt daher als besonders sicher.

Da reguläre Views nicht automatisch registriert werden, müssen sie manuell bekanntgegeben werden. Dies geschieht durch einen Eintrag in die Variable `urlpatterns` in der Datei `urls.py`. [13]

### 2.5.2.5 Datenbankabfragen

Datenbankabfragen, fortlaufend Queries genannt, werden in Django durch `Queryset`-Objekte getätigt. Das Definieren eines `Queryset`-Objekts löst nicht sofort eine Datenbankabfrage aus. Erst, wenn Werte aus einem `Queryset`-Objekt gelesen werden, wird eine Datenbankabfrage ausgelöst. So kann ein `Queryset`-Objekt beliebig oft verändert werden, bevor davon ausgelesen wird. Ein Beispiel hierfür ist das Anwenden der `filter()` Methode.

In dem folgenden Code-Auszug<sup>15</sup> werden aus dem Modell `Entry` bestimmte Einträge gefiltert:

```
# Erstelle ein Queryset aller Entry-Objekte,  
# dessen Attribut "headline" mit "What" beginnt.  
q = Entry.objects.filter(headline__startswith="What")  
  
# Filtere aus dem erstellten Queryset alle Entry-Objekte,  
# dessen Attribut "pub_date" kleiner oder gleich  
# dem aktuellen Datum ist.  
q = q.filter(pub_date__lte=datetime.date.today())  
  
# Schließe aus dem erstellten Queryset alle Entry-Objekte,  
# dessen Attribut "body_text" den Text "food" beinhaltet, aus.  
q = q.exclude(body_text__icontains="food")  
  
# Ausgabe des erstellten Querysets.  
# Erst hier kommt es zu der ersten Datenbankabfrage!  
print(q)
```

Weitere Beispiele und Methoden sind der offiziellen Django-Dokumentation zu entnehmen. [11]

---

<sup>15</sup> entnommen aus der offiziellen Django Dokumentation [11]



### 2.5.2.6 Designgrundlagen

Designgrundlagen für Django-Entwickler sind auf der offiziellen Dokumentationsseite von Django abrufbar. [20] Die Erweiterung von Django durch die vorliegende Diplomarbeit wurde anhand dieser Grundlagen entwickelt.

Das Konzept des Mixins wird von der Ralph-Plattform besonders häufig genutzt. Mixins sind Klassen, die anderen von ihnen erbende Klassen, bestimmte Attribute und Methoden hinzufügen. Manche Mixins setzen implizit voraus, dass die davon erbenden Klassen ebenfalls von bestimmten anderen Klassen erben. Beispiel ist die Klasse `AdminAbsoluteUrlMixin`, welche eine Methode `get_absolute_url` zur Verfügung stellt. Diese Methode liefert die URL, die zu der Detailansicht der Modellinstanz führt, die die Methode aufruft. Voraussetzung für das Erben einer Klasse von `AdminAbsoluteUrlMixin` ist daher, dass sie ebenfalls von der Klasse `Model` erbt.

... irgendwo im Intro: bezüglich folgender zwei Aspekte: \* Eine Inventurfunktion für die Validierung von Gegenständen und die Verwaltung von Änderungen die durch eine Inventur entstehen, sowie die benötigte Kommunikation mit der Client-Applikation, die für die Durchführung von Inventuren benutzt wird. \* Eine zentralisierte Verwaltungsplattform der in der HTL Rennweg geführten Inventarlisten, besonders der SAP-Datenbank.

Ignore  
below  
content  
for now



## 3 Planung



# A Anhang 1

was auch immer: technische Dokumentationen etc.

Zusätzlich sollte es geben:

- Abkürzungsverzeichnis
- Quellenverzeichnis (hier: Bibtex im Stil plaindin)



# Index

.xlsx: Format einer Excel Datei, 13

API: Application-Programming-Interface - Eine Schnittstelle, die die programmiertechnische Erstellung, Bearbeitung und Einholung von Daten auf einem System ermöglicht, 17, 25

Batteries included: Das standardmäßige Vorhandensein von erwünschten bzw. gängigen *Features*, zu Deutsch: Batterien einbezogen, 14

CMDB: Configuration Management Database - Eine Datenbank, die für die Konfiguration von IT-Geräten entwickelt ist [2], 14

CSRF: Cross-Site-Request-Forgery - eine Angriffsart, bei dem ein Opfer dazu gebracht wird, eine von einem Angreifer gefälschte Anfrage an einen Server zu schicken [31], 15, 18

DCIM: Data Center Infrastructure Management - Software, die zur Verwaltung von Rechenzentren entwickelt wird, 14

Dekorator: Fügt unter Python einer Klasse oder Methode eine bestimmte Funktionsweise hinzu [25], 16

DRF: Django REST Framework - Implementierung einer *REST-API* unter Django [27], 17

Feature: Eigenschaft bzw. Funktion eines Systems, 14, 15, 25

Framework: Eine softwaretechnische Architektur, die bestimmte Funktionen und Klassen zur Verfügung stellt, 14, 15

generisch: in einem allgemeingültigen Sinn, 17

Metadaten: Daten, die einen gegebenen Datensatz beschreiben, beispielsweise der Autor eines Buches, 16

Paginierung: engl. pagination - Die Aufteilung von Datensätzen in diskrete Seiten [24], 17

primärer Schlüssel: engl. primary key, abgek. pk - ein Attribut, das einen Datensatz eindeutig identifiziert, 16

REST-API: Representational State Transfer *API* - eine zustandslose Schnittstelle für den Datenaustausch zwischen Clients und Servern [16], 17, 25

SQL-Injections: klassischer Angriff auf ein Datenbanksystem, 15, 18

Subklasse: Eine programmiertechnische Klasse, die eine übergeordnete Klasse, auch Superklasse, erweitert oder verändert, indem sie alle Attribute und Methoden der Superklasse erbt, 16, 17

Syntax: Regelwerk, sprachliche Einheiten miteinander zu verknüpfen  
[3], 17

Template: zu Deutsch: Vorlage, Schablone, 17

Text to Speech: Umwandlung von Texten in Sprache, 7



# Literaturverzeichnis

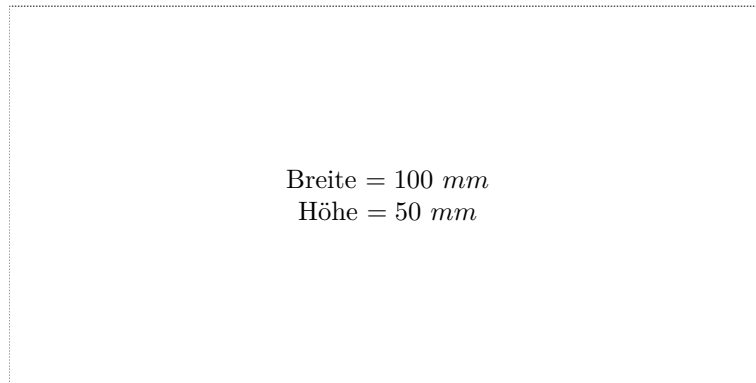
- [1] *Überblick von Django auf der offiziellen Website.*  
<https://www.djangoproject.com/start/overview/>, Abruf: 2020-01-01
- [2] *Datacenter-Insider Webseite mit Informationen über CMDB.*  
<https://www.datacenter-insider.de/was-ist-eine-configuration-management-database-cmdb-a-743418/>, Abruf: 2020-01-02
- [3] *Definition von "Syntax" im Duden.*  
<https://www.duden.de/rechtschreibung/Syntax>, Abruf: 2020-01-02
- [4] *Demo-Webseite des Ralph-Systems von Allegro (Login-Daten: Benutzername: ralph/ Passwort: ralph).* <https://ralph-demo.allegro.tech/>, Abruf: 2020-01-02
- [5] *Django Admin-Dokumentation (Django Version 1.8).*  
<https://docs.djangoproject.com/en/1.8/ref/contrib/admin/>, Abruf: 2020-01-01
- [6] *Django Dateinamen-Nomenklatur.*  
<https://streamhacker.com/2011/01/03/django-application-conventions/>, Abruf: 2020-01-01
- [7] *Django Datenbank-Dokumentation (Django Version 1.8).*  
<https://docs.djangoproject.com/en/1.8/ref/databases/>, Abruf: 2020-01-01
- [8] *Django Datenbank-Modell-Dokumentation (Django Version 1.8).*  
<https://docs.djangoproject.com/en/1.8/topics/db/models/>, Abruf: 2020-01-01
- [9] *Django Dokumentation von klassenbasierten Views (Django Version 1.8).*  
<https://docs.djangoproject.com/en/1.8/topics/class-based-views/>, Abruf: 2020-01-02
- [10] *Django Model-Options-Dokumentation (Django Version 1.8).*  
<https://docs.djangoproject.com/en/1.8/ref/models/options/>, Abruf: 2020-01-01
- [11] *Django Query-Dokumentation (Django Version 1.8).*  
<https://docs.djangoproject.com/en/1.8/topics/db/queries/>, Abruf: 2020-01-01

- [12] *Django Template-Dokumentation (Django Version 1.8)*.  
<https://docs.djangoproject.com/en/1.8/topics/templates/>, Abruf: 2020-01-01
- [13] *Django URL-Dokumentation (Django Version 1.8)*.  
<https://docs.djangoproject.com/en/1.8/topics/http/urls/>, Abruf: 2020-01-02
- [14] *Django View-Dokumentation (Django Version 1.8)*.  
<https://docs.djangoproject.com/en/1.8/topics/http/views/>, Abruf: 2020-01-02
- [15] *Hagenberg LaTeX Thesis Template*.  
<https://github.com/Digital-Media/HagenbergThesis>, Abruf: 2016-10-09
- [16] *Internet-Posting über die Funktion von REST-APIs*.  
<https://www.cloudcomputing-insider.de/was-ist-eine-rest-api-a-611116/>, Abruf: 2020-01-01
- [17] *LaTeX-Kompendium – Wikibooks, Sammlung freier Lehr-, Sach- und Fachbücher*.  
<http://de.wikibooks.org/wiki/LaTeX-Kompendium>, Abruf: 2014-09-14
- [18] *LaTeX-Tipps*. <http://www.siart.de/typografie/latexipps.xhtml#layout>, Abruf: 2014-09-17
- [19] *LaTeX: Umlaute in utf8 Listings korrekt ausgeben*.  
<http://uweziegenhagen.de/?p=1500>, Abruf: 2014-09-17
- [20] *offizielle Design-Grundlagen der Django-Entwickler*.  
<https://docs.djangoproject.com/en/dev/internals/contributing/writing-code/coding-style/>, Abruf: 2020-01-02
- [21] *Die offizielle Django-Website*. <https://www.djangoproject.com/>, Abruf: 2020-01-01
- [22] *Offizielle Dokumentationsseite der Ralph Admin-Klasse von Allegro*.  
<https://ralph-ng.readthedocs.io/en/stable/development/admin/>, Abruf: 2020-01-02
- [23] *Offizielle Dokumentationsseite der Ralph-API von Allegro*.  
<https://ralph-ng.readthedocs.io/en/stable/development/api/>, Abruf: 2020-01-02
- [24] *Offizielle Dokumentationsseite des Paginierungs-Feature im Django Framework*.  
<https://docs.djangoproject.com/en/3.0/topics/pagination/>, Abruf: 2020-01-02
- [25] *offizielle Dokumentationsseite für Python Dekoratoren*.  
<https://wiki.python.org/moin/PythonDecorators>, Abruf: 2020-01-02

- 
- [26] *Die offizielle Flask-Website*. <https://palletsprojects.com/p/flask/>, Abruf: 2020-01-01
- [27] *Offizielle Infopage des Django-REST Frameworks*. <https://www.django-rest-framework.org/>, Abruf: 2020-01-01
- [28] *Die offizielle Pyramid-Website*. <https://trypyramid.com/>, Abruf: 2020-01-01
- [29] *Die offizielle Web2Py-Website*. <http://www.web2py.com/>, Abruf: 2020-01-01
- [30] *Die offizielle Website von "Ralph" des Unternehmens "Allegro"*. <https://ralph.allegro.tech/>, Abruf: 2020-01-01
- [31] *OWASP-Weiseite mit Informationen über CSRF*. [https://www.owasp.org/index.php/Cross-Site\\_Request\\_Forgery\\_\(CSRF\)](https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)), Abruf: 2020-01-01
- [32] *Reife- und Diplomprüfung, Abschlussprüfung an technischen, gewerblichen und kunstgewerblichen Lehranstalten*. [https://moodle.htl.rennweg.at/MoodleKurs-Matura/HTL\\_RDP-AP\\_Leitfaden.pdf](https://moodle.htl.rennweg.at/MoodleKurs-Matura/HTL_RDP-AP_Leitfaden.pdf), Abruf: 2018-04-24
- [33] *Urheberrechtsgesetz*. <http://www.ris.bka.gv.at/Dokumente/Bundesnormen/NOR40041632/NOR40041632.html>, Abruf: 2014-09-14
- [34] DITTRICH, Robert: Zur Quellenangabe bei Zitaten. In: *Urheberrecht im Informationszeitalter. Festschrift für Wilhelm Nordemann*. (2004), S. 617–624. ISBN 3406516831
- [35] KOPKA, Helmut ; RAHTZ, Sebastian: *LaTeX: Einführung*. Bd. 2. Addison-Wesley, 1991
- [36] WIKIPEDIA: *Quellenangabe — Wikipedia, Die freie Enzyklopädie*. <http://de.wikipedia.org/w/index.php?title=Quellenangabe&oldid=131343956>, Abruf: 2014-09-14
- [37] WIKIPEDIA: *Zitat — Wikipedia, Die freie Enzyklopädie*. <http://de.wikipedia.org/w/index.php?title=Zitat&oldid=134064196>, Abruf: 2014-09-14



— Druckgröße kontrollieren! —



— Diese Seite nach dem Druck entfernen! —

Diese  
Seite  
nach dem  
Druck  
entfer-  
nen!