

manage me

A Django app to manage your business tasks

Ulrike Exner

FH Brandenburg

Matr.-Nr. 20149502

WiSe 2015/16

1. Aufgabenstellung

Als Ergebnis des Moduls *Objektorientierte Skriptsprachen* sollte ein Semesterprojekt mit der Programmiersprache **Python** entstehen, wobei Django als Framework herangezogen werden konnte. **Django** ist ein Python geschriebenes quelloffenes Web Application Framework, das einem Model-View-Presenter-Schema folgt. Als Webserver fungiert ein Server der **Amazon Cloud Services** mit der IP 52.29.42.195 (ec2-52-29-42-195.eu-central-1.compute.amazonaws.com). Mit dem **Bitnami Django-Stack** ist es möglich, einen Apache und eine komplette Django-Installation mit vielen Packages, sowie u.a. einer MySQL-Datenbank zu installieren.

3. Projektbeschreibung

Entstehen soll eine Web Application, die es ermöglicht, Dienstleistungsaufträge zu verwalten (z. B. für Freelancer). Aufträge setzen sich dabei aus Kunden, dem Angebot, einer Dienstleistungsart und Rechnungen zusammen. Aufträge können neu angelegt, bearbeitet und gelöscht werden. Sie erhalten Anlagen eine eindeutige ID. Das Angebot, die Dienstleistungsart und die Angabe des Kunden sind Pflichtfelder. Rechnungen können auch später eingefügt werden. Zu jedem Auftrag kann es nur ein Angebot geben, jedoch mehrere Rechnungen. Die Angebote können heruntergeladen werden.

Kunden, Dienstleistungsarten und Rechnungen können bisher nur neu angelegt werden. Auch sie erhalten eine eindeutige ID, wobei Rechnungen darüber hinaus noch eine frei zu wählende Rechnungsnummer tragen können.

Unter der URL: <http://52.29.42.195/Project/auftraege/> ist eine Gesamtübersicht aller Aufträge verfügbar.

Die Navigation wird durch eine einfache Menüführung in der linken Spalte gewährleistet. Von dort können neue Aufträge, Rechnungen, Dienstleistungen und Kunden angelegt werden:

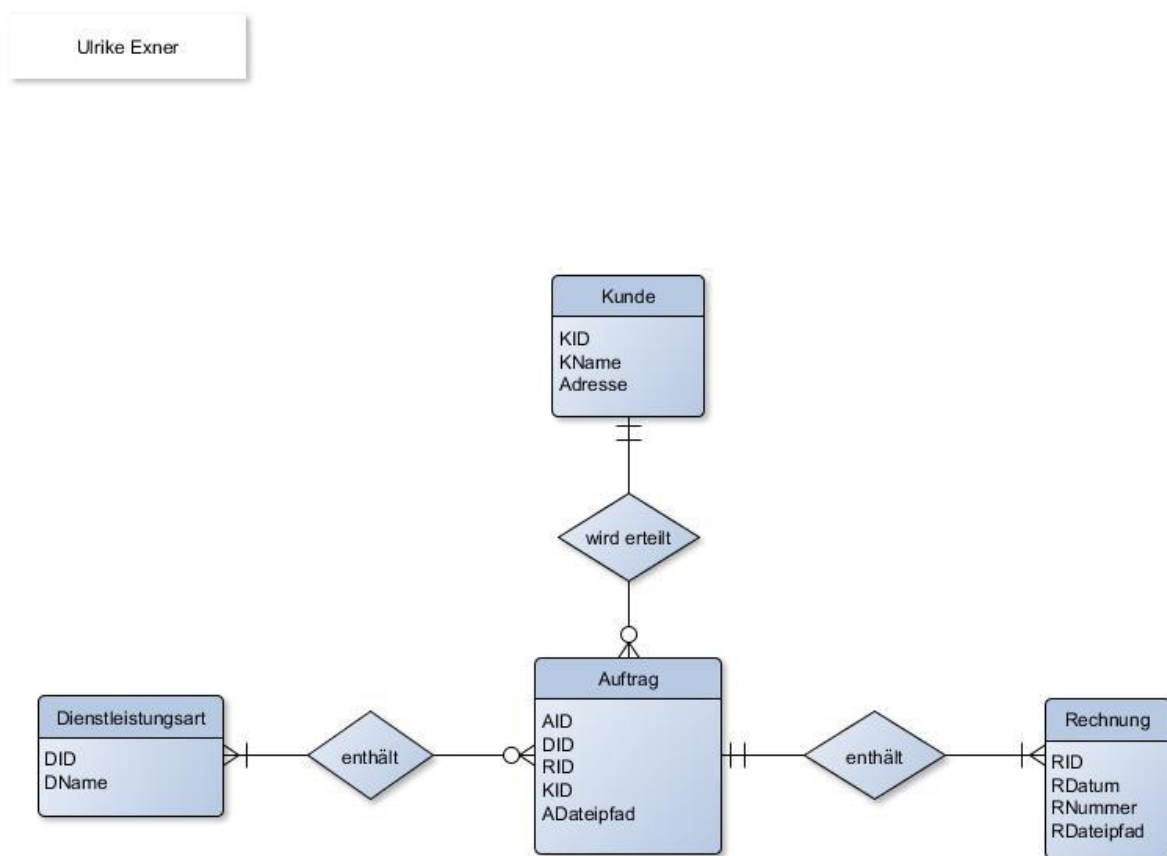
<http://52.29.42.195/Project/auftrag/add/>

<http://52.29.42.195/Project/rechnung/add/>

<http://52.29.42.195/Project/service/add/>

<http://52.29.42.195/Project/kunde/add/>

Das nachfolgende ER-Diagramm zeigt das zugrunde liegende Datenmodell, bzw. das Model, das Django zum Anlegen der Tabellen und Beziehungen verwendet.



Als Beispiel stellt sich demnach die Klasse Dienstleistungsart in **models.py** folgendermaßen dar:

```
class Dienstleistungsart(models.Model):
    dname = models.CharField(max_length=30)
    did = models.AutoField(primary_key=True)

    def __str__(self):
        return self.dname
```

4. Umsetzung

Nach dem Aufsetzen des Django-Stacks auf einem neuen Server der Amazon Cloud Services wurde die zuvor lokal programmierte Django-App dort zum Laufen gebracht werden. Django gewährleistet dabei ein gutes Handling von Datenbanken, so dass sowohl die lokal getestete SQLite-Datenbank als auch die MySQL-Datenbank sehr gut mit der App verknüpft werden können. Beim Programmieren der App ist das Anlegen der **urls.py**, der Formulare in **forms.py**, der **views.py** sowie der Template-Files essentiell.

Das Handling der static- und media-Files sowie die Anbindung an die MySQL-Datenbank wird in der **settings.py** vorgenommen.

Um das Frontend ansprechender zu gestalten, wurde auf ein Bootstrap-Template von <http://www.bootply.com/> zurückgegriffen. Die verwendeten css, js-Dateien und Bilder werden von Django als static-Files angesprochen.

5. Source Code

Der komplette Source Code (ohne media-Files) kann auf github nachgeschlagen werden:

<https://github.com/rika087/manage-me>

Verwendete Packages (requirements.txt):

```
boto==2.6.0
Django==1.8.7
django-bootstrap3==6.2.2
docutils==0.10
elementtree===1.2.7-20070827-preview
gevent==1.0.1
greenlet==0.4.3
html5lib==0.90
lxml==3.3.6
Markdown==2.0.3
MarkupSafe==0.15
mercurial==2.1
```

MySQL-python==1.2.3rc1

pbr==0.10.0

PIL==1.1.7

Pillow==2.6.1

psutil==0.6.1

psycopg2==2.6

pycairo==1.8.4

pycrypto==2.6.1

pydot==1.0.28

pyparsing==1.5.5

pysqlite==2.6.3

python-ldap==2.4.10

python-Levenshtein==0.10.2

python-memcached==1.45

python-openid==2.2.5

PyXML==0.8.4

PyYAML==3.10

reportlab==2.5

simplejson==2.0.9

six==1.7.3

stevedore==1.1.0

virtualenv==1.11.6

virtualenv-clone==0.2.5

virtualenvwrapper==4.3.1

ZSI==2.0