EXPENSIO – Expense Tracker

Web project using Django & HTML/CSS)

Alexandra Jelea – Kaoru Steinböck – Gerhard Nägele | SS 20 Information Systems Development| University Liechtenstein

CONTENT

1. Project Idea – EXPENSIO
2. Software & Tools
3. Data Model
4. Demo
5. HTML Documentation - Sphinx’s
6. Project Documentation PDF – inclusive User Manual
7. Version Control – Github
8. Challenges & Findings

1. EXPENSIO

Project Idea

The basic idea is to program a **simple expense tracker** using Django, with which you can **save expenses**, **assign categories**, **create budgets** for different expenses and a simple **csv export and import** function as well as overviews via **charts.** The App should be **responsive** and follow an intuitive usage concept.

- Tracking Expenses (CRUD)
- Import/Export Function
- Create Budget (CRUD)
- Categories (CRUD)
- User Management (Multiuser compatible)

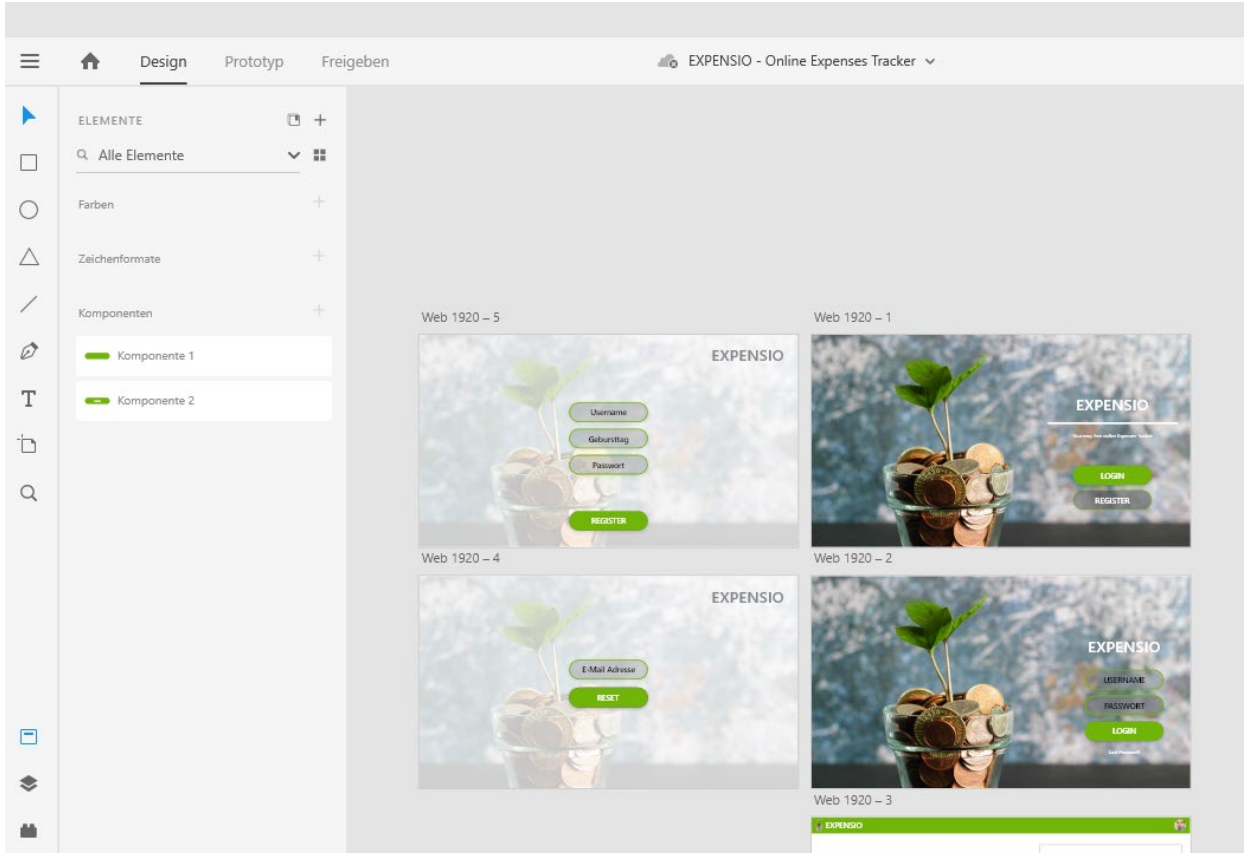


Fig. 1 : Mockup developed with Adobe XD

2. SOFTWARE & TOOLS

Programming Language	**Python, HTML, JavaScript**
Webframework	Django V. 3.0.6
Mockup Tool	Adobe XD
IDE / Editor	PyCharm / Emacs
Database	SQLite V. 3
OR-Mapper	Django-ORM
Version Control System	Github, Git Version 2.26.0
Documentation	Docstrings Sphinx's
Project Management	Github Kanban Board

External Packages/Libraries

django-crispy-forms V. 1.9.1, Bootstrap V. 4.4.1, Chart.js V. 2.9.3, Font Awesome

3. DATA MODEL

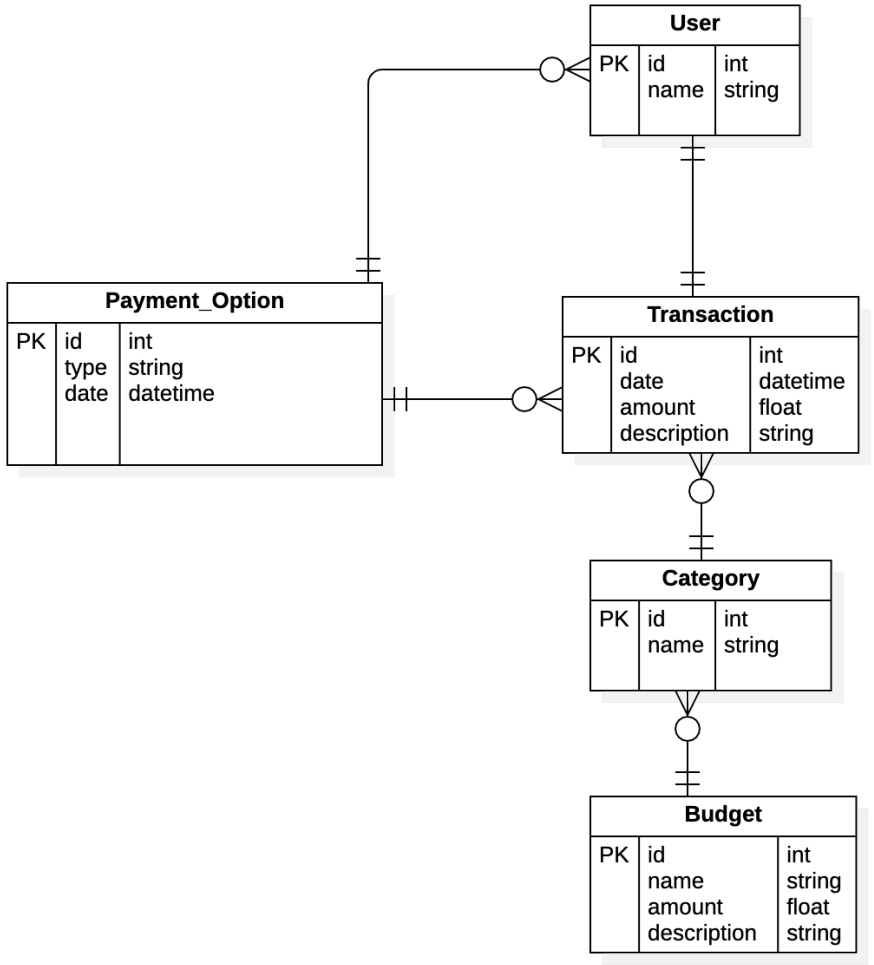


Fig. 2 : ER DIAGRAM EXPENSIO

DATA MODEL: APP transaction_expense

```
class Transaction_Expense(models.Model):
    date = models.DateTimeField()
    amount = models.DecimalField(max_digits=19, decimal_places=2)
    description = models.CharField(max_length=200, default="")
    user = models.ForeignKey("auth.User", on_delete=models.CASCADE)
    category = models.ForeignKey(Category, on_delete=models.CASCADE, null=True)
    payment = models.ForeignKey(Payment, on_delete=models.CASCADE, null=True)
```

DATA MODEL: APP payment

```
class Payment(models.Model):
    VISA = "VISA"
    MASTERCARD = "MASTERCARD"
    BANK = "BANK"
    CRYPTO = "CRYPTO"
    CASH = "CASH"
    PAYMENT_TYPE = [
        (CASH, "Cash"),
        (VISA, "Visa"),
        (MASTERCARD, "Mastercard"),
        (BANK, "Bank"),
        (CRYPTO, "Crypto"),
    ]
    type = models.CharField(max_length=50, choices=PAYMENT_TYPE, default=CASH, )
    date = models.DateTimeField()
    description = models.CharField(max_length=100, default="")
    user = models.ForeignKey("auth.User", on_delete=models.CASCADE)

    def __str__(self):
        return f"Type: {self.type}, Description: {self.description}"
```


EXPENSIO
Login
Register
EXPENSIO
The Open Source web-based Expense Tracker build with Django.
Register now for Free!
DEMO
EXPENSIO ©2020

5. HTML Documentation - Sphinx’s

Fig. 3 : Screenshot Sphinx’s HTML Documentation Page

Can be found under: SS-20-Information-Systems-Development---Expense-Tracker\doc_build\html

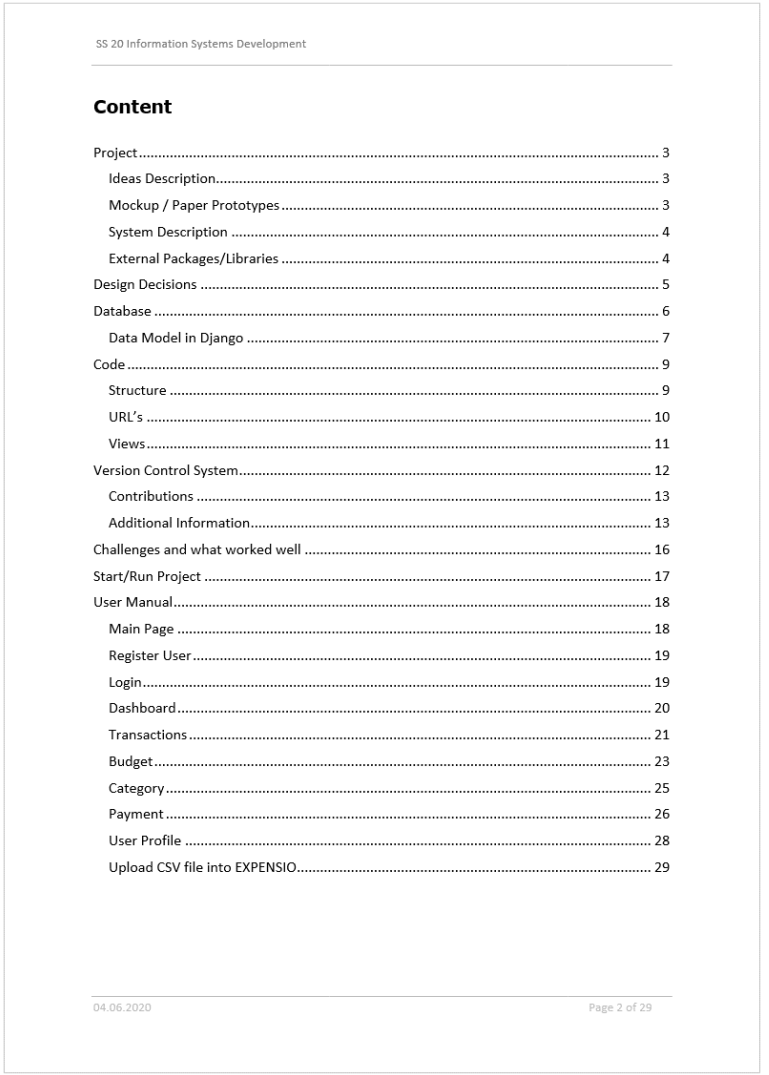
DEMO

6. PROJECT Documentation – User Manual

PROJECT Documentation

- PDF on GITHUB and ZIP Submission File
- Documentation of used Technologies, Software, Design Decisions
- Project Installation and Start Procedure description
- User Manual with Description of Functionality

DEMO



SS 20 Information Systems Development

Content

Fig. 4 : Content of Documentation PDF

7. VERSION CONTROL

VCS was realized with Github

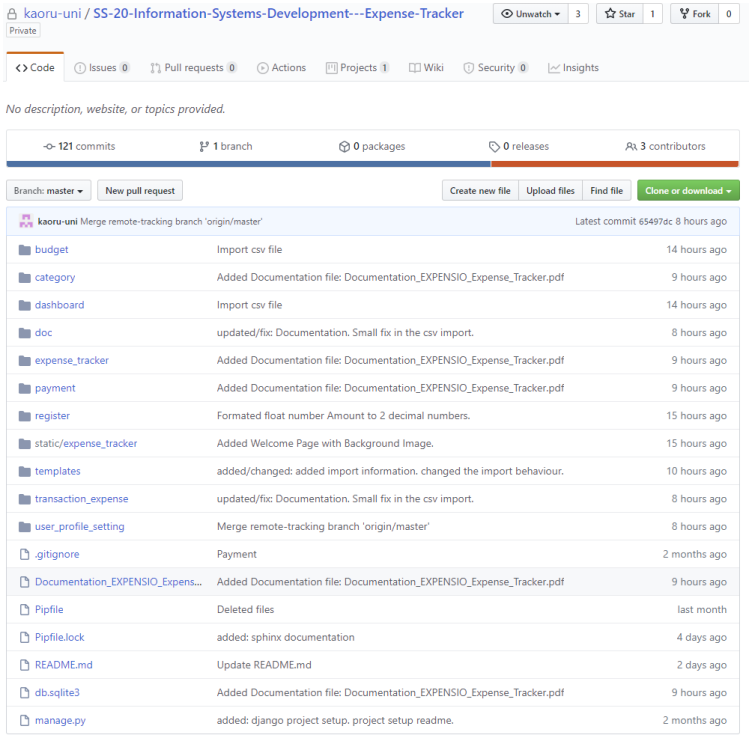


Fig. 5 : Screenshot Github Repository 04.06.2020

Fig. 6 : Screenshot Github Contribution 04.06.2020

DEMO

8. CHALLENGES & FINDINGS

CHALLENGES:

- Python was a **new language** for some of us
- Difference between Class and function based programming
- Understand **the MVT Pattern**
- VCS – we had some issues with **merge conflicts**. (These do not appear with git but with the PyCharm build in VCS Module)

FINDINGS:

- Django has a **big Community**. Solution for every problem we encountered.
- In Contrast to other Programming Frameworks, **Django Documentation** is easier to read/understand.
- Even **older Solutions from Stackoverflow** worked e.g. 2012
- Django does most of the Job (ORM, Return Values, Forms,…)