In aeternum

<in aeternum>
Software Architecture Document

Version <1.0>

Revision History

Date	Version	Description	Author
23. 05. 2017	1.0	Intial Fill up	Louisa
7.1.2017	2.0	final	Max

In aeternum

Table of Contents

In aeternum

Software Architecture Document

1. Introduction

1.1 Purpose

This document provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system.

2. Architectural Goals and Constraints

Since general customer information (real name, date of birth, etc) as well as more private data related to a customer's death and funeral is used within the application and service, data security was of high importance.

3. Use-Case View

3.1 A user can create a new account with some basic information. That account can then be used to plan their own funeral and related events. The intention was to help them overcome the societal stigma that prevents most people from planning their own funeral ahead of time, leaving all of the work and decisions to their families.

4. Logical View

As dictated by Django, the used framework, the entire application was built around the Django views. Necessary forms and data inputs were generated from Django database models, which were also used to create all tables in the database management system.

5. Deployment View

As of now, the application has not been deployed on any webserver. The application can, however, be deployed on any computer running Python 3 with the Django package installed.

6. Data View (optional)

As stated previously, the application is using the Python-based framework Django. Since Django is database agnostic, most common database systems can be used for deployment. For our test system, SQLite was used, however, albeit possible, this is not advised for actual deployment.

7. Size and Performance

The hardware requirements (CPU, RAM, bandwidth, etc.) are mostly dependent on the number of users served by the application and can thus not be estimated accurately.