Initial Design and Architecture

The main components of the web application

The front-end contains HTML files corresponding to each main page on the website, as well as a sign up, login, and delete account page. These are styled to match what is specified in the UI wireframe. In order to display the correct data, the front end need to be able to communicate with the backend. The web page therefore contains a backend layer containing functions to handle the different frontend requests. These functions communicate with the database and return information to the page. Website data is stored in a SQLite database, and accessed through the model.

How the Django framework is being used.

The system has been designed to have three main layers, following Django's Model, View, Template structure. A model layer is used to define the database's structure and handle communications with the database. The system also has a template layer, which defines how the web page will be shown to users. It is comprised of HTML pages, which present the website's user interface. Finally the system has a view layer. This receives the requests and responds to them, by communicating with the model to retrieve and update data, and also passing data to the template for it to display and render.

Interaction between various system components

Interactions follow a client**-**server architecture, where users interact with the system via a web-based front end, while the back end handles data processing, authentication, and task/event management.

Basic overview of the chosen database structure

The website's data is stored in a relational structure, using SQLite. To represent each user, the CustomUser table extends the User table provided by Django. The users' pets are represented by a Pet table. The database also consists of Task and Event tables, as well as UserTask and UserEvent tables, which connect the two, storing whether they have been completed.