
Ikindo WebEdit

**Ikindo WebEdit
Software Architecture Document**

Version 1.0

Revision History

Date	Version	Description	Author
22.12.2020	1.0	Filled in V1.0 of the SAD	Moritz, Jonas A., David, Jonas S.
11.04.2021	1.1	Visual Update of the Diagrams	Jonas S.
21.04.2021	1.2	Changed Scope	Jonas A.
03.05.2021	1.3	Update of the UseCase Diagram	David

Table of Contents

1.	Introduction	4
1.1	Purpose	4
1.2	Scope	4
1.3	Definitions, Acronyms, and Abbreviations	4
1.4	References	4
1.5	Overview	4
2.	Architectural Representation	5
3.	Architectural Goals and Constraints	5
4.	Use-Case View	6
4.1	Use-Case Realizations	6
5.	Logical View	8
5.1	Overview	8
5.2	Architecturally Significant Design Packages	8
6.	Process View	8
7.	Deployment View	8
8.	Implementation View	8
8.1	Overview	8
8.2	Layers	8
9.	Data View (optional)	9
10.	Size and Performance	9
11.	Quality	9

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.

1.2 Scope

This document will describe how our software architecture will be build up and organized. It will define were specific data will be saved and how. Affected by these documents are underlying client-server connection happening all the time you interact with the website.

1.3 Definitions, Acronyms, and Abbreviations

Abbreviation	Description
n/a	Not Applicable
MVC	Model-View-Controller
MVT	Model-View-Template

[Use case diagram](#)

[Activity diagram for view graphical editor use case](#)

[Activity diagram for view site statistics use case](#)

[Activity diagram for view html editor use case](#)

[Feature file for site statistics](#)

[Feature file for the graphical editor](#)

[Feature file for the html editor](#)

<https://github.com/IkindoWebEdit/ikindo-code>

<https://github.com/IkindoWebEdit>

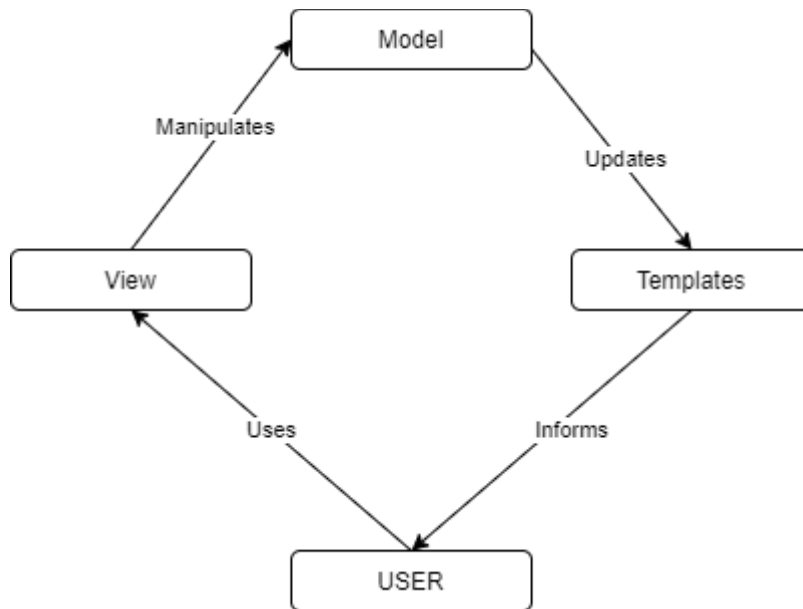
<https://ikindowebedit.wordpress.com/>

1.4 Overview

We need a client server construction as well as a database to store information such as passwords and user data. This document will be used to give newcomers of our project as well as our customer an overview about the software architecture of the project. Information in this document is not fixed and are subject to change.

2. Architectural Representation

Django adheres to the Model-View-Template design pattern rather than the MVC pattern, this pattern can be seen here:

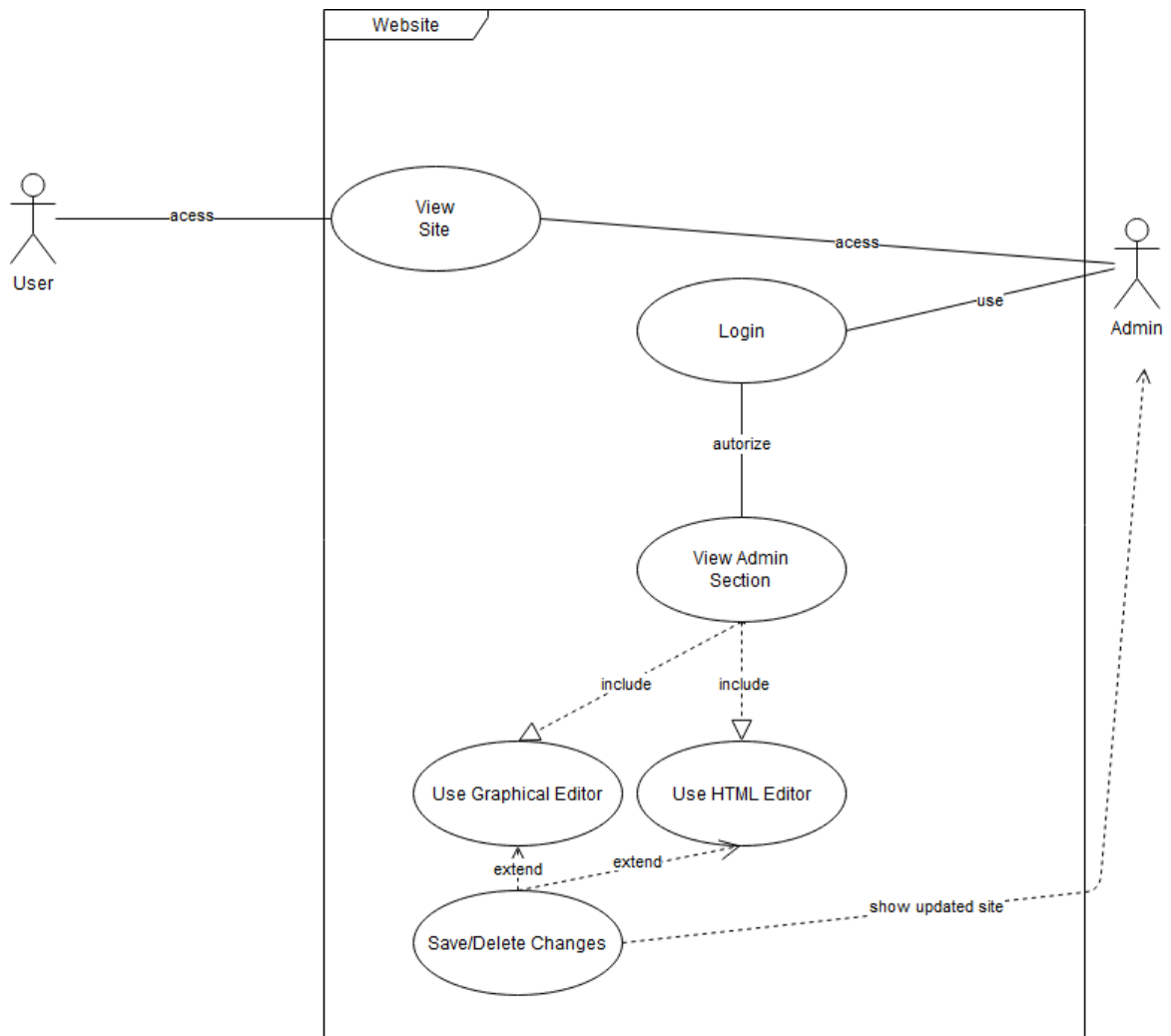


3. Architectural Goals and Constraints

Our frontend consists of JavaScript, CSS and HTML.

As our backend we use Django. Django is a Python based web framework. The main goal of Django is to simplify the creation of complex, database-driven websites, which perfectly fits our needs. Due the frontends functionality comes from JavaScript our Project will not contain any class structure. The Javascript will most likely contain functions triggered by the user's interaction.

4. Use-Case View



4.1 Use-Case Realizations

4.1.1 UC View HTML-Editor

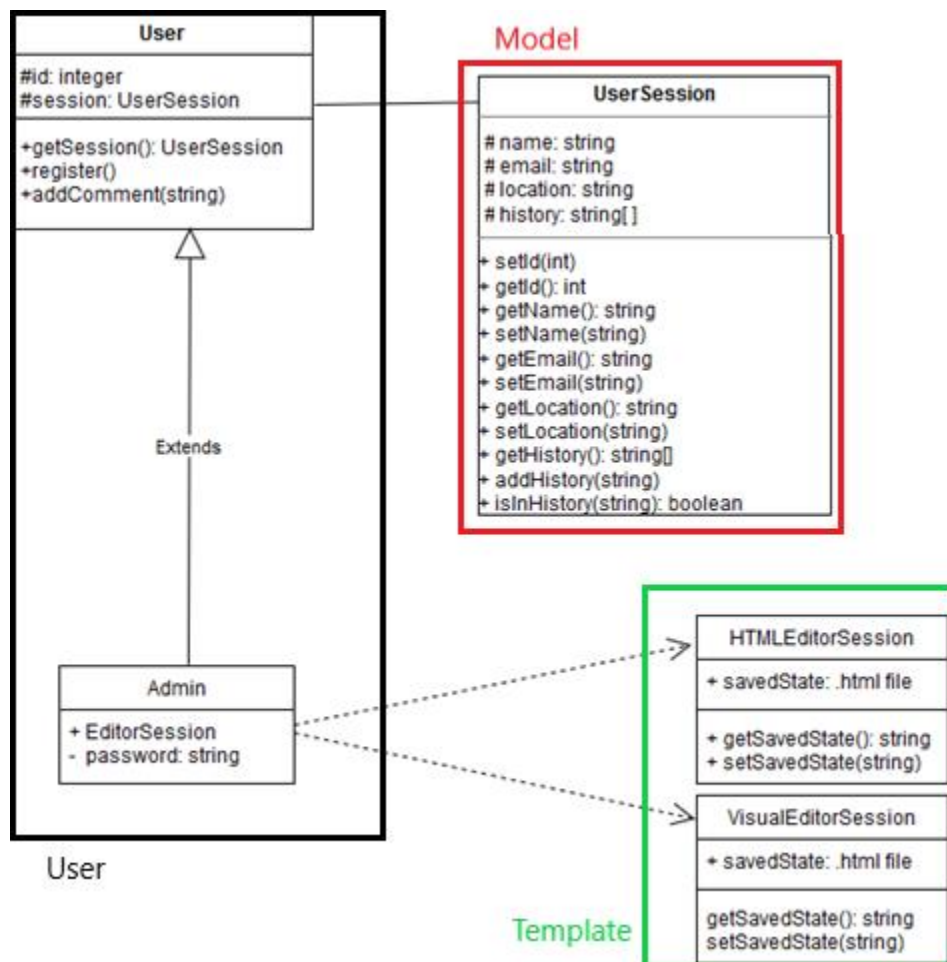
- **Brief Description:**
The user is logged in on the Admin-Section and clicks on HTML-Editor. A new site with a text-editor opens up, in which the HTML-Code itself can be edited textually.
- **Activity Diagram:**
https://github.com/IkindoWebEdit/ikindo-docs/blob/main/ActivityDiagram_ViewHTMLEditor.png
- **.feature File:**
https://github.com/IkindoWebEdit/ikindo-docs/blob/main/Narrative_HTMLEditor.png

4.1.2 UC Get Graphical-Editor

- **Brief Description:**
The user is logged in on the Admin-Section and clicks on Graphical-Editor. The site with the editor opens up and gives the user several (and easy to use) opportunities to edit his website.
- **Activity Diagram:**
https://github.com/IkindoWebEdit/ikindo-docs/blob/main/ActivityDiagram_ViewGraphicalEditor.png
- **.feature File:**
https://github.com/IkindoWebEdit/ikindo-docs/blob/main/Narrative_GraphicalEditor.png

5. Logical View

5.1 Overview



5.2 Architecturally Significant Design Packages

Our Templates make up the view-part of the MVC-pattern. The model consists of our user information Database and the stored web-edit-sessions. This can not really be seen on the diagram above, as we don't have the framework fully populated right now.

6. Process View

n/a

7. Deployment View



8. Implementation View

[Link to our code\(GitHub\)](#)

9. Data View

n/a

10. Size and Performance

n/a

11. Quality

n/a