**NoteOfTheEverything**

System Design Document

1.0

26.04.2021

Anzel Öztürk 215SE2303

Seray Şimşek 216CS2025

Prepared for

COMP/SOFT4902 Graduation Project



**SYSTEM DESIGN DOCUMENT**

1. **Introduction**

NoteOfTheEverything is the website that wanted to be created as a note sharing website. Working principles of the website are user-oriented and user-friendly design for users and admins. Our main purposes on designs are, system to be easy to use, to be safe, system’s react time to be fast and reliable.

* 1. **Purpose of the System**

Main purpose of NoteOfTheEverything is to create a website that anyone can use without help, find what they are looking for their lessons with ease with the help of our ChatBot.

* 1. **Design Goals**

**Usability:**

* NoteOfTheEverything is user-oriented, so every user can easily understand and use it.
* Admins should be able to easily approve to be published on the notes or delete notes that are not suitable for publication to other users.

**Information Security:**

* NoteOfTheEveryThing protects user’s personal information and password using PBKDF2 algorithm with a SHA256 hash.

**Modifiability:**

* When the NoteOfTheEverything comes across a problem about lecture notes, admin can easily fix it.

**Reliability:**

* All the notes that are sent by the user are double checked with admin approval so the system is purified from possible errors.

**Performance:**

* NoteOfTheEverything checks all the requested uploads from users and the system reacts fastly to the pending notes processes.

**Maintainability:**

* NoteOfTheEverything is a user action base website which only admins need to check incoming uploads by the users, by that way the system can be easily repaired.

**Packaging:**

* Users can easily upload documents on the system and users easily download the notes.
  1. **Definitions, Acronyms, and Abbreviations**

SDD: System Design Document

RAD: Requirement Analysis Document

MVC: Model View Controller Architecture

MVT: Model View Template Architecture

HTML: Hypertext Markup Language

CSS: Cascading Style Sheets

JS: Java Script

UI: User Interface

DB: Database

DTL: Django Template Language

NIST: National Institute of Standards and Technology

SHA256: Secure Hash Algorithm 256

PBKDF2: Password-Based Key Derivation Function

* 1. **References**

-Requirement Analysis Document (RAD) is based for overall functional and nonfunctional requirements on our project.

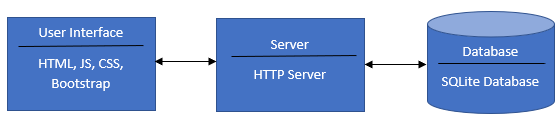
-“Not kutusu” and “Notedu” was focused for developing and improving of our system.

1. **Current Software Architecture**

As we mentioned in RAD, we researched the most searched note sharing websites such as “NotKutusu” and “Notedu”, as we have observed that these sites are complicated for the users. These website’s programs and software based on Python programming language, JS, CSS and HTML languages. They use DB programs for their data such as MySQL. NoteOfTheEverything system has a web UI for users and based on an authentication system for admins and users. NoteOfTheEverything system has a user-oriented interface different from the other note sharing websites.

1. **Proposed Software Architecture**
   1. **Overview**

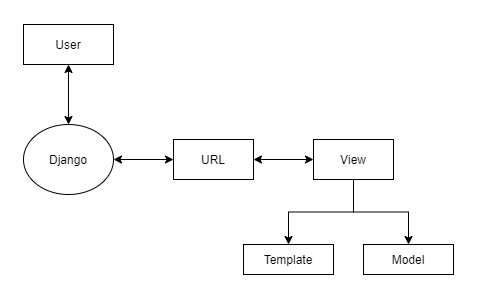
NoteOfTheEverything is a web application because of this we used the Django framework and it is a high-level language also Python Web framework uses open source. For the user interface, we used HTML5, JavaScript and CSS. We used Python HTTP as a server and it combined databases and frontend each other. Our system is a web application and Django based, for this reason we used MVT as an architecture. First of all, the model layer is in charge of dealing with data and lets users make changes like update or insert their information. This user information sent to the controller and controller transfers to the model layer and keeps in the database. Second layer is the view layer which includes the user interface side and frontend such as HTML, CSS etc... We see the data that is stored in the model layer through this layer. Finally, the last layer is the template layer and this layer comprises static parts of the HTML and process user interface.



* 1. **System Decomposition**

In the NoteOfTheEverything website, users should login to the system. Login subsystem is essential for authentication of users and access to the website. Also, registration is compulsory for creating an account and sharing notes. When the user logs in to the system, the first page user will see is the Homepage. In this page, the user can search notes and access his/her accounts, also the system's tasks. Users can select notes through this page and make decisions about his/her wants. User’s information is stored in the Account subsystem and ensures services for instance selecting and adding favorite notes or editing a user's profile. In the admin side of the system, there is a confirm page and admin can publish or remove notes through this page. Also, with this subsystem, users can reach approved notes.

TheNoteOfEverything website is based on the Django framework and MVT architecture.

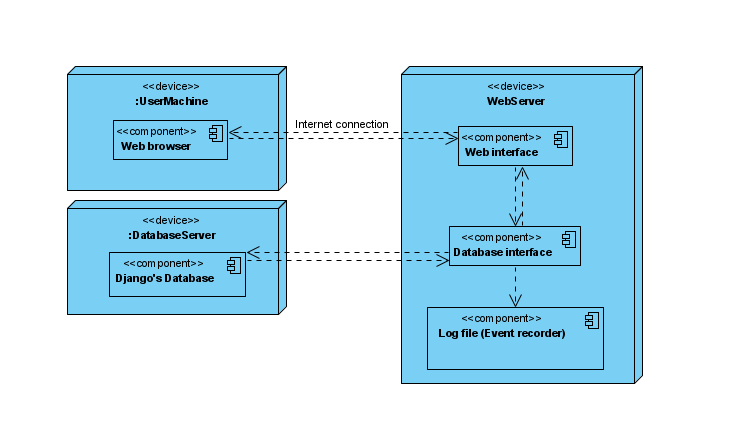


There is a slight difference between MVT and MVC. The main difference between two models; Django makes the controller part which is the software created for controlling the actions between the Model and View. Also, as the template is left, it is a HTML file basically a mix of Django Template Language (DTL).

As it can be seen from the upside, the “Model” part is the section in which the database and its operations are done, also as Django used, we can make a database most of the software languages that Django supports. One of the parts is the “View” part, which is the section in template and model within. This part is the location where all the codes that are written are located. General use of this part is that we take codes that we want to use in model and template, as python codes from this part. İn the last part of the schema is the “Template” part, which is the part that is seen from outside, in this part we manage how the page is designed for the user.

* 1. **Hardware/Software Mapping**

In this hardware diagram general connection principles on our system are shown. User machine connects to the system via the internet and after the connection is done, the web interface reacts with the database interface for data communication. All the events done in the operating system are recorded into the log file for checking and the database interface communicates with the database server.



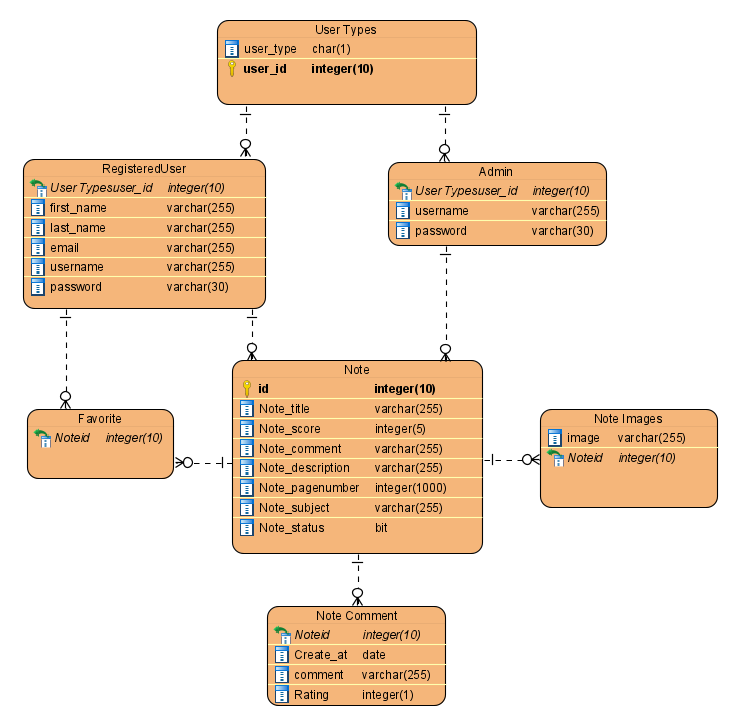
* 1. **Persistent Data Management**

In the section of persistent data management, we describe our system’s data management. As we mentioned in the RAD, our website is based on the Django framework and Django supports SQLite for databases. So, NoteOfTheEverything system’s relational database will be created by SQLite database. We make inferences to use SQLite because it has easy installation and user-friendly access. Also, SQLite has a quality that it creates a database as a single file and thanks to this feature using SQLite, makes our tasks much easier.

Django is a useful framework in terms of databases because it automatically makes bonds between persistent connections and databases. Furthermore, for accessing databases Django implements authentication mechanisms automatically.

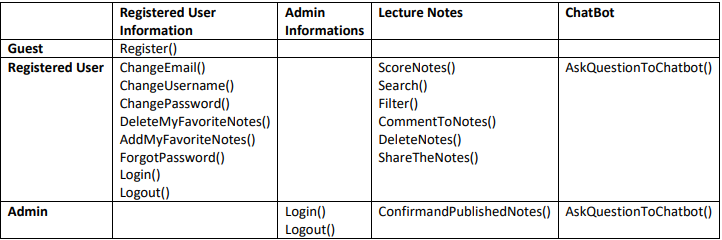
Django did not keep static data in the database such as images or CSS files. Instead of storing this static data in the database, Django keeps these files in a static folder.

Relational database schema of the NoteOfTheEverything is as shown below:



* 1. **Access Control and Security**

As the privacy of people is the most ethical area these days, we will be using the best possible security system in the NoteOfTheEverything. When the user login to the system, the system will verify if the user is real, if the verification email process is done and the system will ask for a password which was created by the user, that requires one upper case letter, at least one number and one symbol. The system is using Django base structure, which increases the security by using its own encryption type. This encryption is done by PBKDF2 algorithm with a SHA256 hash, and additionally the system has a security password extension mechanism done by NIST. For the user side, as the NoteOfTheEverything is a website which users acts affect the system, for blocking the possible system attacks to jam, Django contains an automatic clickjacking protection. By these preventions and as the only admins are available to reach the control systems, the system will be protected.

****

* 1. **Global Software Control**

This section describes the global software control of our NoteOfTheEverything Website. This website is accessed via the internet and uses website browsers like, Google Chrome, Internet Explorer, Firefox etc… User needs for accessing our site is an internet connection and web browser.

Django has an event-driven control flow. Whenever a user sends a request to the database, the system responds to these requests and displays what is asked for. Such as, when a user clicks on a register part which clicks are events, our website responds to the user with an open register page.

* 1. **Boundary Conditions**

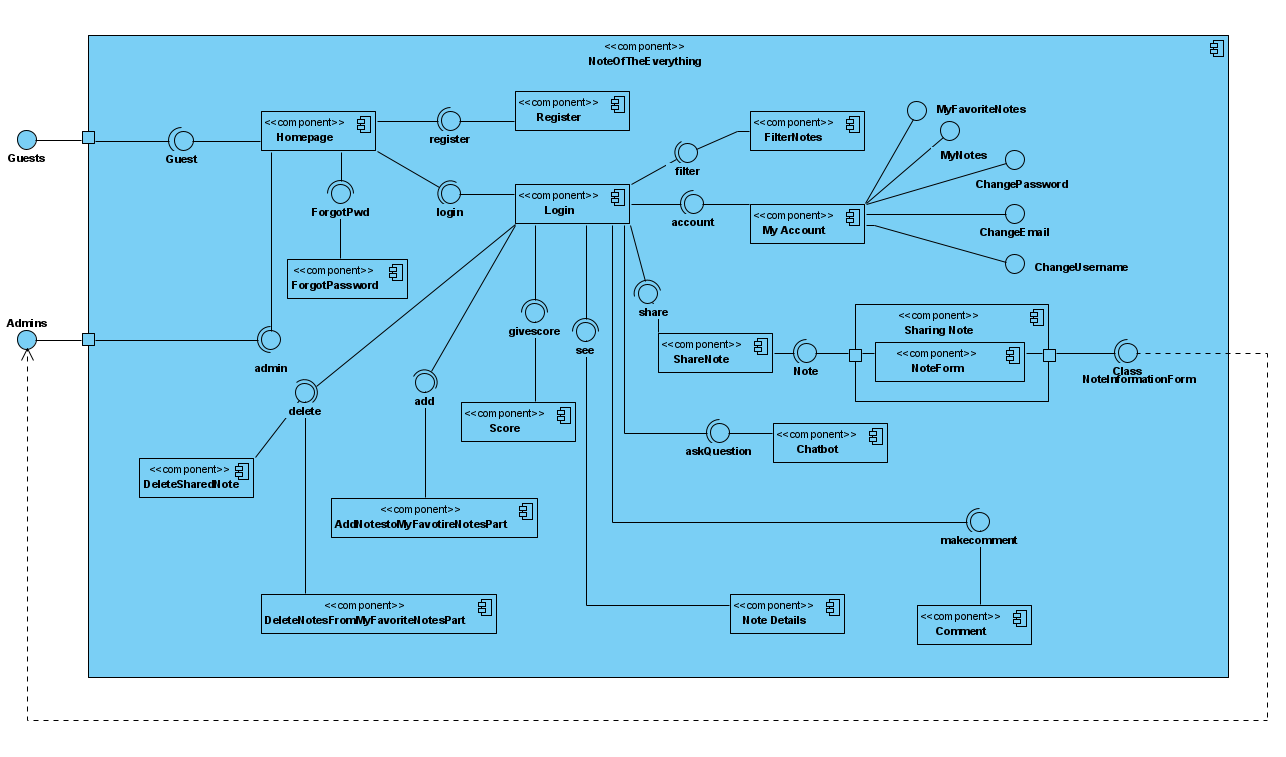
**Initialization:** NoteOfTheEverything system has three users such as visitor, registered user and admin. Visitors only can visit the website and make registration using a browser and internet connection. Registered users have an account in the system and they can login and access websites features through browser and internet connection. Registered users can search or add notes to the system. Admin can login to the website through browser and internet connection in their computer. Admin can make a confirmation about notes and decide to publish notes on the website or not. Except visitors, all users must fill the blanks with their username and password to authenticate. For login to our website, users must enter correct information, otherwise the user cannot have logged in and there will be an error message to the user.

**Termination:** NoteOfTheEverything website has a logout button for exit the system. When the user clicks the logout button, the system exits the user's account and cuts the user's session. As long as the user does not click the logout button, he/she remains logged in.

**Failure:** NoteOfTheEverything website, users must save changes that they did in the site such as editing profile information or adding a note to the system before logged out. In case of connection error, the website saves data only if it is saved by users. If there is an error about unsaved changes, the website sends a message to the user before the user logs out.

1. **Subsystem Services**

NoteOfTheEverything system has a related operation which is a subsystem and each of these subsystems have a connection with other necessary components. Users only visit the website's Homepage which includes a login and register page unless they are logged in or register to the system. When a user registers to the system, the website creates an account for the user and directs the user to the login subsystem. Login subsystem is necessary for the authentication of the user and letting the users enter their account or searching notes. User’s accounts have multiple services such as updating email or changing passwords. Users can access his/her notes or favorite notes through this Account subsystem. The Select note component is useful for notes that are selected upon user’s wants. Admin side of the system requires a confirm subsystem.



1. **References**
2. Bruegge B. & Dutoit A.H.. (2010). *Object-Oriented Software Engineering Using UML, Patterns, and Java*, Prentice Hall, 3rd ed.
3. RAD (Requirement Analysis Document)

1. <https://docs.djangoproject.com/en/3.1/topics/security/>

1. <http://notkutusu.com/>

1. <https://notedu.com/>

1. <https://www.javatpoint.com/django-mvt>