|  |  |
| --- | --- |
|  | untitled |

**NoteOfTheEverything**

Bachelor’s Thesis

Anzel Öztürk 215SE2303

Seray Şimşek 216CS2025

Supervised by

İlknur Karadeniz

June 2021

Abstract

In these days, people having trouble to find enough resources for studying. Everyone wants to get access to other lecture notes freely because only one resource doesn’t allow people to

have enough knowledge about the subjects. Another problem for users is spending so much time finding the correct notes and having easy access to these notes.

What NoteOfTheEverything offers is that anyone can reach any lecture note without a payment and they can also share their own with others. Also, if any user needs help with something there is a Neural Network based chatbot “NoteBot” which can answer their questions immediately. For additional help to the user, the system shows the uploaded document’s file type and if the user wants to review these notes later, the user can add these notes to the ‘My Favorite Note’ section on the user menu. For showing the most related and best note uploaded, the system shows the most rated notes first then the others, by that way no additional time will be spent finding the correct notes.

Acknowledgements

*To my family and loved ones;*

Thanks for supporting me through all the steps that I made in my life. For enduring all my stress and frustration during my university period, for never leaving me alone, making me smile even in the hardest times and most importantly telling me that “You can”, even when I thought I could not do it.

Also, I want to thank my coworker Seray Şimşek for helping each other through all the thesis and I want to thank to our supervisor Asst. Prof. İlknur Karadeniz for supporting us during our thesis project.

Anzel ÖZTÜRK

I would like to thank our supervisor Asst. Prof. İlknur Karadeniz who guided us during our thesis work, my dear friend and coworker Anzel Öztürk, who worked patiently with me throughout this process, and my loving family who have shown me their love, support and always believed in me throughout my life.

Seray ŞİMŞEK

**Table of Contents**

[**Chapter 1** 1](#_Toc73653667)

[1. INTRODUCTION 1](#_Toc73653668)

[*1.1. Problem Definition 1*](#_Toc73653669)

[*1.1.1. Getting help 1*](#_Toc73653670)

[*1.1.2. Finding the ‘best’ 1*](#_Toc73653670)

[*1.2. Purpose of the system 1*](#_Toc73653671)

[*1.3. Scope of the system 2*](#_Toc73653671)

[*1.4. Objectives and Success Criteria of the Project 2*](#_Toc73653671)

[*1.5. Definitions, Acronyms and Abbreviations 2*](#_Toc73653671)

[*1.6. Glossary 2*](#_Toc73653671)

[*1.7. Organization of the Thesis 3*](#_Toc73653671)

[**Chapter 2** 4](#_Toc73653672)

[2.LITERATURE REVIEW 4](#_Toc73653673)

[**Chapter 3** 5](#_Toc73653674)

[3.PROPOSED SYSTEM 5](#_Toc73653675)

[*3.1. Introduction 5*](#_Toc73653676)

[*3.2. Functional Requirements 5*](#_Toc73653676)

[*3.3. Nonfunctional Requirements 6*](#_Toc73653676)

[*3.4. Overview of the NoteOfTheEverything Architecture 7*](#_Toc73653676)

[*3.4.1. System Decomposition 7*](#_Toc73653670)

[*3.5. Hardware Software Mapping 8*](#_Toc73653676)

[*3.6. Persistent Data Management 9*](#_Toc73653676)

[*3.7. Access and Control Security 10*](#_Toc73653676)

[*3.8. Global Software Control 11*](#_Toc73653676)

[*3.9. Boundry Conditions 12*](#_Toc73653676)

[**Chapter 4** 13](#_Toc73653677)

[4.IMPLEMENTATION,TESTS AND EXPERIMENTS 13](#_Toc73653678)

[*4.1 Implementation 13*](#_Toc73653679)

[*4.2 Tests 15*](#_Toc73653680)

[*4.3 Experiments 16*](#_Toc73653681)

[**Chapter 5** 18](#_Toc73653682)

[5.CONCLUSIONS AND FUTURE WORK 18](#_Toc73653683)

[**Chapter 6** 19](#_Toc73653684)

[6.APPENDIX 19](#_Toc73653685)

[6.1.System Models 19](#_Toc73653686)

[*6.1.1 Use case Model 15*](#_Toc73653680)

[*6.1.2 Object Model 32*](#_Toc73653680)

[*6.1.3 Dynamic Models 32*](#_Toc73653680)

[**Chapter 7** 39](#_Toc73653687)

[7.REFERENCES 39](#_Toc73653688)

**LIST OF FIGURES**

[Figure 1. System Decomposition of NoteOfTheEverything 8](#_1t3h5sf)

[Figure 2. Hardware-Software Mapping 9](#_2s8eyo1)

[Figure 3. Visualization of Data Management 10](#_17dp8vu)

[Figure 4. Access Control and Security Table 11](#_26in1rg)

[Figure 5. HTTP request- response diagram 11](#_1ksv4uv)

[Figure 6. Neural network 2 hidden layer 14](#_44sinio)

[Figure 7. ReLU Function 14](#_2jxsxqh)

[Figure 8. Example of Train.py output 15](#_z337ya)

[Figure 9. Transition time 15](#_1y810tw)

[Figure 10. Note sharing time 15](#_49x2ik5)

[Figure 11. Note download time 16](#_49x2ik5)

[Figure 12. Communication time with Chatbot 16](#_49x2ik5)

[Figure 13. Tag, Pattern and Responses example 17](#_49x2ik5)

[Figure 14. Chatbot’s learning experiment 17](#_49x2ik5)

[Figure 15. Use case Model 31](#_49x2ik5)

[Figure 16. Object Model 32](#_49x2ik5)

[Figure 17. User login 32](#_49x2ik5)

[Figure 18. User Registration 33](#_49x2ik5)

[Figure 19. Logout 33](#_49x2ik5)

[Figure 20. User rating a note 33](#_49x2ik5)

[Figure 21. User using the search box 34](#_49x2ik5)

[Figure 22. Share note 34](#_49x2ik5)

[Figure 23. Forgot password 34](#_49x2ik5)

[Figure 24. Filter option 35](#_49x2ik5)

[Figure 25. Delete notes in the my favorite notes part 35](#_49x2ik5)

[Figure 26. Delete notes in the my notes part 35](#_49x2ik5)

[Figure 27. Comment 36](#_49x2ik5)

[Figure 28. ChatBot(Notebot) 36](#_49x2ik5)

[Figure 29. Change Username 36](#_49x2ik5)

[Figure 30. Change Password 37](#_49x2ik5)

[Figure 31. Change Email 37](#_49x2ik5)

[Figure 32. Add notes to my favorite notes part 37](#_49x2ik5)

[Figure 33. Admin login 38](#_49x2ik5)

[Figure 34.Admin Confirm 38](#_49x2ik5)

# Chapter 1

# 1. INTRODUCTION

With development in technology, our lives have changed mostly on-line. All the resources, information and even the books can be read or downloaded from the internet all over the world. But this technological environment has created some problems. Our main goal is to eliminate one of these problems by helping people to find good and reliable notes for whatever subject they want.

**1.1 Problem Definition**

Whenever an on-line source is wanting to be used, there is always a payment waiting for the user to make, with this technology level we live in, anyone should be able to find and share notes freely and easily.

**1.1.1 Getting help**

Whenever a person faces a problem or tries to find a thing, the traditional way is to find

someone in charge to speak with. This method is a waste of time and energy and needs to

be solved.

**1.1.2. Finding the ‘best’**

By making a little search on the internet everything can be found, but is it the best one?

There should be a way to understand it from the ones who looked at it earlier.

**1.2 Purpose of the system**

‘NoteOfTheEverything’ website created to help those who want to share or find notes over the internet. Main idea is to be accessible to anyone without a payment or another commitment as well as to help all the users within seconds.

**1.3 Scope of the system**

While the Note project is made, it is aimed that all users can use the site easily and there is an admin control page to provide users with safe note content and show related comments. Notes and comments posted by users are first checked by the admin, if accepted and approved by the admin, the notes and comments will be accessible to other users.

In short, the project allows users to access the notes they want easily and at no cost. In addition, NoteOfTheEverything has a NoteBot to provide immediate support to user questions.

**1.4 Objectives and Success Criteria of the Project**

“NoteOfTheEverything” aims to be the first preferred note sharing website by the users from all over the world, to be the one that provides the most accurate resource opportunities to users. Also, while achieving the above goals, it always aims to be user-oriented and make the site easy to understand for every user.

**1.5 Definitions, Acronyms and Abbreviations**

**NoteOfTheEverything:** The name of the developed project.

**NoteBot:** It is the chatbot name on the site that instantly answers user questions on the site.

**DB**: Database

**ReLU**: Rectified Linear Unit

**1.6 Glossary**

This project consists of the 3 main users and a system;

**Registered user:** It is the person who can register to the system and access all notes and comments, share comments and notes, give points to the notes and have the authority to ask questions using NoteBot.

**Guest:** it is only the person whose login and registration page can be accessed.

**Admin:** It is the person who checks the notes and comments uploaded to the system and gives approval for their publication.

**System:** NoteOfTheEverything.

**1.7 Organization of the Thesis**

Chapter 1 starts with the introduction and general information about project purpose, scope and Objectives and our success criteria for the project. Continues with the overview of the chapter. In Chapter 2, Literature review is which shows the current system and used algorithms, frameworks and difference between created and other systems. Chapter 3 is the proposed system part, which explains our system difference and what we offer to detected problems. With Chapter 4, implementation steps, used techniques, learning algorithms and tests done for both website testing for best user experiment and finding the best working range of the ChatBot with its development levels. Lastly, in Chapter 5 the conclusion about the ‘NoteOfTheEverything’ and in future, how this project can be improved and updated for better usage overtime.

# Chapter 2

# 2.LITERATURE REVIEW

Students are the primary users of modern website designs. These websites are valuable to students, but they are also costly to them. Other websites are unable to cover all of a student's demands with lack of number of the notes or to be able to see those notes there is a highly loaded payment is done. For example, “Notedu”.

We looked into the most popular note-sharing websites, and we noticed that these websites are difficult to use. These websites' programs and software are built using the Python programming language, as well as JS, CSS, and HTML.

For their data, they use DB tools like MySQL. The NoteOfTheEverything system provides a web interface for users and is built on a user and admin authentication system. The NoteOfTheEverything system differs from other note-sharing websites in that it features a user-friendly interface. Also, we have a chatbot option for users who are confused and want to ask questions about the site.

# Chapter 3

**3.PROPOSED SYSTEM**

**3.1 Introduction**

As a result of our research, we realized that other note sharing sites are more money-oriented, or we realized that it is necessary to log in with an e-mail address in that institution or school in order to create and access notes for a particular school and institution. While developing this project, we aimed to eliminate these restrictions against users. The NoteOfTheEverything project has a score-based system to enable them to find the note with the best content and the most liked one in the system because the note most liked by other users will appear at the top of the ranking. Also, NoteOfTheEverthing has a chatbot called NoteBot who aims to have user questions answered immediately. In this way, users will not have to wait to get answers to their questions.

**3.2 Functional Requirements**

* Guests must be able to register.
* The registered user must be able to login.
* The registered user must be able to share the lecture notes.
* The registered user must be able to search.
* The registered user must be able to filter.
* The registered user must be able to change their password.
* The registered user must be able to change email.
* The registered user must be able to change username.
* The registered user must be able to add favorite lecture notes to the ‘My Favorites Notes’ part.
* The registered user must be able to score the notes of the other user’s notes.
* The registered user must be able to comment on the notes of the other user’s notes.
* The registered user must be able to delete the notes she/he has uploaded.
* The registered user must be able to delete the notes in the ‘My Favorites Notes’ part.
* The registered user must be able to ask questions to the Chatbot.
* The registered user must be able to logout.
* The registered user must be able to click the forgot password option.
* Admin must be able to login.
* Admin must be able to confirm the sent notes and publish the confirmed notes.

**3.3 Nonfunctional Requirements**

* **Usability**
* The website must be understandable for all users.
* Admins should approve or delete notes on the website easily.
* **Reliability**
* When the registered user searches for lecture notes, the system should be showing the correct lecture notes to the user in an acceptable time.
* Admin responsible for validating notes for reliability.
* **Performance**
* The website is accessed over the internet.
* System is always available and access is guaranteed.
* The system must return to the user within 3 seconds for each operation.
* **Supportability**
* The NoteOfTheEverything note system should be managed by admin. Admin responsible for the approval of the notes for reliability.
* When a problem occurs in the note section, it is immediately fixed by the admin.
* **Implementation**
* The database will be created by using Django’s Database.
* PyCharm will be used as a coding IDE.
* Python will be used.
* **Interface**
* \*There is no such constraint in the project.
* **Packaging**
* Our system is a website; you do not need to download anything. System is reachable through the server.
* **Legal**
* The content of this project is protected by copyright law.

**3.4 Overview of the NoteOfTheEverything Architecture**

NoteOfTheEverything is a note sharing site that can be used by anyone who needs lecture notes. In this system, there are 3 main users as Admin, Guest and registered user. Guests can only access the registration and login page. The registered user can access other notes, share notes, my notes, my favorite notes, NoteBot, comment and score, update personal information, search and filter sections. And the admin is the user who approves the publication of the notes and comments submitted by the users, waiting to be published.

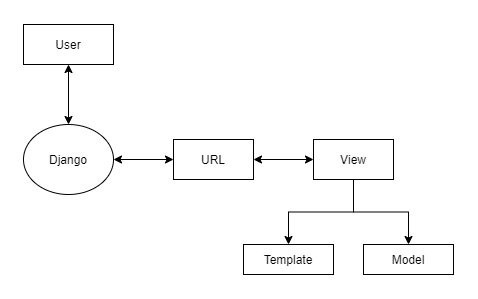
Because NoteOfTheEverything is a web application, we selected to use the Django framework, which is a high-level language that also uses open sources. HTML5, JavaScript, and CSS were utilized to create the user interface. As a server, we used Python HTTP, which merged databases and frontends. Because our system is a web application based on Django, we chose MVT as an architecture. First and foremost, the model layer is in charge of dealing with data and allowing users to make changes to their information, such as updating or inserting it.

This user data is transmitted to the controller, which then passes it on to the model layer, which stores it in the database. The view layer, which includes the user interface side and frontend such as HTML, CSS, and so on, is the second layer. Through this layer, we may see the data that is stored in the model layer. The template layer is the final layer, and it contains static parts of the HTML and the process user interface.

**3.4.1 System Decomposition**

In order to access all functions of the website, users must be logged in to the system. The login subsystem is required to verify the user, according to this information the user can access to the system. When the users log into the system, they first see the note list page, the search box and the filtering feature. The registered user's information is stored in the NoteOfTheEverything subsystem, so users can enter their own shared notes, always easily access the notes they have added to their favorites, and easily update their account information. There is an admin page in the system, the admin can approve or delete pending notes and comments.

NoteOfTheEverything is based on MVT architecture and Django framework.



**Figure 1. System Decomposition of NoteOfTheEverything**

* **Django**

Django is a software that controls the actions between Model and view sections.

* **Model**

Operations and database are done in the “Model” part.

* **View**

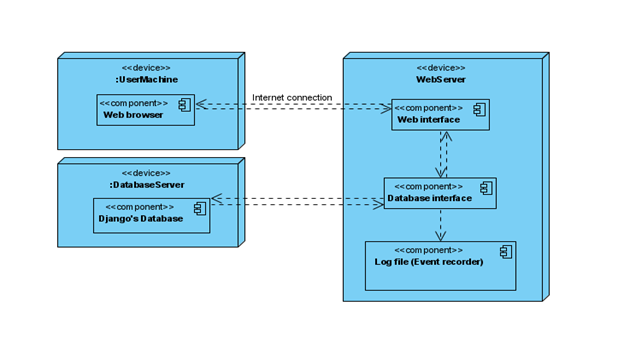
The written python codes are found in the “View” part.

* **Template**

Manages the pages we design for the user from the “Template” part.

**3.5 Hardware Software Mapping**

The general connecting principles for our system are represented in the hardware diagram. The user's machine connects to the system over the internet, and once connected, the web interface communicates with the database interface. The database interface communicates with the database server, and all occurrences in the operating system are logged in the log file for further review.

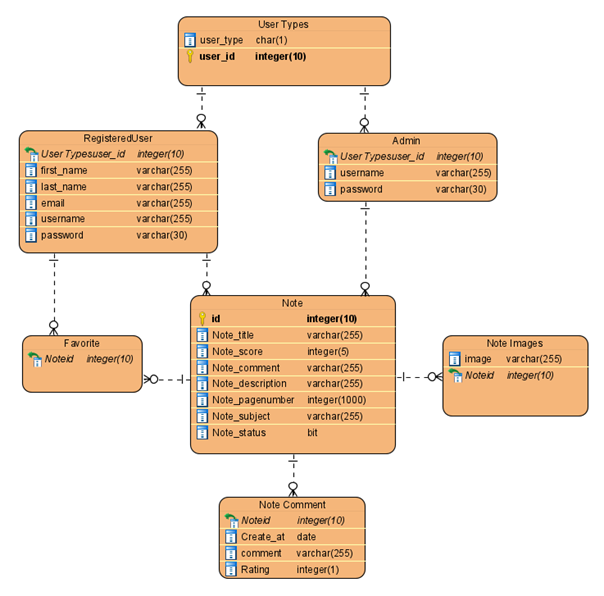


**Figure 2. Hardware-Software Mapping**

**3.6 Persistent Data Management**

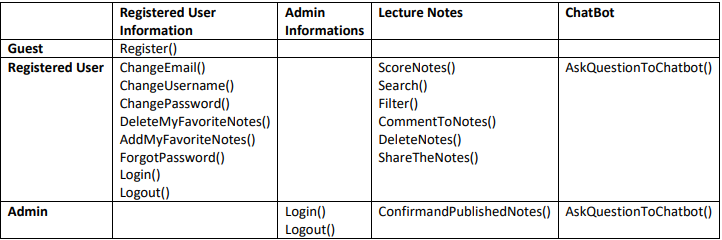
We detail our system's data management in the section on persistent data management. Our website is built on the Django framework, which supports SQLite databases. As a result, SQLite will be used to develop the relational database for the NoteOfTheEverything system. We assume that SQLite is the best option because it is simple to set up and utilize. SQLite also provides the feature of creating a database as a single file, which makes our work a lot easier. In terms of databases, Django is a suitable framework since it automatically creates bindings between permanent connections and databases. Django also adds authentication techniques automatically when accessing databases. Static data, like photos and CSS files, were not stored in the database by Django. Django puts these files in a static folder rather than keeping them in the database.

The NoteOfTheEverything's relational database schema is as follows:



**Figure 3. Visualization of Data Management**

**3.7 Access and Control Security**

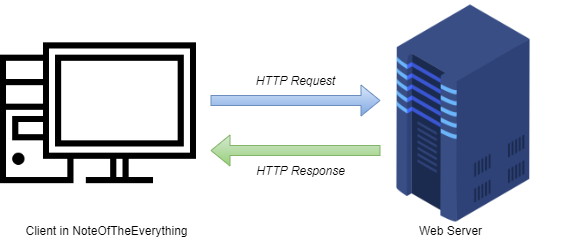
We shall use the greatest feasible security mechanism in the NoteOfTheEverything because people's privacy is the most ethical issue these days. When a user logs in to the system, the system will check to see whether the user is real, and if the verification email procedure has been completed, the system will ask for password that the user has established, which must include at least one upper case letter, one number, and one symbol. The system is built on the Django framework, which increases security by using its own encryption type. This encryption is performed using the PBKDF2 method with a SHA256 hash, and the system also includes a NIST based security password extension mechanism. Because NoteOfTheEverything is a website where users' actions have an effect on the system, Django includes an automatic clickjacking defense to prevent probable system attacks. The system will be safeguarded by these precautions and the fact that only administrators have access to the control systems.

**Figure 4. Access Control and Security Table**

**3.8 Global Software Control**

The global software control of our NoteOfTheEverything Website is described in this part. This website is accessed through the internet and is viewed using web browsers such as Google Chrome, Internet Explorer, Firefox, and others. An internet connection and a web browser are required for users to access our site.

The control flow in Django is event-driven. When a user sends a request to the database, the system answers and shows the information requested. When a user clicks on a register part, which clicks are events, our website responds by opening a register page to the user.

****

**Figure 5. HTTP request- response diagram**

**3.9 Boundary Conditions**

**Initialization:** Visitors, registered users, and admin are the three types of users in the NoteOfTheEverything system. Visitors can only use a browser and an internet connection to access the website and register. Users who have created an account in the system can log in and use the website's capabilities using a browser and an internet connection. Users who have registered can search the system or submit notes to it. Admins can access the website using their computer's browser and internet connection. Admin can confirm notes and choose whether or not to post them on the website. To authenticate, all users, different from visitors, must fill in the blanks with their username and password. Users must input valid information to log in to our website; otherwise, the user will not be able to log in and will receive an error message.

**Termination:** The website NoteOfTheEverything provides a logout button to leave the system. When a user clicks the logout button, the system logs them out of their account and terminates their session. The user remains signed in as long as he or she does not press the logout button.

**Failure:** Before logging out of the NoteOfTheEverything website, users must save any changes they made, such as changing their profile information or adding a note to the system. In the event of a connection failure, the website only saves data that has been saved by users. Before the user signs out, the website delivers a notice to the user if there is a problem with unsaved modifications.

# Chapter 4

# 4.Implementation, tests and experiments

## 4.1 Implementation

As NoteOfTheEverything project team, first considered the user needs and decided to developed our project according to these requirements. While developing this project, we added these requirements as RegisteredUser, Note, NoteImages, UserType, Comment and PendingComment tables in our project's models.py file using the SQLite database supported by Django.

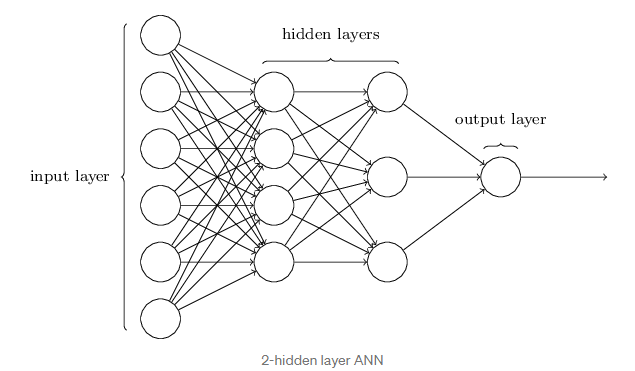
The RegisteredUser table holds the id and information about the registered user. Note table keeps the information of the notes published in the system. NoteImage table keeps the note images if the image of the uploaded lecture notes is added by the user. The UserType table helps us to understand whether the user is an admin or a registered user. The Comment table keeps the comments written under the note, the score given to the note, the creation time and day of the comment. PendingComments table is the table where the comments waiting for approval to be publish by the admin.

In the front-end design of the NoteOfTheEverything project; we organize static content files such as CSS and Bootstrap under the static file on “main” and used python language in other parts of the project. While developing the project, we use Django forms to receive user inputs and implement our project in this way.

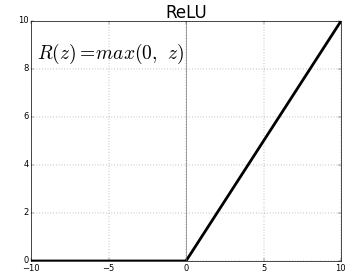
There are many data operations that the user can do in our project, these are; deleting, adding, updating, filtering the notes…

We develop our project with a user-oriented approach, we use the "order\_by" method so that the user can see the note with a higher score in the first place. In this way, the user will be able to access more efficient notes more easily.

In the NoteOftheEverything Project, first importance is to solve users' problems immediately and to develop a chatbot. We made changes on a chatbot implementation that we deemed appropriate for our project and integrated it into our project. We use the nltk library while developing the chatbot. The chatbot has the ability to learn using the Neural Network with 2 hidden layers, and with its ReLU function, it ensures that the active neurons in the network have an efficient computational load.



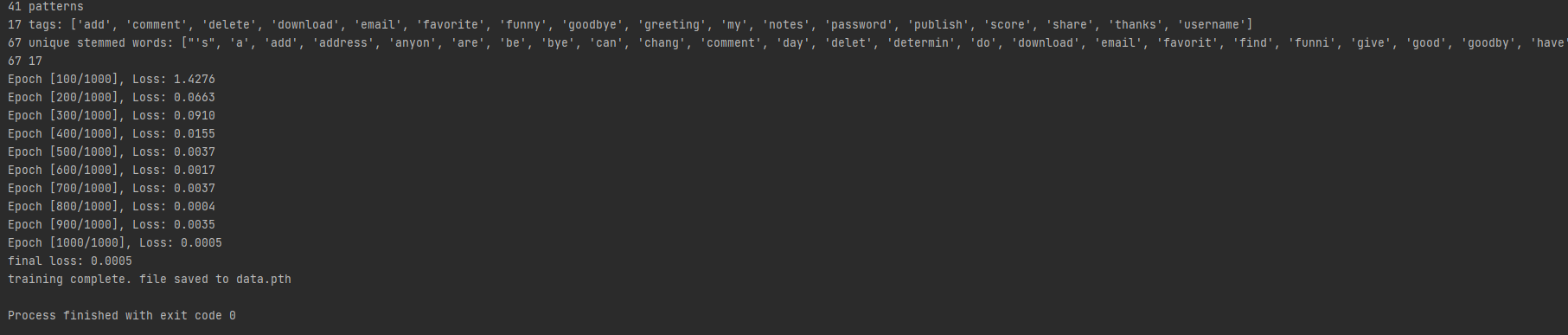
**Figure 6. Neural network 2 hidden layer**

****

**Figure 7. ReLU Function**

The ReLU function is an enable function for neural networks, because this function makes it easier to train a model and provides better results and performance.

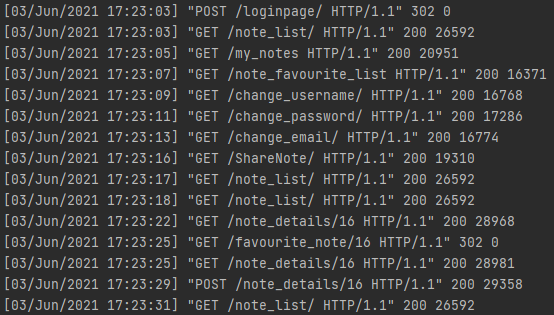
In order for the chatbot to learn, we created a tag (unique name), pattern and response examples in the intents.json file in our project and trained our chatbot with this data. The algorithm used in the chatbot performs its learning by first separating the patterns in the data set into tags and then stems. According to the input entered by the user, it returns a response by looking at the stems it has learned.



**Figure 8. Example of Train.py output**

## 4.2 Tests

1. Our first test; Showing the elapsed time between all pages on the server side. Each row in the figure represents a new request. The time between the pages is at most 3 second time units. This time may vary depending on internet speed.



**Figure 9. Transition time**

1. Users upload a new note;

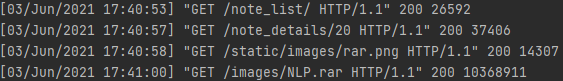
* The user clicks the share note button on the page.
* Enters the information of the note to be loaded.
* Clicks the send button.



**Figure 10. Note sharing time**

1. Time taken while the user downloaded the lecture note;

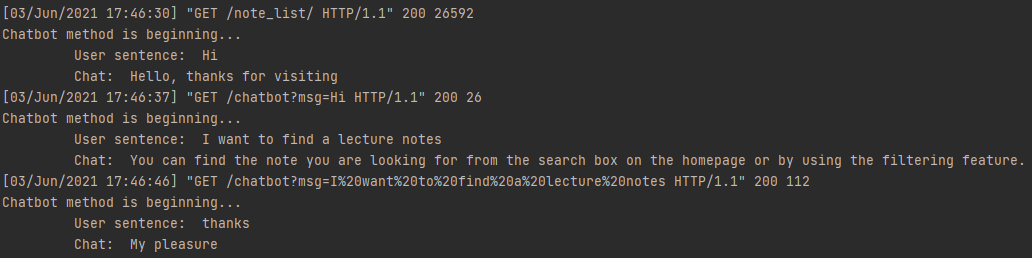
* The user enters the details page of a note selected from the note list page.
* The system shows the file type of the note to the user as the user enters the page.
* Then the user clicks the button to download the note.



**Figure 11. Note download time**

1. The average duration of the correspondence between the chatbot and the user;

* The user presses the chatbot icon at the bottom right of each page and writes what they want to ask.
* Chatbot answers the question instantly.

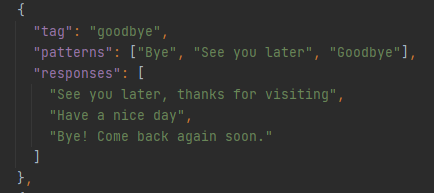


**Figure 12. Communication time with Chatbot**

## 4.3 Experiments

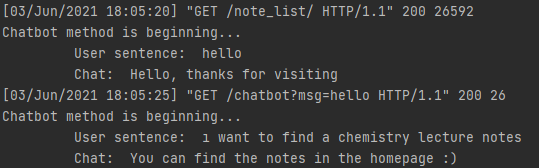
In the project, we used the chatbot implementation that we specified as a reference in our research for the chatbot. The chatbot we have integrated into the NoteOfTheEverything project is learning with the Neural Network (2 hidden layer) algorithm. We created the tag, pattern and response parts using the intents.json file in our project, then we ran the train.py file to teach this information. In this way, the chatbot algorithm completes the learning process by separating the specified patterns into tags and then separating them into stems.

Tag, pattern and response examples are given in the figure below.



**Figure 13. Tag, Pattern and Responses example**

Afterwards, we asked a question that is not found in the chatbot's intents.json file to test whether it works well. The part about the notes in the Intents.json file is given below;



**Figure 14. Chatbot’s learning experiment**

We have tested that the chatbot gives the correct answer to the question asked by the user and learns by doing an experiment in this way.

# CHAPTER 5

# 5.CONCLUSIONS AND FUTURE WORK

We set certain criteria to be successful while doing this project;

* Creating a website that can meet all the requirements of users in this matter
* Improve the functions in our system
* Develop a system with more secure, reliable notes, and where users can get results faster

We accomplished the criteria that we want to achieve, which are;

* We created and designed a website that can fulfil the desired features from a note sharing site.
* We improved and coordinated some features to our system such as adding notes to the favorite notes part.
* On our website, users can access more secure and reliable notes because after the notes are uploaded, they are sent to the admin's approval and uploaded to the system after receiving their approval. Our system provides the user with faster access and easy use.

Our website’s main requirements:

* Searching notes in the website
* Filtering notes
* Adding or deleting notes
* Adding or deleting notes in favorites
* Post comments and scores on notes
* Using chatbot

In the future work of the NoteOfTheEverything website, users can edit their own notes. The chatbot can be developed further, for example, the dataset of the chatbot can be increased. Furthermore, administrative jurisdiction may be increased.

# Chapter 6

# 6.Appendix

# 6.1.System Models

**6.1.1. Use Case Model**

**Use case name** Register

**Participating Actors** Initiated by Guest

**Flow of Events**

1. NoteOfTheEverything homepage shows register button and

login page link.

2. Guest clicks the register button on the homepage to have an

account in the site.

3. NoteOfTheEverything displays a blank form for registration

and this form includes user’s information such as full name, email address and password.

4. Guest fills this fields on the form with his/her information and

then clicks the “Register” button.

5. NoteOfTheEverything responds to the Guest by displaying a

pop-up message which says “Your account is created successfully.” and then directs the User

to the login page of the NoteOfTheEverything.

**Entry condition -** Guest clicks the register button of the NoteOfTheEverything.

**Exit condition** - Guest notices the “Your account is created successfully.”

message of the system.

**Quality requirements** NoteOfTheEverything must deliver the new user to the database in 3 seconds.

**Use case name** Login

**Participating Actors** Initiated by Registered User

**Flow of Events** 1. Registered User fills the blank form with her/his email address

and password in the Login page. After that, he/she clicks the “Login” button to open his/her

account.

2. NoteOfTheEverything opens users’ account and shows the

user’s username in the upper right corner of the homepage.

**Entry condition -**Registered User enters the login page of the NoteOfTheEverything.

**Exit condition** - Registered User enters his/her account in the site and located on the his/her homepage.

**Quality requirements -** NoteOfTheEverything must not show a given password on screen.

**Use case name** ShareLectureNotes

**Participating Actors** Initiated by Registered User

**Flow of Events** 1. Registered User adds their lecture notes to the “MyNotes” section and then clicks the “Share” button.

2.NoteOfTheEverything responds to the Registered User by displaying a message which says “Your note has been sent to admin approval for publication.”.

**Entry condition -** Registered User is logged in to NoteOfTheEverything with his/her username and password.

**Exit condition** - Registered User notices a message of the NoteOfTheEverything

as a response which says “Approval is pending”.

**Quality requirements -**NoteOfTheEverything should keep the user notes safe.

**Use case name** SearchSomething

**Participating Actors** Initiated by Registered User

**Flow of Events** 1. Registered User writes the note they want to find in the search box and then clicks the “Search” button.

2.NoteOfTheEverything responds to the Registered User by showing appropriate notes to the user’s search.

**Entry condition -**Registered User is logged in to NoteOfTheEverything with his/her username and password.

**Exit condition** -Registered User notices a message of the NoteOfTheEverything

as a response which says “Suitable notes have been found”.

**Quality requirements -**NoteOfTheEverything must deliver the search results to the system in 2 seconds.

**Use case name** Filter

**Participating Actors** Initiated by Registered User

**Flow of Events** 1. Registered User choose features of the notes they want to find in the filter boxes and then click the “Find” button.

2.NoteOfTheEverything responds to the Registered User by showing appropriate notes to the user’s search.

**Entry condition -**Registered User is logged in to NoteOfTheEverything with his/her username and password.

**Exit condition** - Registered User notices a message of the NoteOfTheEverything

as a response which says “Appropriate notes found for filtering”.

**Quality requirements -**NoteOfTheEverything must deliver the search results to the system in 2 seconds.

**Use case name** ChangePassword

**Participating Actors** Initiated by Registered User

**Flow of Events** 1.Registered user enters the NoteOfTheEverything website and then clicks the Change Password button under the My Account part.

2.NoteOftheEverything shows a blank form in response to the

user.

3.The user enters the old password once and the new password

twice in the blank form.

4.NoteOftheEverything shows the message "Password change

process completed successfully" to the user.

**Entry condition -** Registered User is logged in to the NoteOfTheEverything with

his/her username and password.

**Exit condition -**Registered User notices a message of the NoteOfTheEverything

as a response which says “Password change process completed successfully”.

**Quality requirements -**NoteOfTheEverything must encrypt passwords for security.

**Use case name** ChangeE-Mail

**Participating Actors** Initiated by Registered User

**Flow of Events** 1. Registered user enters the NoteOfTheEverything website and

then clicks the Change Email button under the My Account part.

2. NoteOftheEverything shows a blank form in response to the

user.

3. The user enters the new email address and verification code

in her/his email address in the blank form.

4.NoteOftheEverything shows the message "Email change process completed successfully" to the user.

**Entry condition -** Registered User is logged in to NoteOfTheEverything with

his/her username and password.

**Exit condition -** Registered User notices a message of the NoteOfTheEverything

as a response which says “Email change process completed successfully”.

**Quality requirements** User should understand the interface of the page.

**Use case name** ChangeUserName

**Participating Actors** Initiated by Registered User

**Flow of Events** 1. Registered user enters the NoteOfTheEverything website and

then clicks the Change Username button under the My Account part.

2. NoteOftheEverything shows a blank form in response to the

user.

3. The user enters the new username once in the blank form.

4. NoteOftheEverything shows the message "Username change process completed successfully" to the user.

**Entry condition -** Registered User is logged in to NoteOfTheEverything with

his/her username and password.

**Exit condition -** Registered User notices a message of the NoteOfTheEverything

as a response which says “Username change process completed successfully”.

**Quality requirements -**User should easily understand the interface of the page.

**Use case name** FavoriteLectures

**Participating Actors** Initiated by Registered User

**Flow of Events** 1.Registered user enters the NoteOfTheEverything website andthen

clicks the “My Favorite Lectures” button.

2. NoteOfTheEverything responds to the Registered User by displaying

all Notes that are his/her favorite ones.

3. Registered User, select one of them to see inside of the note. 4. NoteOfTheEverything responds to the Registered User by opens the

selected note.

**Entry condition -** Registered User is logged in to NoteOfTheEverything with

his/her username and password.

**Exit condition** - Registered User notices a message of the NoteOfTheEverything

as a response which says “Selected note is opening.”.

**Quality requirements** User should easily understand the interface of the page.

**Use case name** GiveScore

**Participating Actors** Initiated by Registered User

**Flow of Events** 1. Registered User enters the selected note’s page.

2. NoteOfTheEverything responds to the Registered User by

displaying notes that he\she selected.

3. Registered User gives a score out of 5 to the note that he/she

read.

**Entry condition -** Registered User is logged in to NoteOfTheEverything with

his/her username and password.

**Exit condition** -Registered User notices a message of the NoteOfTheEverything

as a response which says “Thanks for the score.”.

**Quality requirements** User should easily understand the interface of the page.

**Use case name** ToComment

**Participating Actors** Initiated by Registered User

**Flow of Events** 1. Registered User enters the note’s page that he/she wants.

2. NoteOfTheEverything responds to the Registered User by

displaying notes that he\she selected.

3. Registered User comments the note that he/she read.

**Entry condition -** Registered User is logged in to NoteOfTheEverything with

his/her username and password.

**Exit condition** - Registered User notices a message of the NoteOfTheEverything

as a response which says “Thanks for the comment.”.

**Quality requirements -** User should easily understand the interface of the page.

**Use case name** DeleteNote

**Participating Actors** Initiated by Registered User

**Flow of Events** 1. Registered User opens the “MyNotes” section and chooses the note that he/she wants to delete and clicks the “Delete” button.

2.NoteOfTheEverything responds to the Registered User by displaying a message which says “Do you want to delete Note1?”.

3.Registered User selects yes button.

4. NoteOfTheEverything responds to the Registered User by displaying a message which says “Deleted Successfully”.

**Entry condition -** Registered User is logged in to NoteOfTheEverything with his/her username and password.

**Exit condition** - Registered User notices a message of the NoteOfTheEverything as a

response which says “Deleted Successfully”.

**Quality requirements -**NoteOfTheEverything must delete the note in the system in a short time.

**Use case name** DeleteMyFavoriteNote

**Participating Actors** Initiated by Registered User

**Flow of Events** 1. Registered User opens the “MyFavoriteNotes” section and chooses the note that he/she wants to delete and clicks the “Delete” button.

2.NoteOfTheEverything responds to the Registered User by displaying a message which says “Do you want to delete note?”.

3.Registered User selects yes button.

4. NoteOfTheEverything responds to the Registered User by displaying a message which says “Deleted Successfully”.

**Entry condition -** Registered User is logged in to NoteOfTheEverything with his/her username and password.

**Exit condition** - Registered User notices a message of the NoteOfTheEverything as a

response which says “Deleted Successfully”.

**Quality requirements -**NoteOfTheEverything must delete the note in the system in a short time.

**Use case name** AskToChatbot

**Participating Actors** Initiated by Registered User

**Flow of Events** 1. Registered User opens the chatbot.

2.NoteOfTheEverything responds by opening the chatbot and displays “Hello RegisteredUser!”.

3.Registered User writes questions about the site.

4.Chatbot answers his\her questions.

**Entry condition -** Registered User is logged in to NoteOfTheEverything with his/her username and password.

**Exit condition** - Registered User clicks the close button of the chatbot.

**Quality requirements -**NoteOfTheEverything must deliver the answers to the system in a short time.

**Use case name** Logout

**Participating Actors** Initiated by Registered User

**Flow of Events** 1.Registered User enters the “Logout” section of the MyAccount

part in the NoteOfTheEverything.

2. NoteOfTheEverything responds to the Registered User by redirecting a “login” page.

**Entry condition -** Registered User is logged in to the NoteOfTheEverything with

his/her username and password.

**Exit condition -** Registered User sees the NoteOfTheEverything as a response

which redirects to the “Login” page.

**Quality requirements -**Registered user logged out in 2 seconds.

**Use case name** ForgotPassword

**Participating Actors** Initiated by Registered User

**Flow of Event** 1.Registered user enters the login page of the

NoteOfTheEverything website and then clicks the “Forgot Password” button under the login

form part.

2. NoteOfTheEverything responds to the Registered User

by sending an email.

3. Registered User, clicks the link inside of the email and

enters a new password 2 times and then clicks the “Save” button.

4. NoteOfTheEverything responds to the Registered User by

displaying a message which says “Your new password saved successfully”.

**Entry condition -** Registered User is logged in to the NoteOfTheEverything with his/her

username and password.

**Exit condition -** Registered User notices a message of the NoteOfTheEverything as a

response which says “Your new password saved successfully”.

**Quality requirements** NoteOfTheEverything must encrypt passwords for security.

**Use case name** AdminLogin

**Participating Actors** Initiated by Admin

**Flow of Events** 1. Admin fills the blank spaces with her/his email address and

password. After that, he/she clicks the “Login” button.

2. NoteOfTheEverything responds to the user by showing the

admin’s username in the upper right corner of the homepage.

**Entry condition -**Admin enters the login page of the NoteOfTheEverything.

**Exit condition** - Admin receives the home page of the NoteOfTheEverything and

notices his/her name on the upper right corner of the page.

**Quality requirements -** NoteOfTheEverything must not show a given password on screen.

**Use case name** ConfirmNotes

**Participating Actors** Initiated by Admin

**Flow of Events** 1. Admin opens the incoming notes.

2.NoteOfTheEverything responds by opening the notes.

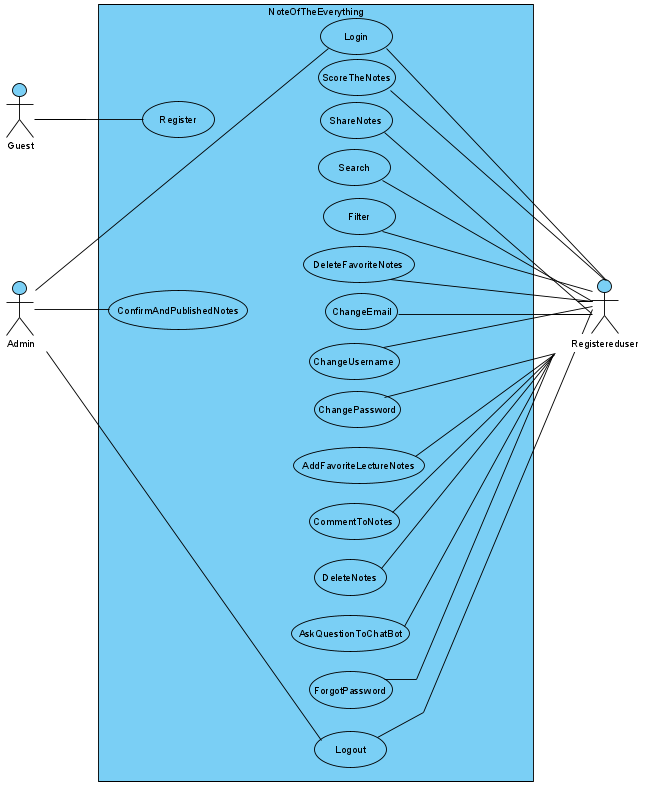
3.Admin checks notes and gives approval to publish on the site.

4. NoteOfTheEverything responds to the Admin by displaying a message which says “Publishing successfully…”.

**Entry condition -** Admin is logged in to the NoteOfTheEverything with his/her username and password.

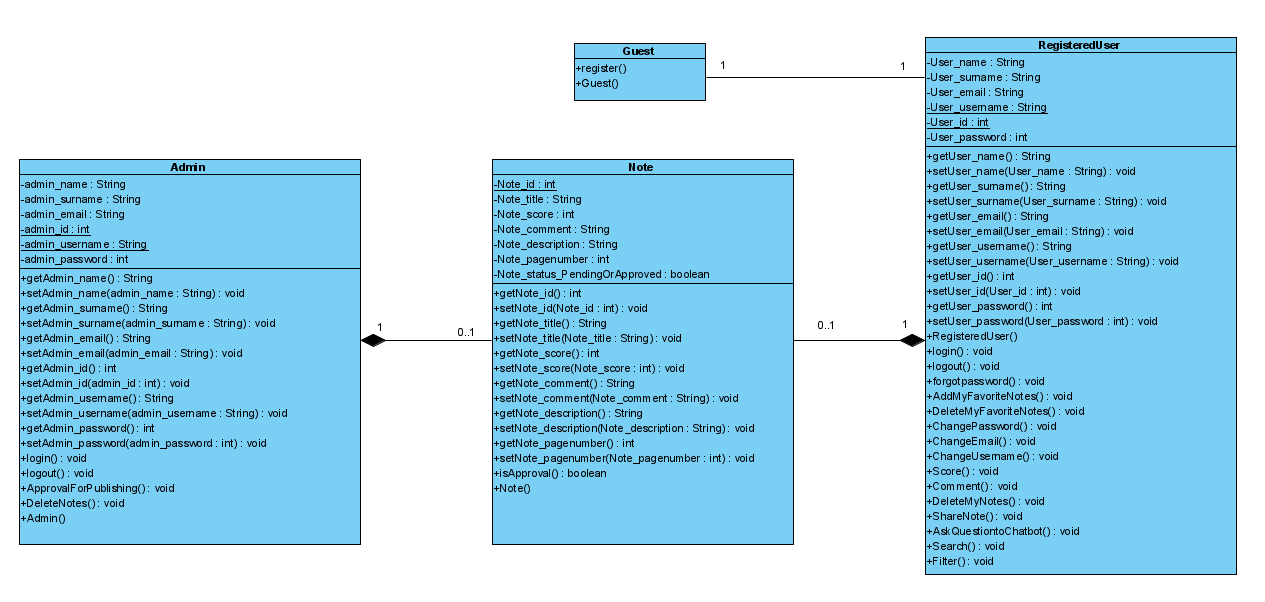
**Exit condition** - Admin clicks on the close button.

**Quality requirements -**NoteOfTheEverything must deliver the notes to the system in a short time.



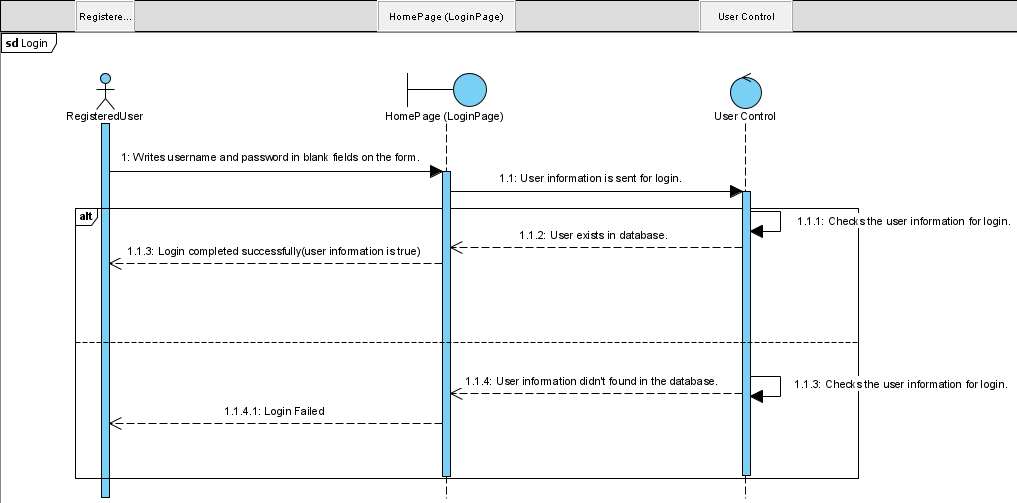
**Figure 15. Use case Model**

**6.1.2. Object Model**

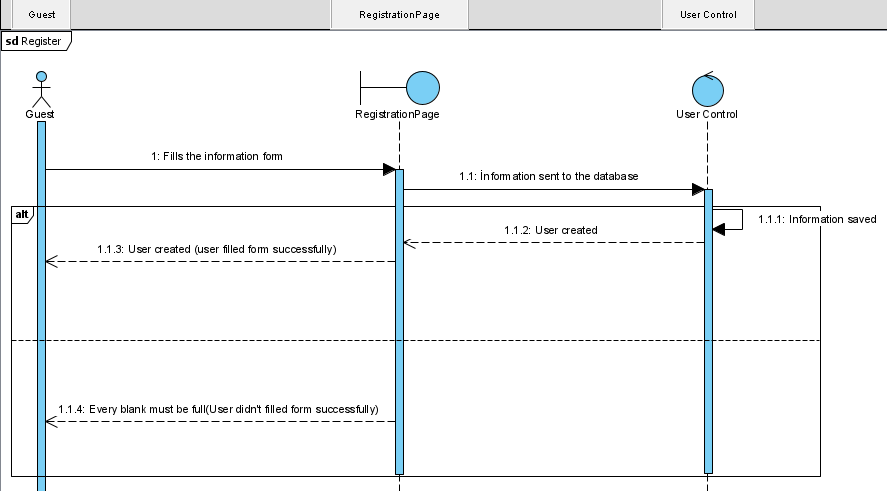


**Figure 16. Object Model**

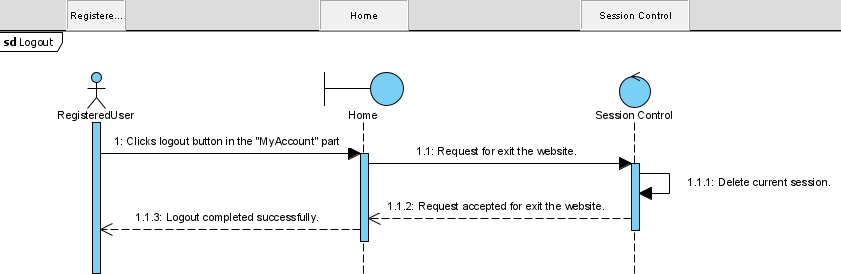
**6.1.3. Dynamic Model**

****

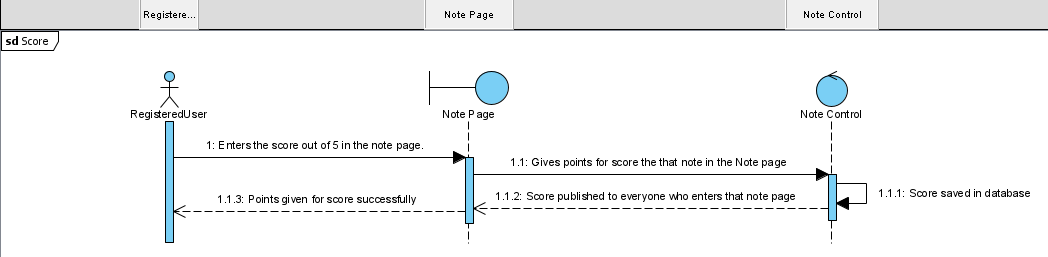
**Figure 17. User login**

****

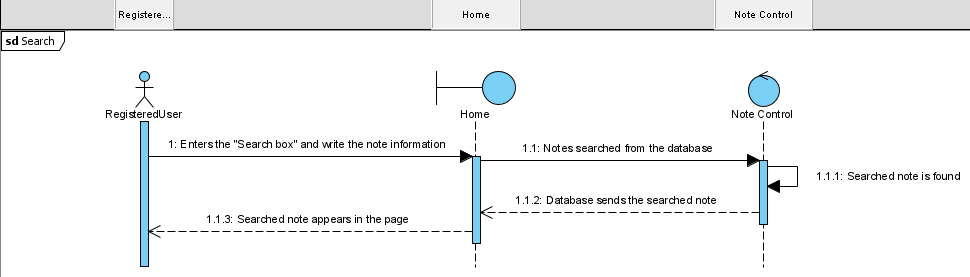
**Figure 18. User Registration**

******

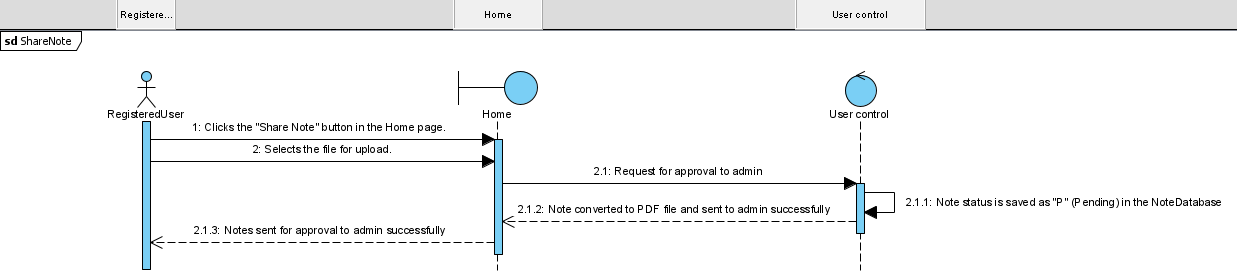
**Figure 19. Logout**

******

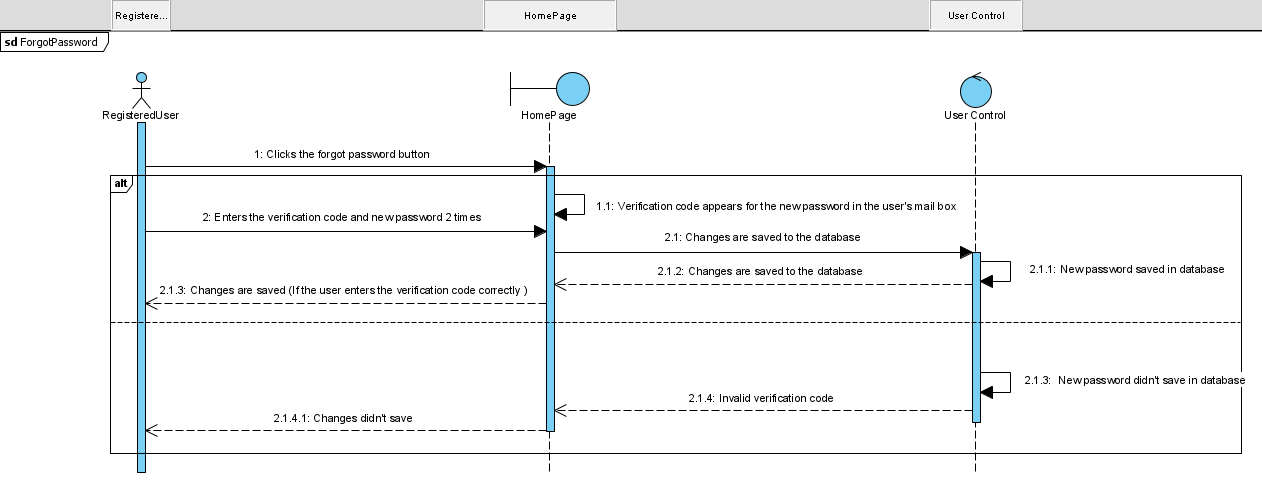
**Figure 20. User rating a note**

******

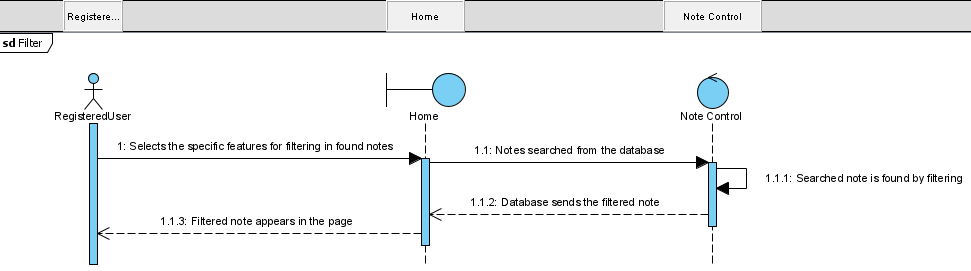
**Figure 21. User using the search box**

******

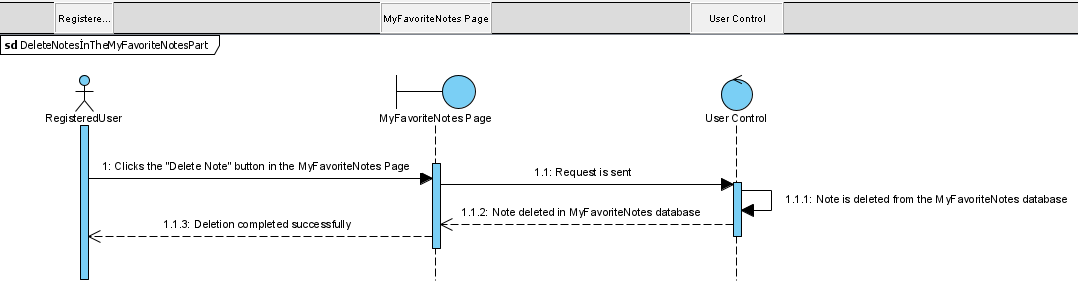
**Figure 22. Share note**

******

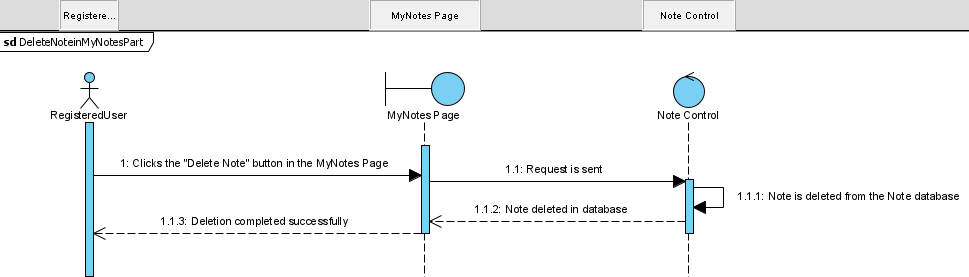
**Figure 23. Forgot password**

******

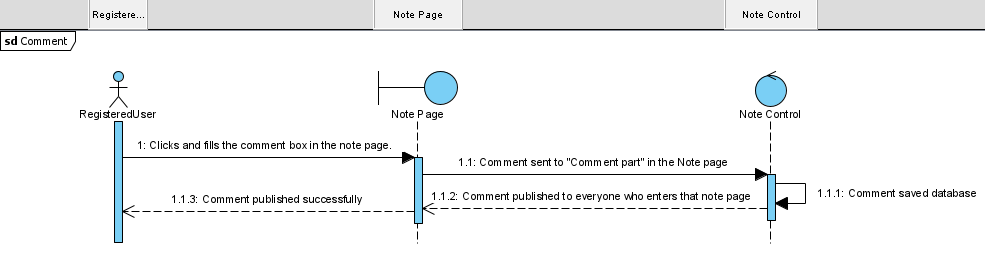
**Figure 24. Filter option**

******

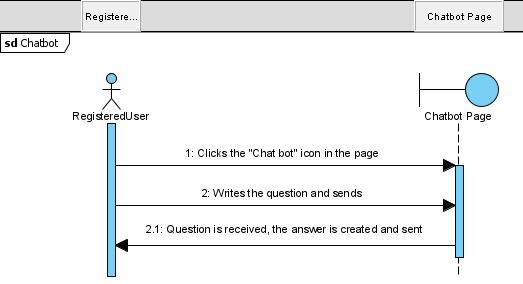
**Figure 25. Delete notes in the my favorite notes part**

******

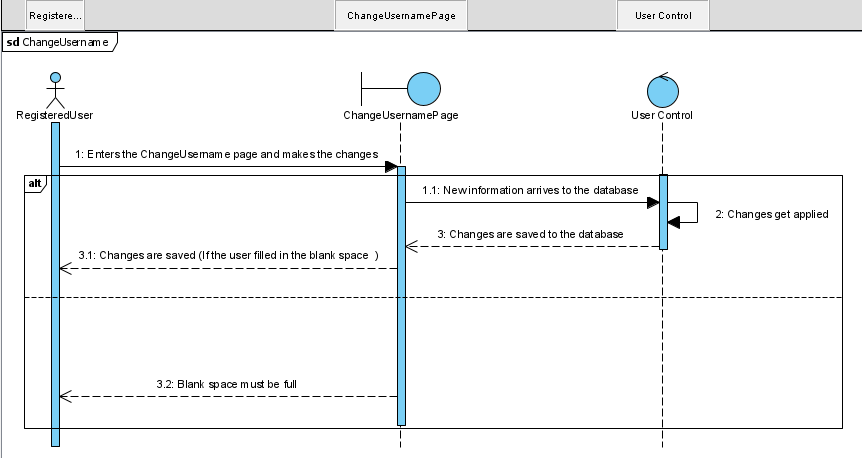
**Figure 26. Delete notes in the my notes part**

******

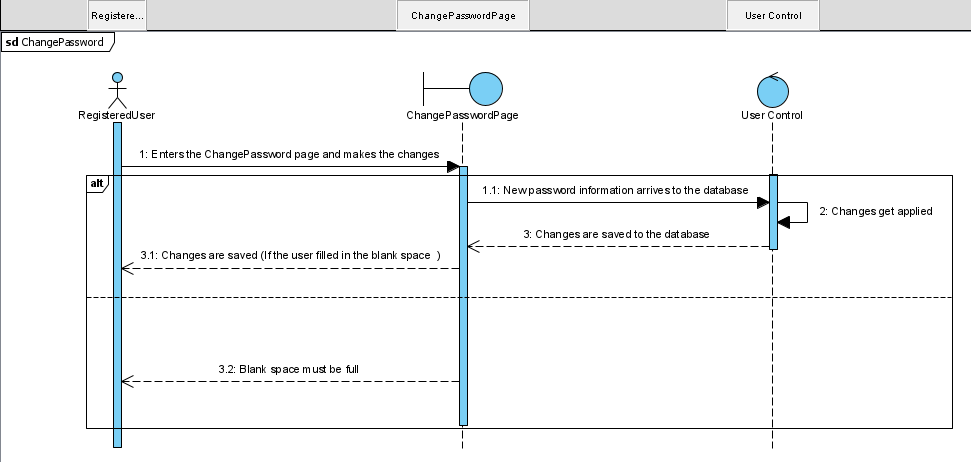
**Figure 27. Comment**

******

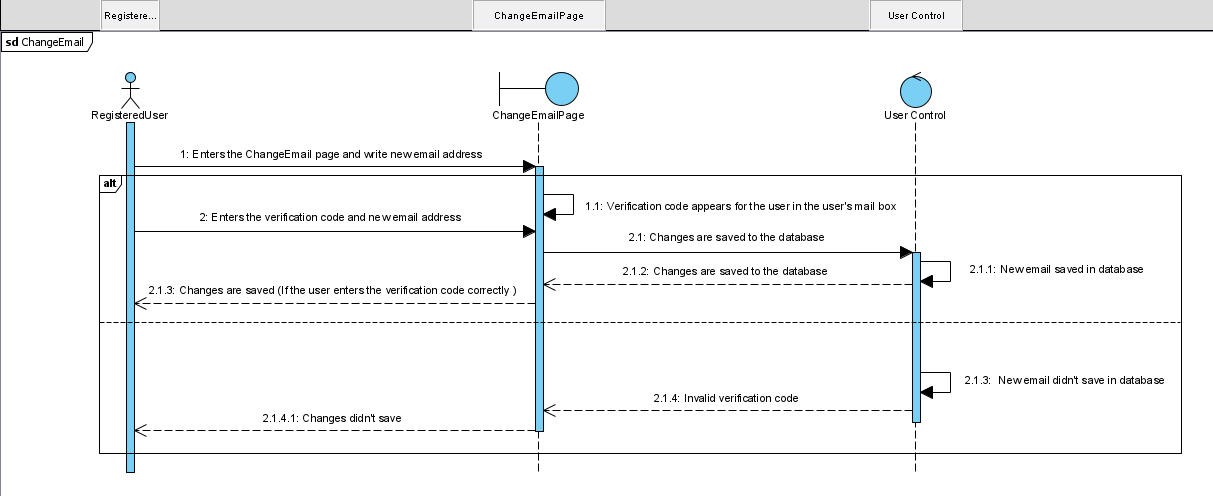
**Figure 28. ChatBot(Notebot)**

******

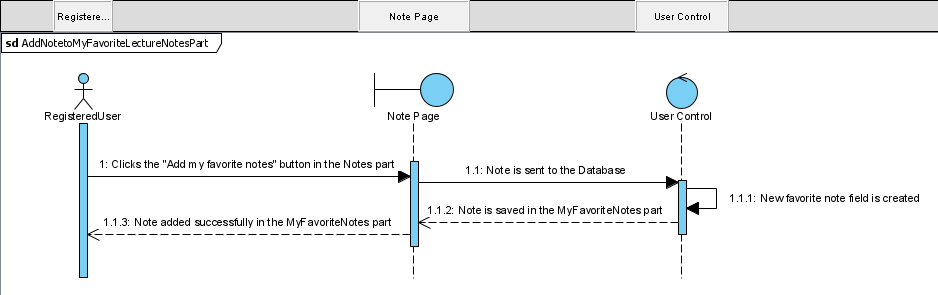
**Figure 29. Change Username**

******

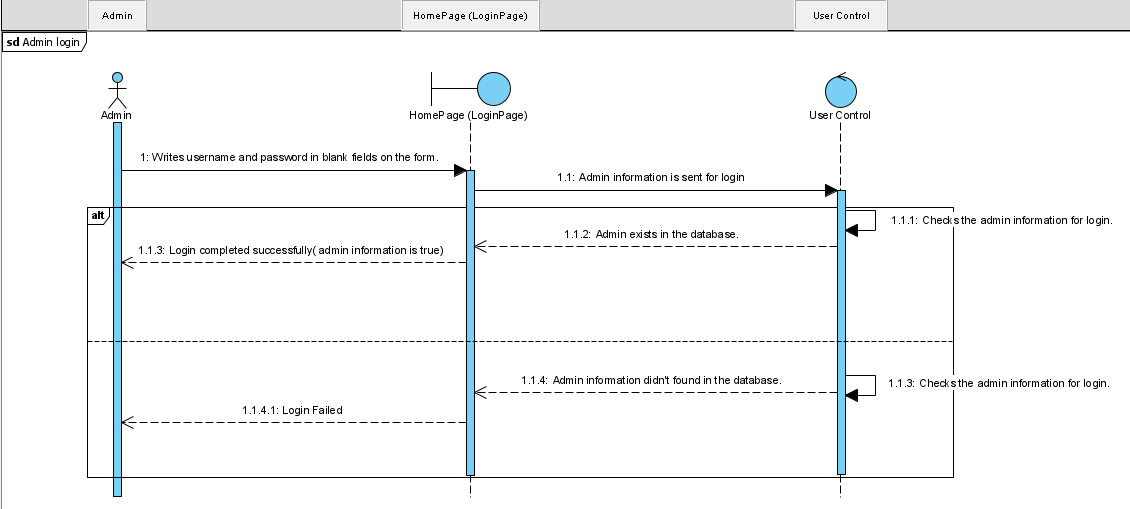
**Figure 30. Change Password**

******

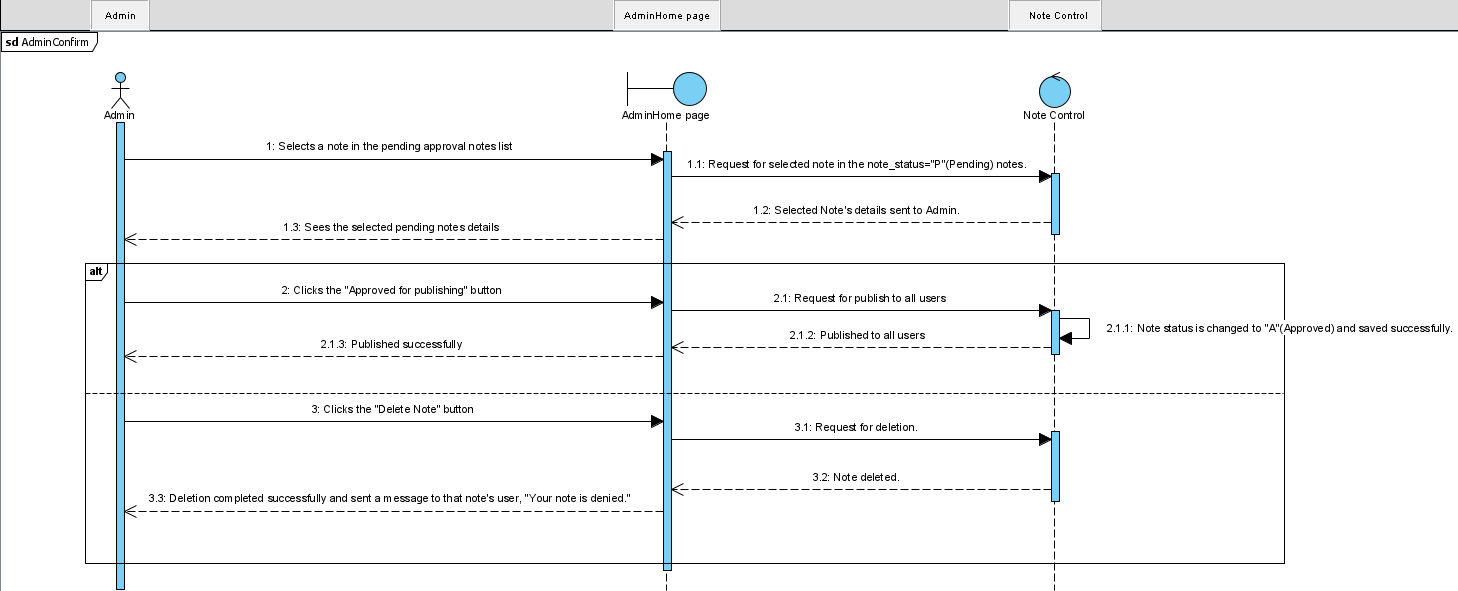
**Figure 31. Change Email**

******

**Figure 32. Add notes to my favorite notes part**

******

**Figure 33. Admin login**



**Figure 34. Admin confirm**

# Chapter 7

# 7.References

|  |
| --- |
| [1]. Şimşek, Seray -Öztürk, Anzel - “Requirement Analysis Document of  NoteOfTheEverything System”, March 26, 2021. |
| [2]. Şimşek, Seray - Öztürk, Anzel - “System Design Document of  NoteOfTheEverything System”, April 26, 2021. |
| [3]. Patrick Loeber, (2020, September 4). *python-engineer/pytorch-chatbot*. GitHub. https://github.com/python-engineer/pytorch-chatbot |
| [4]. G. (2018, June 21). *Contextual Chatbots with Tensorflow - Chatbots Magazine*. Medium. <<https://chatbotsmagazine.com/contextual-chat-bots-with-tensorflow-4391749d0077>>  [5]. G. (2020, January 8). *Deep Learning in 7 lines of code - Chatbots Life*. Medium. <<https://chatbotslife.com/deep-learning-in-7-lines-of-code-7879a8ef8cfb>>  [6]. G. (2020b, January 8). *How Neural Networks Work - Chatbots Life*. Medium. <https://chatbotslife.com/how-neural-networks-work-ff4c7ad371f>  [7]. *Security in Django | Django documentation | Django*. Retrieved at (2021, June 4). Django. <<https://docs.djangoproject.com/en/3.1/topics/security>>  [8].“Notedu.Com | Yeni Nesil Ders Notu Pazaryeri.” Notedu - Yeni Nesil Ders Notu Pazaryeri, Notedu, notedu.com. Accessed 16 May 2016. |