MINISTRY OF EDUCATION, CULTURE AND RESEARCH OF REPUBLIC OF MOLDOVA
TECHNICAL UNIVERSITY OF MOLDOVA
FACULTY OF COMPUTERS, INFORMATICS AND MICROELECTRONICS
DEPARTMENT OF SOFTWARE ENGINEERING AND AUTOMATICS

EGov.AI
Project report

FAF-212 team 6
Dobrojan Alexandru
Cernețchi Maxim
Smocvin Denis
Crucerescu Vladislav
Nastas Corneliu

Chișinău, 2023

Abstract

List of figures

Content

Introduction

In today's world security is a crucial principle in our society. As technologies and software continue to evolve, the threats related to them evolve as well and became a serious concern to the modern software engineer. Digitalization has increased the amount of personal/sensitive data that people expose to software. For example, passwords, personal photos, important messages, notes, location and movement data, payment and income data, etc. Digitalization has quickly become of interest to criminals as they could commit data/money theft, impersonalize people, perform harassment or blackmail in an online, anonymous environment. If security was not an approach in

modern technologies, our society would most probably be doomed by crimes as soon as it would take some steps to achieve something using software. Therefore, security is the backbone of software engineering as it is doing a very important job no less important than fighting real-world crimes like the police do.

Fighting cybercrimes is a paramount aim of modern organizations. Hence, the demand for secure products, technologies, algorithms and engineers is only growing year by year. Organizations are interested in protecting the customer from literally any threat possible and as fast as new ones appear. Cybercrimes produced damage up to six trillion dollars worldwide in 2021[1]. Banks require secure transactions, systems require protection from the outside, business organizations require identity proving and so on. A secure software application is very valued by any organization and is very much in demand. Therefore, security is of interest to a software engineer and the engineer would most probably not be able to create any software product without concern about its security.

Developing a secure software product is now a requirement and it is more and more adopted by academic environments specialized in software engineering to practice on developing such a product. More and more universities adopt the practice of assigning projects with topic on developing software products emphasizing on the security to the students. Our project is an example of such practice. It should emphasize the security aspects of a software product and create a secure environment for the end user. This project would represent a full working application with an idea that is a solution to some existing problem related to privacy/security.

Our key objective is to establish and research the aspect of security of a software product and develop the product in such a way that would represent the real-world analogy of creating a secure application. Starting with the research of the problems and ending with presenting our product to the world.

One of the main objectives is identity proving. I.e., how the user would prove his claim when doing something that would potentially hurt him. For instance, when doing a bank transaction, how would John Doe prove that he is the real John Doe doing the current transaction and not some third-party individuals that would steal all his money through a security breach of the bank application?

Another objective is safe communication between software nodes (website, server, application, service etc.). For example, the bank account information needed for transaction is transmitted in an encrypted format to the bank servers and back so that the third party, even after obtaining the encrypted information would not have the possibility to decrypt it, and hence the possibility to produce theft or any other instance of cybercrime.

This report will cover these two objectives and many other in depth and provide a thorough explanation of problems, considerations, technologies, algorithms and approaches used in developing of our product and provide example of usage and of security approaches using a scientific language and may be considered also as a guide for new software engineers that want to develop a secure product.

The Problem

The problem our team addressed in our project concerns the challenges associated with understanding and processing legal documents in Republic of Moldova. Specifically, we investigated the complexities of interpreting certain documents, the time constraints individuals face in comprehending lengthy legal texts, and the uncertainties involved when undertaking specific legal actions without clarity on the required steps or necessary documentation. An in-depth discussion of each sub-problem follows.

Complexity of documents

Legal documents were always a headache for regular people because they consisted of complex jargon with lots of terms and references foreign to a person not knowing the legal or historical context of the field covered by the documents. Regardless of the county or the field where the documents are issued, the complexity of legal text remains an invariant that an average person would find challenging to understand. There are lots of potential problems that may appear when some party would understand wrong some claims of a document and make mistakes (potentially destructive ones when it comes to important papers or businesses). One solution to that is hiring a lawyer or other person who knows well the legal aspects and with major experience in practice in that field. The problem with that solution is that acquiring work of such person is not a cheap in terms of money action. Many people find it very financially challenging to hire a legal professional. This is a huge impediment in dealing with law for people, so they often proceed in taking actions without fully understanding the claims from the documents because they did not truly understand them.

Project Subject

In an era where technology plays a pivotal role in enhancing accessibility and improving public services, we embarked on an ambitious project to bridge the gap between legal processes and the citizens of Republic of Moldova they impact. Recognizing the complexities inherent in navigating the myriad of legal systems and the challenges many face in understanding them, this project aims to develop and implement an intuitive legal information chatbot. This digital assistant is designed not only to answer common legal queries but also to educate and guide users through various legal processes. Through this initiative, we hope to promote transparency, reduce administrative burdens on legal offices, and ensure that every citizen, regardless of their background, has equal and easy access to essential legal information.

Objectives

1. **Accessibility and ease of use:** Make legal information easily accessible to all citizens, regardless of their background or expertise.
2. **Reduce Administrative Burden:** Decrease the volume of basic inquiries to government offices and legal professionals by providing instant answers to frequently asked questions.
3. **Consistency and Accuracy:** Ensure that citizens receive consistent and accurate information regarding legal processes and laws.
4. **Real-time Updates:** Offer real-time information reflecting the latest changes or amendments in laws and legal processes.
5. **Cost Efficiency:** Reduce costs associated with manual customer service and handling inquiries.
6. **24/7 Availability:** Provide round-the-clock support to citizens, accommodating those with varying schedules or time zones.
7. **User-Friendly Experience:** Implement a user-friendly interface, incorporating easy navigation and natural language processing capabilities, to allow users to pose questions in everyday language.
8. **Privacy and Data Security:** Ensure users' privacy by not storing personal data and maintaining robust security protocols to protect any data the chatbot does handle.

Functional overview

EGov.AI is a multifunctional system embedding features like real-time chatting with an artificial intelligence, chat sessions with the AI, authentication and authorization (identity) system including third-party identity providers. The user would have access to all the platform features at a 24/7 rate without any limitations.

Artificial intelligence

Our product haves an always-running pre-trained artificial intelligence capable of providing a real-time answer-question chatting functionality. The model is capable of understanding romanian text and understand the context of the text, I.e., understand the laws that are related to given sequences of text. Also, it would fully understand legal documents and provide an explanation in simple jargon to the end user so that the user would easily understand all the claims of the legal paper he provides to the AI. The user may ask in romanian to explain what some law means, and the AI would know how to explain in simple words. It is capable of generating concurrently multiple answers so that multiple users could use it at the same time. It would also recommend certain steps or give information about what papers are needed for certain tasks. The answer generation time is fair for the user as it takes several seconds to respond, even when the question is complex, or the document provided has a lot of text.

Chat sessions

- **Initialization:** Every chat session begins when a user initiates a conversation. This could be through a greeting, a direct question, or any other user input. Depending on its programming, the chatbot might respond with a default greeting or jump directly to addressing the user's query.
- **Context Retention:** For the duration of the chat session, it's crucial for the chatbot to retain context. This allows for a natural conversation flow and prevents users from having to repeat information.
- **Real-time Processing:** As users input questions or commands, the chatbot processes this information in real-time, tapping into its database, algorithms, or external data sources to generate an appropriate response.
- **Session Termination:** A chat session ends either when a user decides to close it or after a period of inactivity. Upon conclusion, the bot might offer a summary, ask for feedback, or provide options for further engagement.

- **Data Storage and Privacy:** Post-session, the chatbot system might store session data for analysis and improvement. However, it's crucial to ensure that any stored data complies with privacy regulations and that personal data is either anonymized or deleted.

In essence, a chat session is a dynamic, interactive experience, bridging the gap between users and the vast information repositories or services a chatbot taps into. It's a fusion of real-time data processing, user experience design, and natural language processing, all aimed at providing users with a seamless, informative, and engaging interaction.

Non-Functional overview

Our application contains multiple technologies, tools and patterns combined together. It consists of 4 main modules: client (the UI), gateway (main server), scraper and ML. Next there is a description of each module in part.

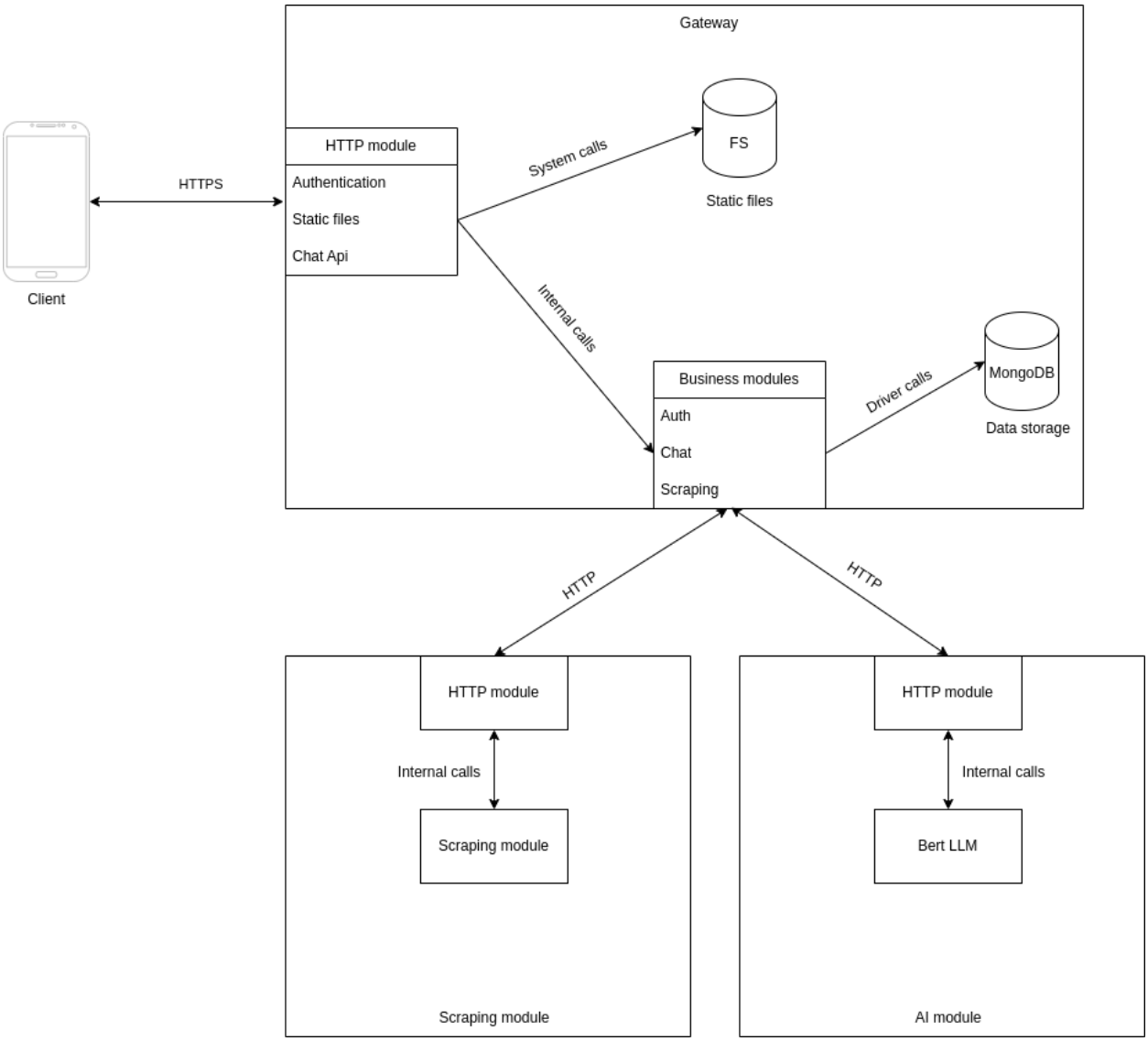

Fig. 1 Overall architecture

Scraper

The scraper is used in the API responsible for the AI part of the project. The result of the scraper is used to train the model, providing it with the necessary context for making a response.

The script provided is an excellent example of data gathering techniques utilized for populating datasets, especially in domains like legal studies where data is often structured yet vast. By automating browser actions with Selenium, the script fetches specific legal pages(The Constitution, the Criminal Code and the Civil Code) from legis.md. Once the content is loaded, BeautifulSoup is employed to parse and extract the relevant "articols" or articles. This method ensures that only pertinent information, delineated by specific markers (in this case, tags containing the word "Articol"), is captured. The resultant structured data, saved in a JSON format,

offers a rich dataset ready for further preprocessing and model training. Such an approach ensures that machine learning models, especially those designed for legal text analysis or natural language processing tasks, are trained on accurate, domain-specific content.

AI API

Because each user can have multiple chats, the API should store the context of each chat. Each chat will have its own instance of the model, with the conversation context saved, hence the chat handler.

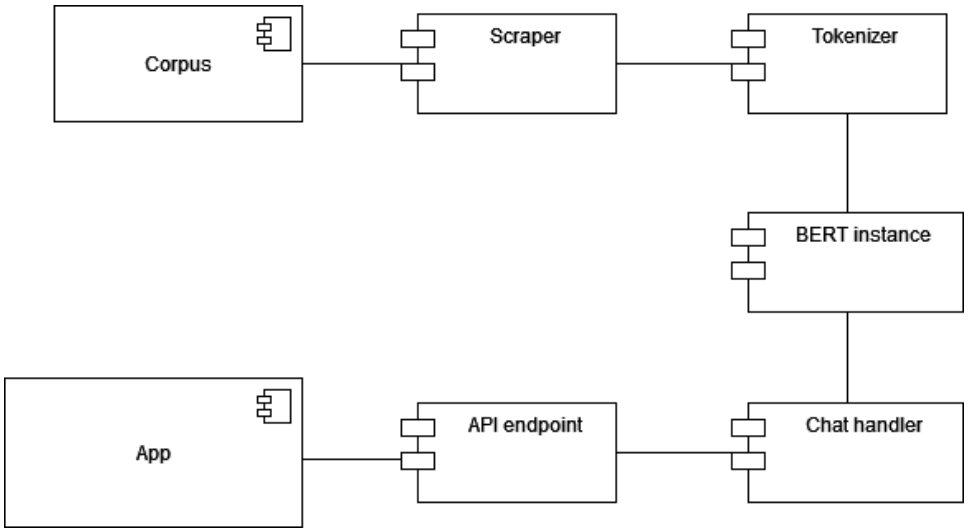

Fig. 1 AI API component diagram

Gateway

Gateway is a NestJs application. It is modular and has a module for each business feature, like static file serving, authentication, scraping, ML communication, environment setup, database connection etc. Its main feature is ease to setup and use. It comes with a handful of modules and built-in features like dependency injection, database management and relational object mapping, static file serving, URL parsing, communication over a variety of protocols, module lazy loading and more.

The gateway consists of an app module that is an entry point to all other modules, an authentication module that provides secure JWT-based authentication with email and password or using Google OAuth2, it comes with sign in and out HTTP endpoints. Chat module provides a bidirectional

communication with the ML module outside of the gateway on HTTP protocol. It is not encrypted because it is running on the same machine/system. Scraping module provides bidirectional communication over HTTP with the scraper module and will store data in database and call ML if needed.

Database

The data storage is realized by MongoDB document database. We chose this data storing approach (document) because of its high performance (because of the lack of traditional relations reading and writing is very fast) and ease of use. It is easily connected to the gateway through a dedicated NestJs module.

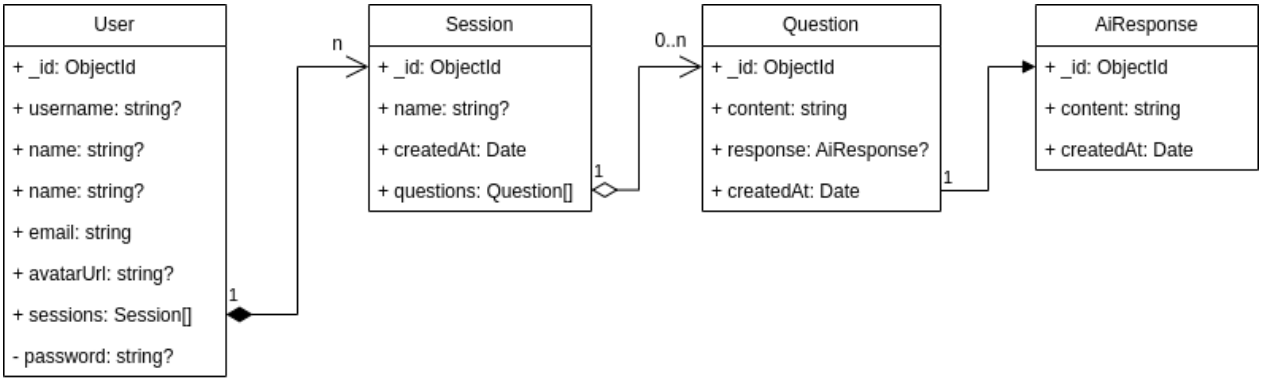

Fig. 3 Database schema

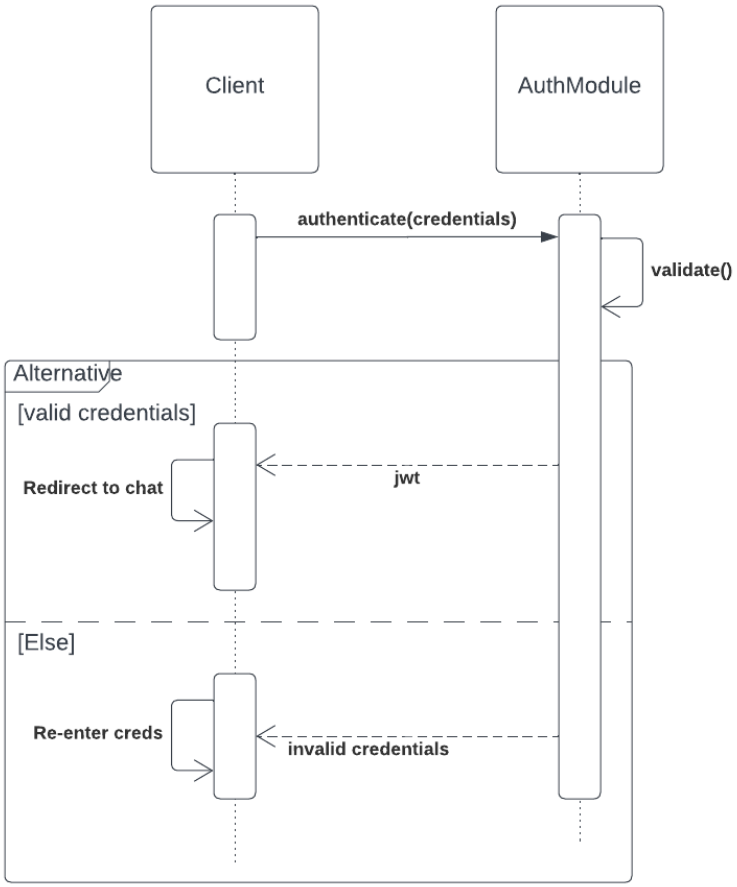

Fig4. Authentication Sequence Diagram

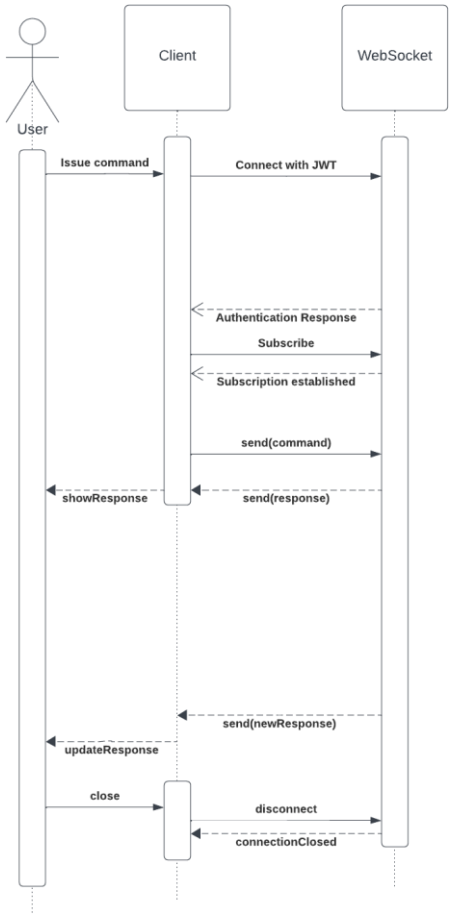

Fig 5. Client – WebSocket Communication Sequence Diagram

References

[1] More Than 70 Cybercrime Statistics – A $6 Trillion Problem [online]. [accessed 9 oct. 2023 13:20] Available: https://dataprot.net/statistics/cybercrime-statistics/