

Computer Science Department

Final Report – Spring 2024

Title of the project:	“AI Libra virtual library assistant: intellectualization of the NU Library web-portal user interface on the basis of AI applications”
Team Members:	Timur Unaspekov, Altay Toktassyn, Sat Slamkhan, Sirazh Gabdullin, Alisher Kenzhebayev
Project Advisor/Co-Advisors	Askar Boranbayev - Advisor Piotr Lapo - Co-Advisor

Executive Summary

This project is made to improve the existing library system of Nazarbayev University. Dr. Askar Boranbayev and Dr. Piotr Lapo offered us to take on this project. Our solution strives to make the NU library system convenient and efficient for students and faculty members. Our goal is to improve the user experience when working with the library system, make it more interactive, and also reduce the time it takes to find a book that matches the user’s interests.

Former NU Library Director, Dr. Piotr Lapo, together with our professor Dr. Askar Boranbayev, advised us to use artificial intelligence based on their experience. Our team, together with our advisors, came to a unified implementation.

We have introduced a virtual assistant. Firstly, it performs the function of voice announcements on the NU library website, using the Google Cloud Platform. Secondly, our assistant answers arbitrary questions from users regarding the library, based on the "spaCy" model. We also developed our own model to improve the skill of working with Natural Language Processing tasks. But in the end, we came to the conclusion that spaCy works better. We used a database of answers to frequently asked questions from the NU library website to train our model. Thirdly, we implemented a smart book recommendation system so that the user can get information about the book based on his interests and other users' reviews.

Our solution works in the format of a Backend web application, which in the future will be integrated with the NU library system. But while we have not integrated with the NU library system, we have written our own Frontend application for simulation in order to fully test it and demonstrate the results. Frontend is written in React. The backend is written using Fast API. PostgreSQL was used to manage the database.

It should be noted that our team led by Dr. Askar Boranbayev presented this project at the International Scientific and Practical Conference "Industrial Development: Technologies for People and Services in the Era of Innovation".

Introduction

In the modern realm of the digital world libraries has been successfully transformed from simple repositories of books to places of learning new things and innovation. Recognizing the need to provide a seamless and personalized user experience, our team embarked on a journey to enhance the library system of our university with the power of artificial intelligence (AI). This report outlines the development of an AI-powered library system for NU, named AI Libra.

NU's existing library system is outdated and falls short of the expectations of students and faculty. AI Libra addresses this need by providing a modern, AI-powered separate microservice for library systems tailored to the NU community's specific requirements. AI Libra is not merely a technological upgrade; it represents a paradigm shift in library services.

This report details the development of AI Libra, from initial planning to the implementation of the full-fledged book recommendation system trained on **200,000+** books using neural networks. It also covers challenges encountered during development and the solutions implemented.

Key Findings of our project:

AI Libra was implemented as a separate microservice.

AI Libra was trained using modern neural networks, especially using the PyTorch library.

AI Libra copes with answering different questions using advanced NLP models.

AI Libra aligns with the NU community's specific needs and preferences.

By leveraging AI's capabilities, AI Libra transcends the limitations of traditional library systems and elevates the user experience to unprecedented levels. Its comprehensive suite of features, including a virtual assistant, a book recommendation engine, and a complete answer generation system, empowers users to navigate the library's vast collection with ease and efficiency.

AI Libra's seamless integration with the existing NU library system ensures a smooth transition for users while expanding the library's capabilities. The system's development underscores NU's commitment to providing its students and faculty with the best possible learning environment. By embracing AI, NU has taken a bold step towards transforming its library into a true hub of knowledge, where innovation and discovery flourish.

Our team chose to work on this project due to its strong ties to our university. The NU library system's current version is outdated and needs to be updated because of the technology that was available when it was first built. AI is becoming increasingly popular, and we want to improve the NU library system by using the most recent Machine Learning advancements, particularly Unsupervised learning techniques and Neural Networks. Our goal is to enhance the NU library system by adopting the latest AI technologies.

Our solution will complement the existing library system. To recommend books, our system utilizes the internal Artificial Intelligence model. Our system will employ the Google Cloud Text To Speech neural network API for text-to-speech conversion.

We are confident that AI Libra will revolutionize the NU library experience and provide our students and faculty with the tools they need to succeed in their academic endeavors.

The development of AI Libra is a significant step forward in the evolution of library services. AI Libra has the potential to transform the way students and faculty interact with libraries and access information. We believe that AI Libra is a model for future library systems and that it will set a new standard for library service excellence.

Background and Related Work

To begin with, we decided to study what the use of artificial intelligence in libraries in general is, how it was used and how useful it was. We were helped with this by such articles as "Artificial Intelligence in Libraries" by Isaiah Michael Omame and Juliet C. Alex-Nmecha (2020), and "Application of Artificial Intelligence (AI) In Libraries and Its Impact on Library Operations Review" by Subaveerapandiyan A (2023).

First of them - "Artificial Intelligence in Libraries" - was written in January 2020 by a professor of University of Port Harcourt and a librarian from Federal University of Lafia. The authors argue that Artificial intelligence plays a crucial role in libraries as it facilitates the organization and accessibility of vast troves of information. Moreover, artificial intelligence stands as the cutting-edge technology employed in managing digital libraries (Omame and Alex-Nmecha, 2020). The article provides a comprehensive overview of the topic, including the different ways that AI is being used in libraries. For example, they (2020) say that Artificial intelligence has permeated a wide range of library information services, like:

Automatic cataloging and classification using Optical Character Recognition (OCR)

Automatic translation of foreign language materials using Natural Language Processing (NLP)

Automatic indexing using Expert Systems

Retrieval of audiovisuals materials Optical Character Recognition and Speech Recognition. Music and pictures in the library's collections can be called up as fast as printed records – a new dimension to knowledge storage and management

Interactive bibliographic instruction using various media

Intelligent gateways to online sources

User-structured information environment

Portable computer reader services for the handicapped

Intelligent Document Delivery Services (DDS)

Authors also studied Romero (2018) and highlighted the ability of artificial intelligence systems to provide personalized recommendations based on past purchases or user interests. This strategic approach can optimize library material acquisition and enhance user experience through tailored suggestions of magazines, journals, authors, books, and other relevant materials. This method will be used in our project, too.

The second article - "Application of Artificial Intelligence (AI) In Libraries and Its Impact on Library Operations Review" - serves as a valuable resource for understanding the transformative potential of AI in libraries. The author wrote that in the study of Panda and Chakravarty (2022) reveals that AI chatbots provide a robust solution for launching virtual assistance, bolstering reference services, and promoting the concept of a "library without walls." The paper concludes that chatbots can elevate service quality, personalization, engagement, automation, revenue generation, and user satisfaction in libraries, serving as a valuable marketing tool and enhancing information literacy (Oname and Alex-Nmecha, 2020).

Subaveerapandiyan (2023) argues that AI algorithms can analyze user preferences and past interactions to generate personalized recommendations for books, articles, and other library resources, mentioning the works of Tundra et al. (2020), Pence (2022), and Lin et al. (2023). These works also inspired us to add this feature to AI Libra.

There is another article that inspired us to integrate on the first steps ChatGPT in our system. It is the article by Chen (2023) called "ChatGPT and Its Possible Impact on Library Reference Services". The author conducted a comparative analysis of ChatGPT with traditional library chatbots currently in use at the University of California, Irvine (ANTSwerssays), Lehman College (Lehman Lightning Bot), and San Jose State University (KingBot). For example, Chen made a query "What is the best library database for finding law reviews and legal cases?", and asked all the chatbots. ChatGPT successfully identified LexisNexis, Westlaw, JSTOR, and HeinOnline as potential top-tier library databases for locating law reviews and legal cases. However, ANTswerssays, Lehman Lightning Bot, and KingBot couldn't find any suitable answer, saying something like "It appears I may not have the specific information you require".

In conclusion, he states that ChatGPT appears far more capable and informed than the pre-ChatGPT chatbots currently employed by some libraries. It possesses the potential to outperform traditional chatbots in providing library reference services (Chen, 2023). That is why our team wants to integrate it to AI Libra as a baseline benchmark.

Project Approach

System features

Book recommendation feature

This feature significantly improves the user experience. Users ask to recommend similar books based on the titles they have provided. Our Neural Network model was trained on more than 220,000 books collected from the Amazon website. For each of these books, we have corresponding book reviews and book ratings with the user information. Our Artificial Intelligence model counted up all these components and created own recommendation model. After finding the corresponding title in its database, it generates top five similar books based on the factors mentioned above. Finally, our text-to-speech Google Cloud feature voices this recommendation.

Sentiment Analysis with LSTM Networks

In order to train our model we had to use Long-Short-Term Memory networks. They are a special kind of Recurrent Neural Network, which aligns perfectly with the given task. This type of network is based on sequential data. Their ability to remember and forget information fits our need to analyze the sentiment of each book. However, to provide books and their data as input to this type of Neural Network we had to perform some data manipulation. Firstly, we converted each book's words into singular tokens(tokenization). After this step, we have removed all common words in English which carry no useful information for our model. Finally, we converted our tokens into high-dimensional vector space. In this vector space, words that are close in meaning have low Euclidean distance. For example, words like "mother" and "father" can lie in similar coordinates, even if their tokens are not similar. This approach allows computers to understand words and sentences as objects on which we can train our model.

books.csv (3.3 MB) ↓ 🔍 >

Detail Compact Column 10 of 23 columns ⌵

id	book_id	best_book...	work_id	# books_cou...	work_id	# books_cou...	isbn	isbn13	authors
1	2767852	2767852	2792775	272	2792775	272	439823483	9.78843982348e+12	Suzanne Coll
2	3	3	4640799	491	4640799	491	439554934	9.78843955493e+12	J.K. Rowling Mary GrandPr
3	41865	41865	3212258	226	3212258	226	316015849	9.78831601584e+12	Stephanie Me
4	2657	2657	3275794	487	3275794	487	61120081	9.78806112008e+12	Harper Lee
							743273567	9.78874327356e+12	F. Scott Fitzgerald
							525478817	9.78852547881e+12	John Green

Figures 6 and 7. Data of book.csv

Figure 1. Data of book.

ratings.csv (12.44 MB)

Detail Compact Column

About this file

book-user-rating triples.

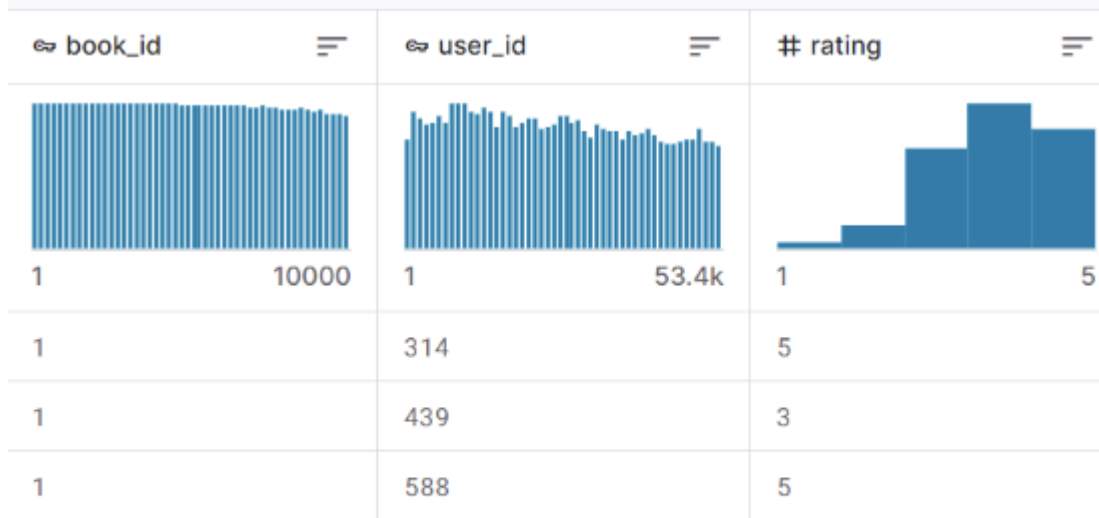


Figure 8. ratings.csv

Figure 2. Each books rating.

Recommendation Engine

Our recommendation system leverages a sophisticated hybrid approach, integrating both collaborative and content-based filtering methods to deliver highly personalized book recommendations. This dual-method strategy enhances the system's ability to match user preferences with appropriate book selections effectively.

a) Collaborative Filtering (CF)

Collaborative Filtering recommends items based on the preference patterns of users. This approach can be subdivided into two primary methods:

- 1) **User-Based Collaborative Filtering:** This method recommends items by identifying users with similar behaviors and preferences to the target user. It utilizes a user-item interaction matrix to discover these similarities, recommending items favored by users in the target user's similarity cluster.
- 2) **Item-Based Collaborative Filtering:** Conversely, this method focuses on the relationships between items rather than users. It recommends items similar to those a user has previously liked, based on the similarities

found in the transpose of the user-item interaction matrix. These methods are particularly effective in environments with rich user interaction data, allowing the system to predict preferences based on collective insights.

b) Content-Based Filtering (CBF)

Content-Based Filtering recommends items by analyzing the features of the items themselves and matching them with the user's preferences. In the context of our book recommendation system, relevant features include genres, authors, and sentiment scores derived from user reviews. Each book is represented by a profile composed of these features, and recommendations are made by matching book profiles with the user's demonstrated preferences

c) Hybrid Approach

Our system combines the strengths of both collaborative and content-based filtering to refine the recommendation process:

Initial recommendations are generated using collaborative filtering to harness the predictive power of user similarities.

These recommendations are then fine-tuned using content-based filtering, which adjusts the recommendations based on the detailed features of each book, such as sentiment scores.

d) Example Application: Suppose the collaborative filtering model suggests three books based on user similarity scores:

Book A (CF score: 0.8)

Book B (CF score: 0.6)

Book C (CF score: 0.7)

Subsequently, the content-based filtering model assesses these books based on their content scores:

Book A (CBF score: 0.9)

Book B (CBF score: 0.7)

Book C (CBF score: 0.8)

The final recommendations are computed by combining the scores from both models:

Book A (Combined score: 1.7)

Book B (Combined score: 1.3)

Book C (Combined score: 1.5)

The system prioritizes the books with the highest combined scores, thus providing a balanced recommendation that considers both user preferences and the intrinsic qualities of the books.

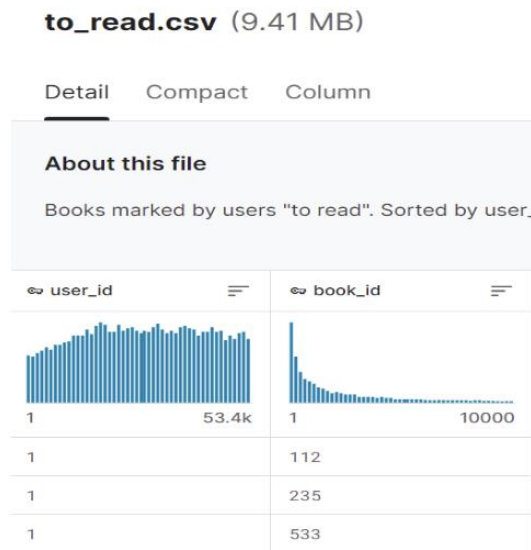


Figure 9: to_read.csv

Figure 3. Books marked "to read".

User Feedback and Iterative Learning

To enhance the system's accuracy continually, we incorporated a feedback loop where users could rate the relevance of the recommendations. This feedback is used to fine-tune both the sentiment analysis model and the recommendation engine. Adjustments are made based on whether the recommendations align with user expectations, allowing the system to learn from real-world interactions and improve over time.

In summary, our machine learning models are integral to creating a personalized and responsive book recommendation system. By combining sophisticated NLP techniques with user-centric recommendation algorithms, the system provides targeted suggestions that resonate with the users' preferences and emotional states, enhancing their reading experience significantly.

Complete answers generation to arbitrary user questions

This feature reduces the time it takes to find an answer to a user's question regarding NU library services. Instead of looking for an answer to the user's question in the Frequently Asked Questions section, the user will enter his question as he sees fit. To be more precise, there is no need to ask a question using a template. The interaction takes place in the form of a chat. The user enters his question and then AI Libra generates an answer for him based on what it knows from the FAQ section. Her answers are displayed in the chat, and the virtual assistant also voices them. To implement this feature our team used a step-by-step approach. The first step is to fill out our database with all answers to FAQ questions and save their URL. Since the NU website has more than 200 frequently asked questions, filling out them manually is inefficient and time-consuming. We decided to implement our parser for these questions. As a consequence, we started analyzing the library website's {HTML} structure to accurately parse it. First, we encountered the problem that all of the questions on the NU library website were not stored as straightforward HTML tasks but rather were generated by JavaScript. This means, that we cannot just simply download HTML pages and find what we need.

```

::before
▼ <div id="s-la-page-column-1" class="s-la-page-column col-md-8">
  ▼ <div id="s-la-box-40835-container" class="s-la-box-container s-la-box-border-simple s-la-box-borderless s-la-box-type-11 s-la-box-position-1">
    ▼ <div id="s-la-box-40835" class="s-la-box s-la-box-html">
      ▶ <div class="s-la-box-title s-la-box-title-simple s-la-box-border-simple">⋮</div>
      ▼ <div id class="s-la-box-content clearfix">
        ::before
        ▶ <ul class="nav nav-tabs" role="tablist">⋮</ul>
        ▼ <div class="tab-content">
          ▼ <div id="s-la-box-40835-container-tab0" class="tab-pane active" role="tabpanel" aria-hidden="false" tabindex="0" aria-labelledby="ui-tab-140">
            ▼ <div class="tab-pane-content">
              ▼ <div id="s-la-content-faqlist-39811" class="s-la-content-faqlist s-la-content clearfix">
                ::before
                ▼ <ul class="list-unstyled">
                  ▼ <li class="s-la-faq-listing">
                    ▼ <div class="s-la-faq-listing-q"> == $0
                      <a href="https://nu-kz.libanswers.com/faq/240345">How can I use my library account?</a>
                    </div>
                    ▶ <div class="s-la-faq-listing-meta">⋮</div>
                  </li>
                  ▶ <li class="s-la-faq-listing">⋮</li>
                  ▶ <li class="s-la-faq-listing">⋮</li>
                  ▶ <li class="s-la-faq-listing">⋮</li>
                  ▶ <li class="s-la-faq-listing">⋮</li>
                  ▶ <li class="s-la-faq-listing">⋮</li>
                  ▶ <li class="s-la-faq-listing">⋮</li>
                  ▶ <li class="s-la-faq-listing">⋮</li>
                  ▶ <li class="s-la-faq-listing">⋮</li>
                </ul>
                ::after
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

```

Figure 4. NU Library HTML Structure.

After some research, we found that there exists a **Selenium** web browser, which can emulate real browsers for parsing purposes. After writing a script to access our NU website using the Selenium browser, we had to deal with the second problem. The answer was stored on a separate web page. This added additional complexity to our project because we had to open each question's URL, parse additional answers and save it to DataBase. Each request was too slow and took too long time to parse the answer and corresponding URL. To fix this issue, we enhanced our script with **multi-threading** functionality. As a consequence, each request was executed in a separate thread, which boosted the speed of the parser by **5** times.

```

1 usage altay.toktassyn *
def parse_answers_from_urls(faq_list, browser):
    with ThreadPoolExecutor(max_workers=5) as executor:
        future_to_url = {executor.submit(fetch_and_parse, question['url'], browser): question for question in faq_list}
        for future in concurrent.futures.as_completed(future_to_url):
            question = future_to_url[future]
            try:
                question['answer'] = future.result()
                print(f"Answer: {question['answer']}\n")
            except Exception as exc:
                print(f"{question['url']} generated an exception: {exc}")
    return faq_list

```

Figure 5. Code snippet for multithreading functionality.

```
---
Question: How do I search for patents?
URL: https://nu-kz.libanswers.com/faq/178959
---
Question: How to find DOI?
URL: https://nu-kz.libanswers.com/faq/180346
---
Question: Fines and Fees
URL: https://nu-kz.libanswers.com/faq/241415
---
Question: Damaged Books
URL: https://nu-kz.libanswers.com/faq/184546
---
Question: How can I conduct a case study?
URL: https://nu-kz.libanswers.com/faq/183114
```

Figure 6. Example of working script.

```
browser.get(url)
browser.implicitly_wait(3)
number_of_pages = 10
faq_list = []
for current_page in range(0, number_of_pages):
    try:
        next_page_button = browser.find_element(By.CSS_SELECTOR, f'a[data-page="{current_page + 1}"]')
        next_page_button.click()

        time.sleep(2) # Example: wait for 5 seconds
        soup = BeautifulSoup(browser.page_source, 'html.parser')
        faq_links = soup.find_all('div', class_='s-search-result-title')
        faq_list = faq_list + parse_page(faq_links)

    except Exception as e:
        print("Failed to click the next page button or scrape its content:", e)
        break

answers = parse_answers_from_urls(faq_list, browser)
```

Figure 7. Code snippet for FAQ parsing.

Finally, as an outcome we received a fully filled database with all FAQ questions and their URL's which were seamlessly integrated into our Front-End service. However, this was the first part of this functionality. As a second part, we had to deal with Natural Language Processing.

Name	Description
question * required (query)	<input type="text" value="How can I scan?"/>

Figure 8. Question example.

```
"There are 2 types of scanners you can use at the library:\n\nOn the 2nd floor there are flatbed scanners in the computer lab.  
These work well for ordinary books and documents\n\nOn the  
2nd and 3rd floor near the self station, you can find bigger scanners  
if you need to scan a book that is hard to keep open flat on small scanners. Use these for  
voluminous books like encyclopedias.<0xa0>\n\nAlways remember to bring your USB with you to save your files!"
```

Figure 9. Answer example.

We decided to use **the spaCy** Python library, which generated precise answers to the required questions.

```

@router.post("/get_faq_answer/")
async def get_faq_answer(
    question,
    db: Session = Depends(deps.get_db),
):
    query = select(FrequentlyAskedQuestion)
    faqs = (db.scalars(query)).all()

    user_question_keywords = preprocess(question)

    max_matches = 0
    best_match_answer = "Sorry, I couldn't find an answer to your question."
    for faq in faqs:
        faq_keywords = preprocess(faq.question)
        common_keywords = set(user_question_keywords) & set(faq_keywords)
        if len(common_keywords) > max_matches:
            max_matches = len(common_keywords)
            best_match_answer = f'answer = {faq.answer}, url = {faq.url}'

    return best_match_answer

```

Figure 10. Code snippet for custom FAQ answering.

altay.toktassyn

```
@router.post("/get_faq_advanced_answer/")
async def get_faq_advanced_answer(
    question,
    db: Session = Depends(deps.get_db),
):
    query = select(FrequentlyAskedQuestion)
    faqs = (db.scalars(query)).all()

    user_query = nlp(question)
    max_similarity = 0
    most_relevant_answer = ""

    for faq in faqs:
        faq_question = nlp(faq.question)
        similarity = user_query.similarity(faq_question)
        if similarity > max_similarity:
            max_similarity = similarity
            most_relevant_answer = faq.answer

    return most_relevant_answer
```

Figure 11. Code snippet for advanced FAQ answering.

Broadcast of Library announcements

This system feature allows the NU library system to interactively provide information about upcoming events, library services, and changes in the library's work. It is also necessary to note that the virtual assistant AI Libra voices the information using **Google Cloud Text to speech**. Students can listen to this as if they were listening to the radio and do their own thing at the same time.

	id	full_name	email	hashed_password	is_active	is_superuser	role
1	1	Altay	altay@example.com	\$2b\$12\$SnxzSpvQMrYIoYmfaZc6...	• true	false	client
2	2	altay	altay@gmail.com	\$2b\$12\$w4RqEjpgf6Bmi2VHvfsC...	• true	false	client
3	3	altay12	altay123@example.com	\$2b\$12\$UYI5f4kahDW1Q4rBLBoN...	• true	• true	library_...

Figure 12. Role diversification.

To implement this feature we CRUD service from the backend site with the separation of users and library workers. Library workers can create, update, and delete all announcements, while users can only view particular news. The role diversification is being done on the database level. Apart from that we used JSON Web Tokens which helps us to distinguish users and admins from each other without accessing the database several times, since each token stores all information about the current user. Apart from that, it provides top-level security for our service, since each token has a signature which is checked after each request.

```

altay.toktassyn
@router.get("/{announcement_id}", response_model=schemas.AnnouncementOut)
def read_announcement_by_id(
    announcement_id: int,
    current_user: models.User = Depends(deps.get_current_active_user),
    db: Session = Depends(deps.get_db),
) -> Any:
    """
    Get a specific user by id.
    """
    current_announcement = crud.announcement.get(db, announcement_id)
    return current_announcement

altay.toktassyn
@router.delete("/{announcement_id}")
def remove_announcement_by_id(
    announcement_id: int,
    #current_user: models.User = Depends(deps.get_current_active_superuser),
    db: Session = Depends(deps.get_db),
) -> Any:
    """
    Get a specific user by id.
    """
    crud.announcement.remove(db, id=announcement_id)
    return 'Success'

```

Figure 13. Code snippet for announcements CRUD.

```
class CRUDUser(CRUDBase[User, UserCreate, UserUpdate]):
    6 usages (5 dynamic) altay.toktassyn
    def get_by_email(self, db: Session, *, email: str) -> Optional[User]:
        return db.query(User).filter(User.email == email).first()

    altay.toktassyn
    def create(self, db: Session, *, obj_in: UserCreate) -> User:
        db_obj = User(
            email=obj_in.email,
            hashed_password=get_password_hash(obj_in.password),
            full_name=obj_in.full_name,
            is_superuser=obj_in.is_superuser,
        )
        db.add(db_obj)
        db.commit()
        db.refresh(db_obj)
        return db_obj

    3 usages (3 dynamic) altay.toktassyn
    def update(
        self, db: Session, *, db_obj: User, obj_in: Union[UserUpdate, Dict[str, Any]]
    ) -> User:
        if isinstance(obj_in, dict):
            update_data = obj_in
        else:
```

Figure 14. Code snippet for user role diversification.

```

if TYPE_CHECKING:
    from .item import Item # noqa: F401

10 usages  👤 altay.toktassyn
class Announcement(Base):
    id = Column(Integer, primary_key=True, index=True)
    text = Column(String)
    date = Column(
        DateTime,
        nullable=False,
        default=datetime.datetime.utcnow,

```

Figure 15. Model description using SQLAlchemy ORM.

Requirements and Functionalities

Functionality and Architecture

- 1) Our AI Libra system was developed as a separate microservice using FastAPI Python Framework that interacts with the main NU library system via REST API.
- 2) The system consists of server-side and front-end side. This is due to the fact that the client side for processing requests from the user will be added to the NU library system itself because we are not allowed to access the source code of the current NU Library system. In order to demonstrate the work of AI Libra, our

team developed a separate front-end client side for demonstration of interaction with AI Libra using React.Js.

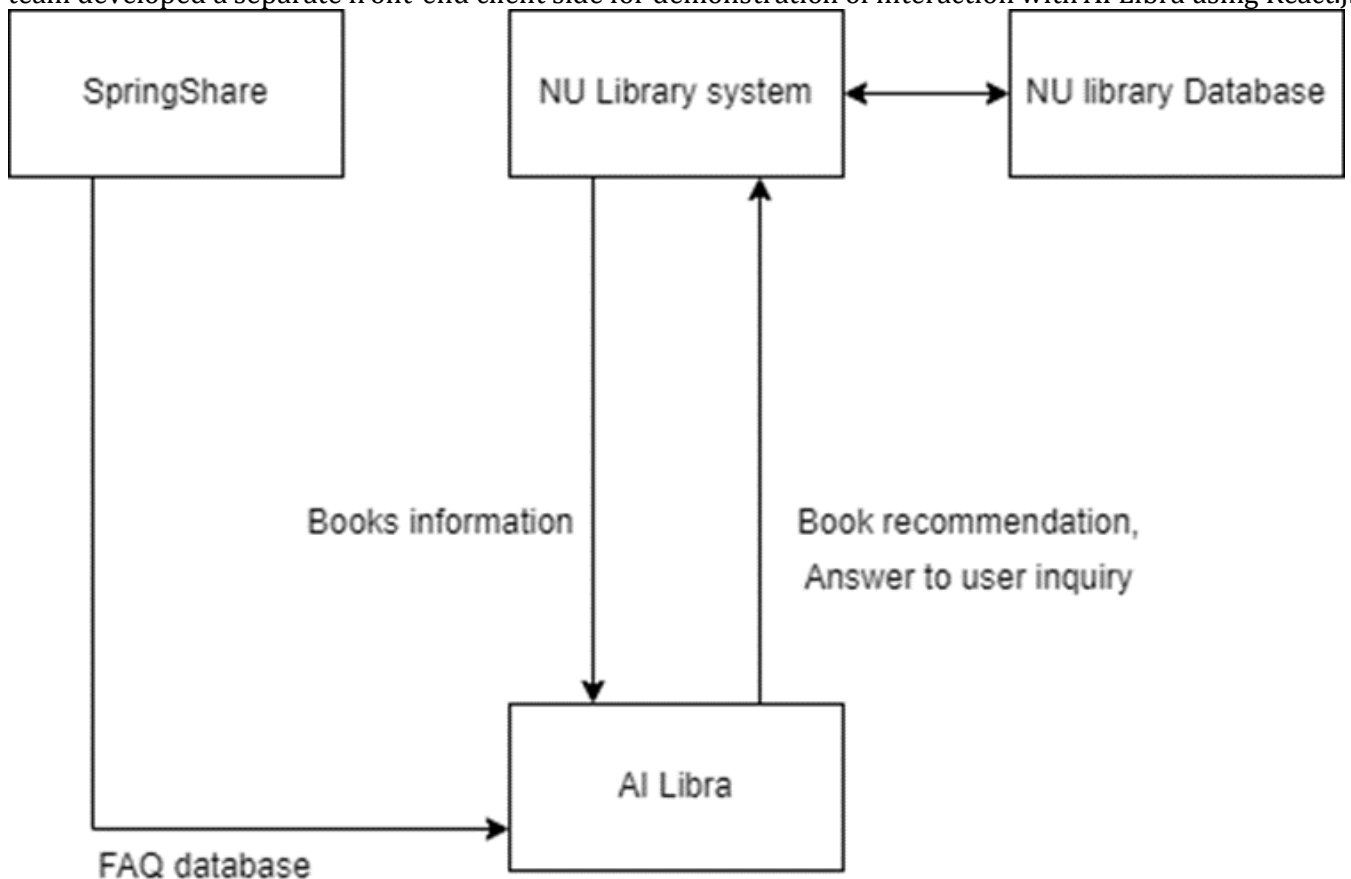


Figure 16. System behavior.

3) Technology stack:

- For the frontend - HTML, CSS, JavaScript, React.js - This is the golden standard for modern frontend systems, which has great community, appropriate speed, and good maintainability.
- For the backend - FastAPI, SQLAlchemy, Alembic, Docker, Git, PyTorch - These technologies best suit our needs, because they are fully asynchronous and parallel, which means that we can serve a lot of requests. Apart from that, they allow us to fastly create changes to DataBase using migrations.
- Database - PostgreSQL - Most of our team members have experience with this popular DBMS, so we have decided to use it.

4) Integration:

- To access the FAQ database, AI Libra will interact with the Springshare platform, with which the main NU library system closely interacts.
- AI Libra will request data on all existing books in the NU library system according to the NU Library system API. It is worth noting that AI Libra will not request any confidential information. For example, we will not have access to information about which students read a given book. We will store information about books in our own database so that we will not burden the NU library system with constant requests.
- We will provide the developers of the NU library system with our own API, through which they can interact with our system.

Use Case Specifications

The Figure below represents a structured use cases diagram. The user has four use cases. First, he can ask the virtual assistant to recommend a book based on book reviews and author information. Secondly, the user can simply find a book by title. In both cases, the user receives a preview of brief information on the books. In terms of the last two use cases, a user can ask questions arbitrarily to the virtual assistant and turn on library information broadcasting. When it comes to library assistant, his use cases are only related to entering information: adding text for the library information broadcasting and adding/editing information about new/existing books, respectively.

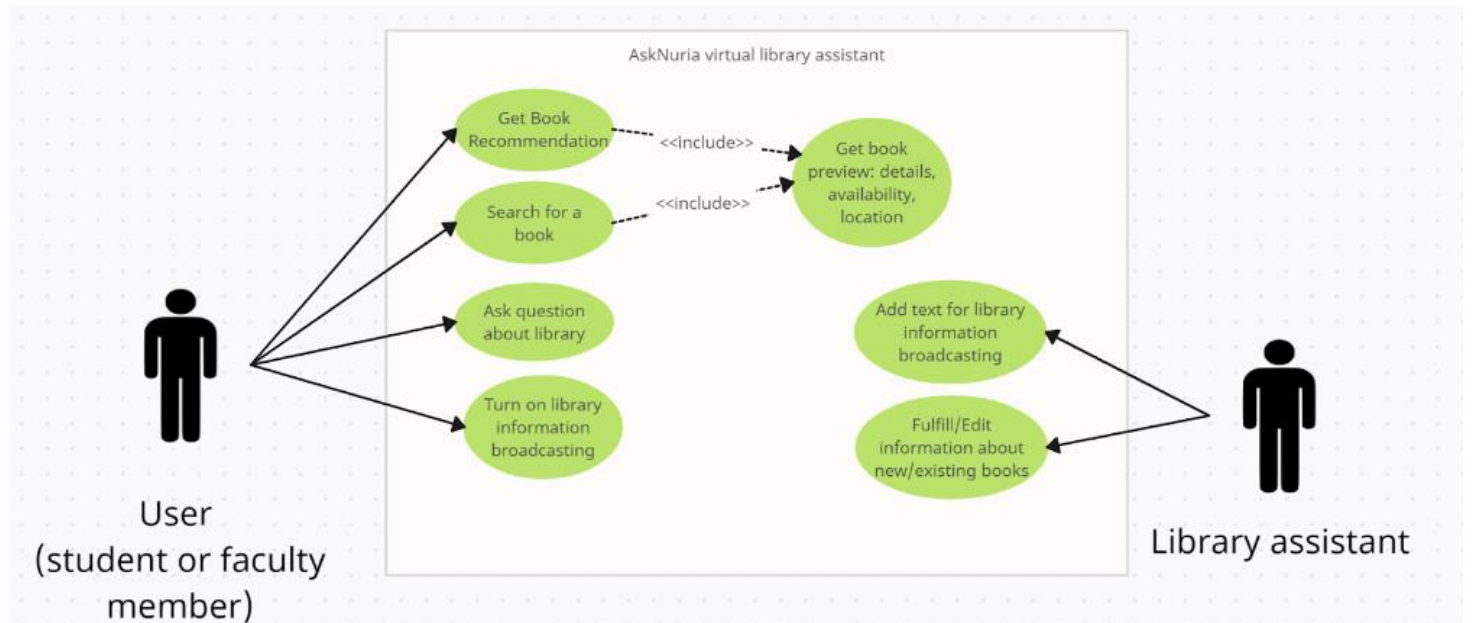


Figure 17. Use cases.

Use Case Name: **Get Book Recommendation**

Actor: User (student of faculty member)

Preconditions:

- a) The user have access to the chat
- b) The user provided book title.

Postconditions: The user receives a list of relevant book recommendations based on author information and book rating.

Basic flow:

- a) The user asks Nuria for recommendations.
- b) The Nuria asks the user for books titles they previously read.
- c) The user provides data.
- d) The AI-Libra model generates personalized response.
- e) Nuria displays/tells results to the user through chat.
- f) User reads/listens to the recommendations and asks for more details about specific recommendation(s).
- g) Nuria responds with relevant data (author, year, summary, etc.).

Alternative flow (starts from 1.b)

- a) The user refuses to give data.
- b) The AI-Libra generates general response based on request text (e.g. "give me books on statistics").
- c) Nuria displays/tells results to the user through chat.
- d) User reads/listens to the recommendations and asks for more details about specific recommendation(s).
- e) Nuria responds with relevant data (author, year, summary, etc.).

{Use Case Name:} Search for a book

{Actor: }User (student of faculty member)

Preconditions:

The user has access to the chat.

The book exists in the database.

Postconditions: The user receives details about the book they searched.

Flow of events:**Basic flow:**

The user searches for a specific book by making query to Nuria through the chat

AI-Libra finds the requested data in the database and generates responses.

Nuria displays/tells results to the user through chat.

User reads/listens to the details about the book they looked for.

Alternative flow

The user searches for a specific book.

AI-Libra fails to find the requested data in the database.

Nuria tells the user about the failed search attempt. Prompts the user to ask for recommendations as an alternative way to find relevant book(s).

{Use Case Name:} Ask Question about Library

Actor: User (student of faculty member)

Preconditions:

The user has access to the chat.

Postconditions: The user receives an answer to their query.

Flow of events:

Basic flow:

The user asks AI-Libra in chat a question about the library.

AI-Libra generates an answer based on parsed FAQ database.

Nuria displays/tells results to the user through chat.

User reads/listens to the answer for their question.

Alternative flow

The user searches for a specific book

AI-Libra fails to generate a response

Nuria tells the user about the failed query.

Nuria offers to ask a question to the library assistant. Provides ways to contact them.

{Use Case Name:} Turn on Library Information Broadcasting

Actor: User (student of faculty member)

Preconditions:

The user has access to the chat.

There current events list is not empty

Postconditions: The user receives information about interesting library events

Basic flow:

The user finds interesting event/announcement on the chat sidebar

The user asks Nuria more about the specific event

AI-Libra generates an answer based on the events database

Nuria displays/tells results to the user through chat.

User reads/listens to the Nuria broadcasting information about the selected event(s)

Alternative flow:

The user finds interesting event/announcement on the chat sidebar

The user asks Nuria more about the specific event

AI-Libra fails to generate an answer based on the events database

Nuria informs the user about the failed query.

Nuria offers to ask a question to the library assistant. Provides ways to contact them.

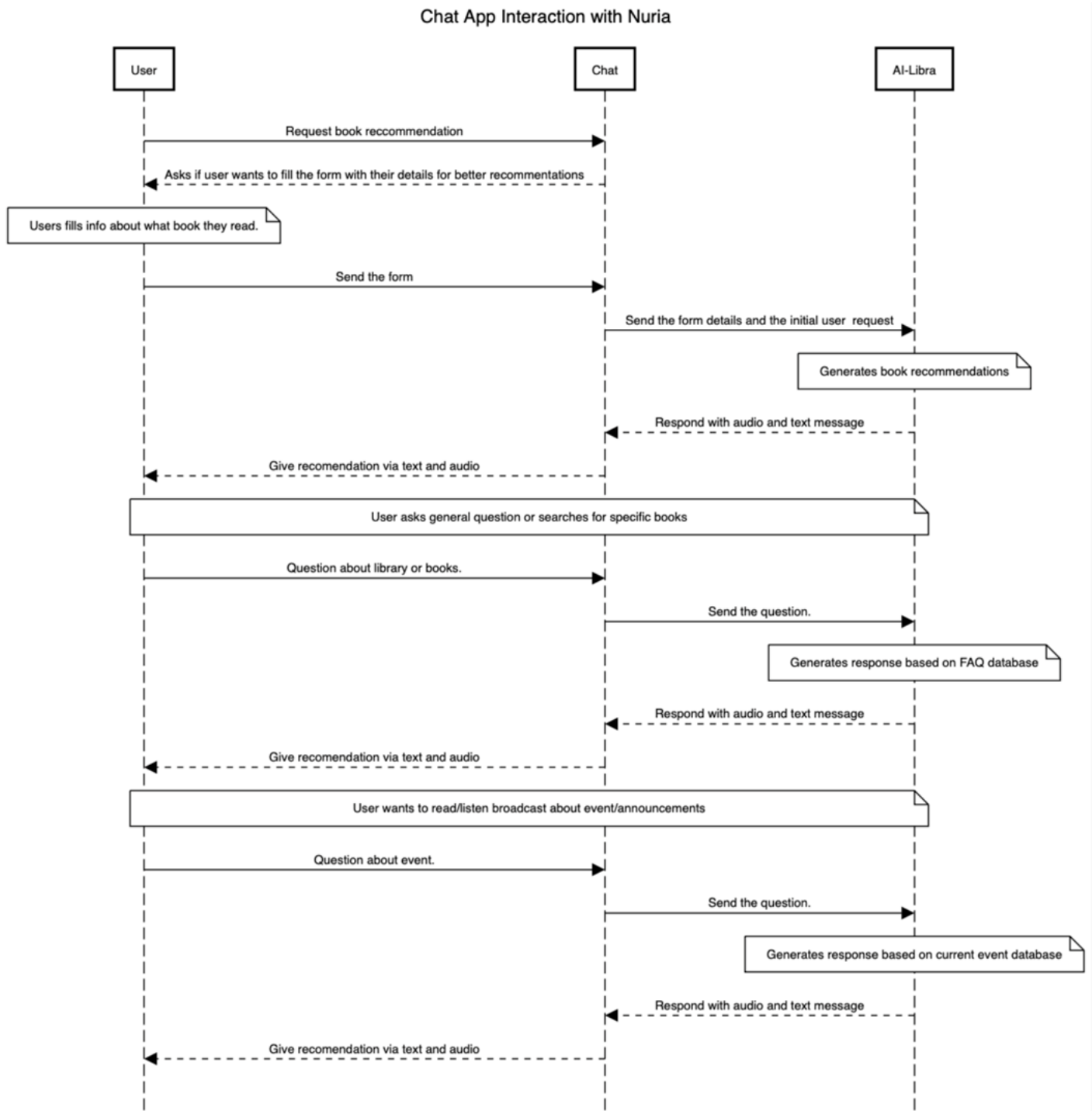


Figure 18. Interaction with chat bot.

The figure 8 is a sequence diagram that demonstrates how the user will interact with our system to use its main functionalities. It consists of 3 main parts.

1) Users asks for book recommendation;

a) To generate responses our model will need some user specific data. In our case it will be a book title that the user already read. To achieve that we ask the user to provide this info whenever the user asks for a recommendation. The model analyzes the data based on trained data and gives audio and text responses.

b) If the user does not provide the data, the model will try to answer as relevant as possible based on its knowledge of books. You can see the example in the design prototype (fig 2).

2) The user asks questions about the library or book.

a) AI-Library can generate answers to library-related questions. Whenever a user makes a request, the model provides the relevant answer based on the current FAQ database. Each answer contains a URL to the official answer at the NU library website.

b) Users can also ask questions about specific books, you can see the example in the design prototype (fig 2). Our system will provide book info and cover logo.

3) The user wants to read/listen broadcast about event/announcements

The chat user will be able to see the current events and announcement on the right sidebar (see fig 2). Whenever a user is interested in a specific event, they can click on it or ask Nuria directly in the chat. The Nuria then sends a request to our AI-Libra model that generates audio and text responses.

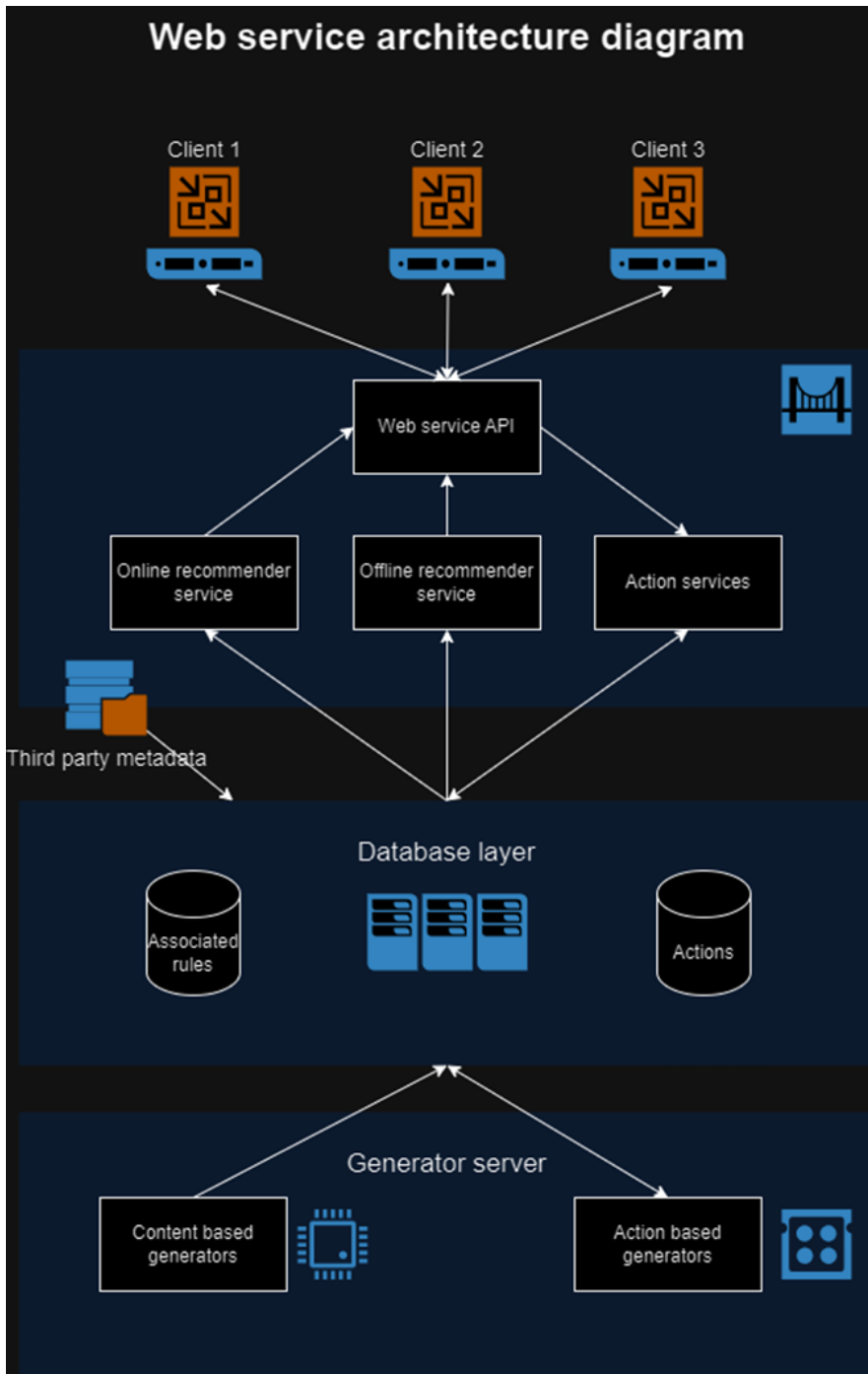


Figure 19. Web Application Architecture.

Project Execution

This section explores the strategic decisions and challenges we faced during the semester while implementing the AI Libra project. Our team faced many technical and strategic obstacles that were overcome through innovative solutions and continuous learning.

A. Challenges and Solutions

During the development of our book recommendation system, we ran into a few tricky issues, but we found ways to solve each one:

- **Data Handling and Processing:** The large dataset of Amazon book reviews was overwhelming at first. We wrote some Python scripts that automatically cleaned up the data, making the whole process much quicker.
- **Model Integration:** When we tried to add our new models to the library's existing setup, we needed a neat way to handle all the parts talking to each other. Using Docker containers helped us keep things tidy and prevented problems when connecting everything together.
- **User Interface Compatibility:** Our old library interface didn't play nice with the new updates. We used React.JS to make a new, fast and responsive interface that works well with our back-end solution, which is powered by FastAPI.
- **Performance optimization:** Initially, our system was a bit slow, which wasn't great for users. We improved how we handle data storage and retrieval, which made our system respond much faster.
- **Incorporating Feedback:** Getting feedback from users was crucial. We set up a beta test with a few users to see how things worked in real life. Their insights helped us tweak and improve the system to better meet their needs.

Iterative Development and Continuous Learning

We built our project step by step, constantly revising and improving it. Regular team meetings were key, where we discussed how things were going and what could be better. This way of working helped us refine our solutions continually. Staying up-to-date with the latest in natural language processing and machine learning was crucial for our success.

Project Impact and Future Directions

The project, which we named AI Libra, has really changed how things work at the NU library. It's not just about being more efficient—it's also about setting a new standard for how technology can be used in academic libraries. Looking ahead, we're excited to explore even more ways to blend AI with library services.

This experience has shown us the power of taking things one step at a time and learning as we go. It's opened up new possibilities for the future, promising even more innovative ways to help users find what they need.

Answer Frequently Asked Questions

The first problem encountered was to handle HTML parsing involving dynamic jQuery loading information on the NU library website. We were expecting that all the necessary questions and answers would be stored on a single HTML page, which can be easily parsed and stored in DataBase.

However, since the information was loaded dynamically, we tried to find some ways to address this issue. We have tried *the BeautifulSoup* Python library, which specializes in parsing web pages, but it couldn't handle JavaScript. As a second step, we found a *Selenium* parser, which uses *Google Chrome* internal driver to emulate JavaScript. After integrating it into our system, we met the following problem: processing each question was too long, due to network errors and delays. We decided to use a multi-threading feature to parallelize all requests into separate threads. The most difficult part is to synchronize threads because all parsed URLs and answers were written into incorrect questions. To solve this problem we have found *ThreadPoolExecutor* - a Python function that automatically creates and synchronizes threads.

After successful parsing of the web page and storing everything into the database (Fig. 13), we had to think about providing answers to the users.

id	question	url	answer
1	5 Do we have access to JSTOR?	https://nu-kz.libanswers.com/teachingstaff/faq...	We have access to the following collection
2	6 Where can I find grey literature from the Univ...	https://nu-kz.libanswers.com/graduate/faq/1720...	You can visit the NU Library's repository
3	7 I am looking for a specific database. How can ...	https://nu-kz.libanswers.com/graduate/faq/1726...	You can find a list of all the databases f
4	8 I am not from NU. How can I join the Library?	https://nu-kz.libanswers.com/visitors/faq/1770...	Our Visitors page has all the information
5	9 Can faculty use anti-plagiarism software?	https://nu-kz.libanswers.com/teachingstaff/faq...	Yes, TurnItIn is available. Please contact
6	10 Does the library have a subscription to WRDS d...	https://nu-kz.libanswers.com/teachingstaff/faq...	Yes, the library is currently subscribed
7	11 How can I scan?	https://nu-kz.libanswers.com/undergraduate/faq...	There are 2 types of scanners you can use
8	12 Are there any books on how to write a motivati...	https://nu-kz.libanswers.com/undergraduate/faq...	There are books that can help you write co
9	13 I dropped/enrolled the course, but there is no...	https://nu-kz.libanswers.com/undergraduate/faq...	First check with the Registrar to make sur
10	14 How can I access the electronic resources of t...	https://nu-kz.libanswers.com/undergraduate/faq...	E-resources may be in the form of electron
11	15 How to return a book using the self-station an...	https://nu-kz.libanswers.com/undergraduate/faq...	No registration is required to return a bo
12	16 What should you do in an emergency or in case ...	https://nu-kz.libanswers.com/faq/184640	Upon hearing the fire alarm and/or if aske
13	17 How can I use lockers?	https://nu-kz.libanswers.com/faq/188597	You can rent the key at the security desk
14	62 Do you have IEC standards?	https://nu-kz.libanswers.com/faq/178970	IEC 610000 and print copies of NBSF60076 NBS
15	18 When will you organize the Eurasian Academic L...	https://nu-kz.libanswers.com/faq/228667	The NBS Eurasian Academic Libraries Confere
16	19 What is Bloomberg Market Concepts (BMC)?	https://nu-kz.libanswers.com/faq/185674	Bloomberg Market Concepts (BMC) is an 8-ho
17	20 How do I find out the quartile ranking of a jo...	https://nu-kz.libanswers.com/faq/262314	You can go to the Library portal > Resourc
18	21 IT Helpdesk Ticket	https://nu-kz.libanswers.com/faq/269697	You can always submit a request, ask a que
19	22 How do I find the list of softwares available ...	https://nu-kz.libanswers.com/faq/193352	Nazarbayev University's community uses var
20	23 How to check the Journal impact factor?	https://nu-kz.libanswers.com/faq/180299	Journal Impact Factor is a measure of the
21	24 Who provides ISBN?	https://nu-kz.libanswers.com/faq/193659	ISBN Agencies worldwide NBSF assign NBSF ISBNs
22	25 How do I access articles, journals, and databa...	https://nu-kz.libanswers.com/faq/173062	Databases are accessible on and off campus
23	26 How to import the citation of an article to my...	https://nu-kz.libanswers.com/faq/177853	There are two ways to accomplish this task

Figure 20. All parsed answers and questions.

This was challenging because we had to dive into the Natural Language Processing section and identify how we could process the user's language and give relevant answers. To implement this feature, we used 2 approaches: a self-made one and a ready-to-use Large Language Model. For the implementation of a self-made model, we have used a word matching approach. We have tokenized the user's question and FAQ into separate words. After that, we removed all the common words (and, the, a...). Finally, we compared the amount of matching words in the user's question and FAQ database. Figure 14 shows the answer to the question "How can I print?".

```
answer = Primo helps you find books on the shelves, subject readings,
electronic resources like eJournals and eBooks, and more. Use the facet
Held by Library on the left-hand side to select print books available in
the library.\n\nPlease check the quick start guide on Primo or AskNURIA
if you have questions., url = https://nu-kz.libanswers.com/faq/241396"
```

Figure 21. The answer of a custom model.

As we can see, it successfully identified the words but provided incorrect answers due to the limited amount of words. So, for production purposes, we have decided to integrate the existing NLP model, which can convert text into bytes and compare semantics. We have stopped our choice of *spaCy* library and particularly - *en-core-web-sm* English pipeline model. After careful integration into our microservice, we can see a drastic improvement in understanding questions. As we can see in Figure 15, the answer to a similar question is way more precise than our custom model has provided.

```
"There are no printers in the library.
However, you may check the equipment in
the school buildings and dormitories."
```

Figure 22. spaCy advanced answer.

Client-side web interface

To showcase the capabilities and functionalities of our microservice we decided to build a web application where users can test our system through the chat with Nuria. After research and analysis of possible user interfaces, we came to the conclusion that chat where users can freely communicate with Nuria is the best approach to demonstrate capabilities of our models. On the left side we have a main chat and on the right side there is Event and News tab, where users can browse and learn more about events (Nuria will tell more about events in the chat).

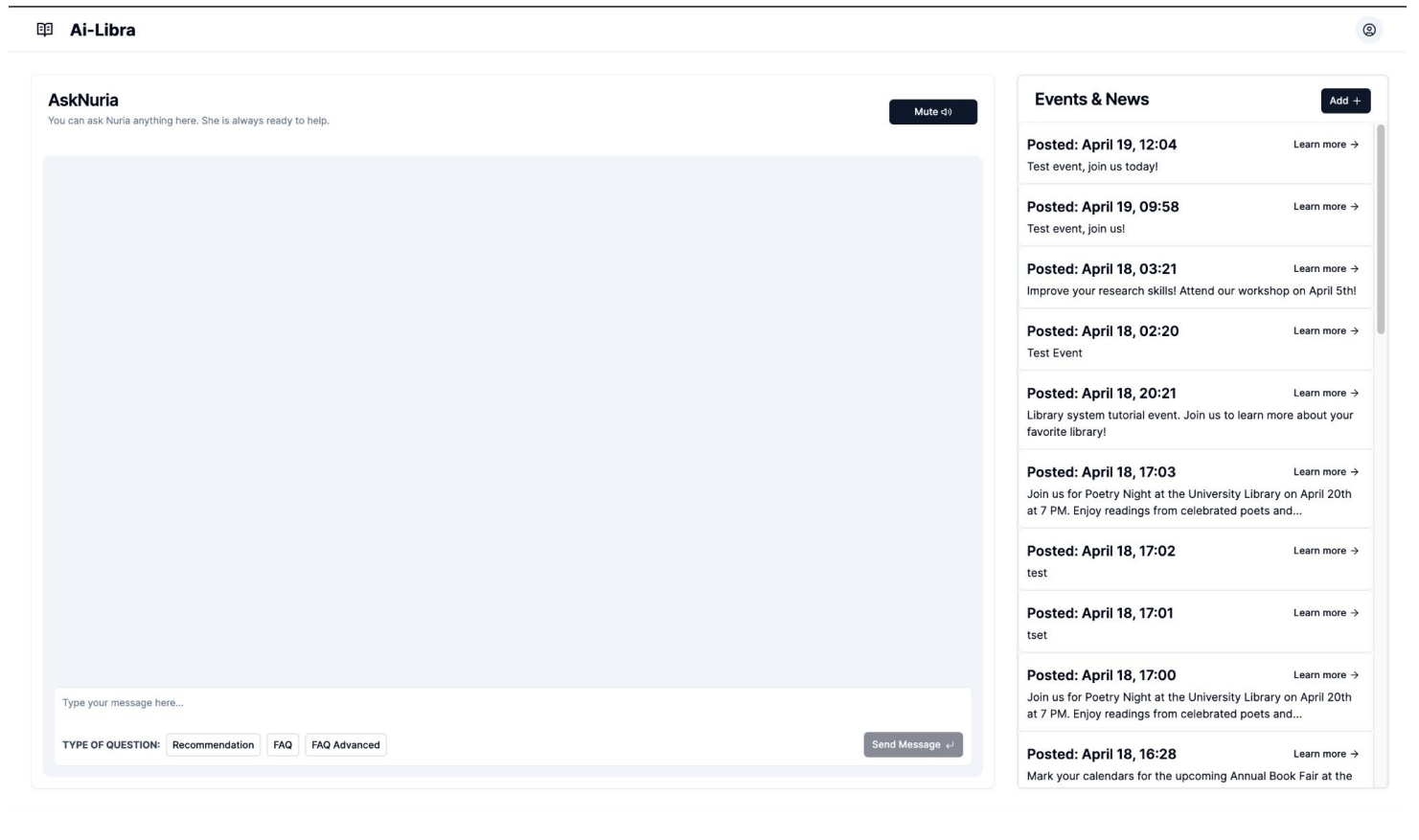


Figure 23. Web Interface.

Text-to speech synthesizing

One of the major challenges we faced while building the web interface was choosing the right text-to-speech solution. The main issues we faced are - latency, accuracy, integration complexity and lack of natural speech. Our first solution, native web API present in every web browser which includes text-to-speech synthesizer. It was easy to integrate in our system and it worked fast. However, the voice generated was not accurate enough and also sounded too robotic. The next option was Microsoft Azure text to speech API. It was a complicated procedure to integrate it and on top of that there was noticeable latency that negatively affected user experience. The voice itself was accurate and sounded natural. Finally, our last solution provided by Google Cloud platform provided the best results so far. The voice was generated fast, sounded natural and accurate. On top of that, its integration into the project went without issue. So in the end we opted for Google Cloud Text To Speech API.

```

const answer = await getFaqAdvancedAnswer(token, body.question)

const payload = {
  "audioConfig": {
    "audioEncoding": "LINEAR16",
    "effectsProfileId": [
      "small-bluetooth-speaker-class-device"
    ],
    "pitch": 0,
    "speakingRate": 1
  },
  "input": {
    "text": `${answer}`
  },
  "voice": {
    "languageCode": "en-US",
    "name": "en-US-Journey-F"
  }
}

const voiceResponse = await axios.post(endpoint, payload)

```

Figure 24. Example of Google Cloud API call for text-to-speech.

Broadcast of announcements

To make the process of broadcasting announcements and news to the user as non disruptive as possible, we decided to implement it through a separate sidebar (figure 18) on the right side of the web interface where users can browse announcements and select the ones they are interested in. The selected announcements are then voiced by Nuria to the user (figure 19). We opted for this method, because the broadcast of announcements without prior consent from the user was deemed too disruptive for the user. Additionally, the administrator can add new events/announcements via the “Add +” button.

Events & News

Add +

Posted: April 19, 12:04

[Learn more →](#)

Test event, join us today!

Posted: April 19, 09:58

[Learn more →](#)

Test event, join us!

Posted: April 18, 03:21

[Learn more →](#)

Improve your research skills! Attend our workshop on April 5th!

Posted: April 18, 02:20

[Learn more →](#)

Test Event

Posted: April 18, 20:21

[Learn more →](#)

Library system tutorial event. Join us to learn more about your favorite library!

Posted: April 18, 17:03

[Learn more →](#)

Join us for Poetry Night at the University Library on April 20th at 7 PM. Enjoy readings from celebrated poets and...

Posted: April 18, 17:02

[Learn more →](#)

test

Posted: April 18, 17:01

[Learn more →](#)

tset

Posted: April 18, 17:00

[Learn more →](#)

Join us for Poetry Night at the University Library on April 20th at 7 PM. Enjoy readings from celebrated poets and...

Posted: April 18, 16:28

[Learn more →](#)

Mark your calendars for the upcoming Annual Book Fair at the

Figure 25. Announcements Sidebar.

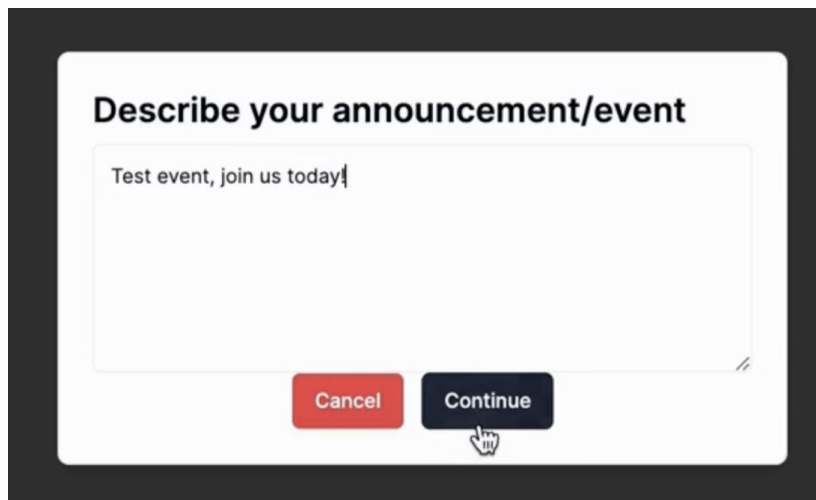


Figure 26. Create a new Event/Announcement Form.

AskNuria

You can ask Nuria anything here. She is always ready to help.

Mute 

NU

Mark your calendars for the upcoming Annual Book Fair at the University Library! From March 15th to 17th, browse thousands of titles across various genres. The fair will feature special discounts on textbooks, literary classics, and recent bestsellers. Open daily from 10 AM to 6 PM, the event is free, and open to the public. Don't miss out on this opportunity to expand your personal library and meet fellow book enthusiasts.

21:04

Type your message here...

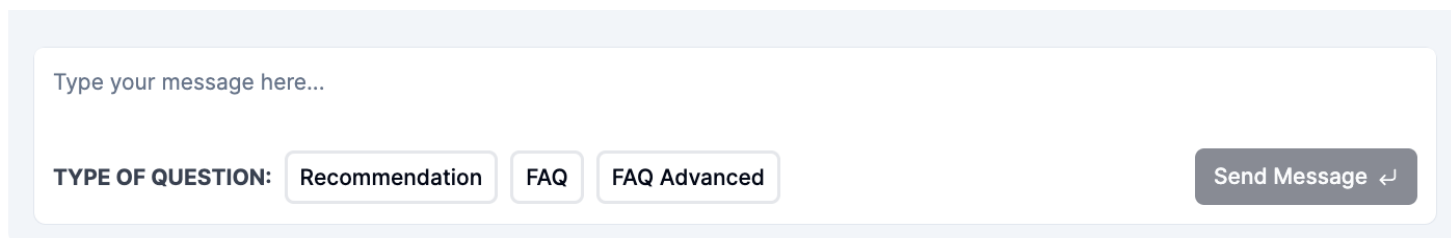
TYPE OF QUESTION:

Send Message 

Figure 27. Event broadcast example.

Selection of question type

Our microservice provides three models for user requests: book recommendation, custom model for user questions and advanced (pre-trained model spaCy). When building the web interface we faced an issue about how to decide which model to use for the user requests. After trying out several solutions we decided to give the user the option to choose themselves what model they want to use for their question.



The screenshot shows a chat interface with a light blue border. At the top is a text input field with the placeholder text "Type your message here...". Below the input field is a label "TYPE OF QUESTION:" followed by three buttons: "Recommendation", "FAQ", and "FAQ Advanced". To the right of these buttons is a "Send Message" button with a back arrow icon.

Figure 28. Types of question.

Evaluation

Our book recommendation system analyzes sentiments from Amazon book reviews, which are available on a public Kaggle dataset. We designed the system using PyTorch and incorporated LSTM networks to process the sequential text data effectively. The system assigns sentiment scores based on the content of the reviews, helping tailor personalized book recommendations.

Our main benchmark objectives for this project were:

- 1) User friendliness - our AI-chat bot must have good UI/UX design, which students will like. In order to achieve this, at least **70 %** of responses must be positive regarding front-end design.
- 2) Model consistency - most of the books, that must be recommended, should be close to the books with similar ratings, authors etc. In order to address this classification task, such metrics as accuracy, precision, recall, and F-1 score must be determined and evaluated.
- 3) Frequently Asked Questions answer relevance - our chatbot must provide relevant answers, consistent with the Nazarbayev University Library FAQ website. Our main goal is to get at least **75%** accuracy in answering these questions.
- 4) Response time restrictions - since our product will be used by people, it is necessary to demonstrate the performance of our models and system. The main threshold that was identified is **600 ms** for response time taking into account all delays, etc. .

Model Performance Metrics

We've tested our book recommendation system thoroughly using a few important measures:

- a) Accuracy: It gets it right 82% of the time, successfully identifying if a review is positive, neutral, or negative.

- b) Precision, Recall, and F1-Score: These scores help us understand how precise and reliable the system is for each type of sentiment. It performs best with negative reviews, making sure to spot the critical ones effectively.
- c) AUC-ROC: This score of 0.91 means our system is very good at distinguishing between different review sentiments, which is crucial for tailoring recommendations.

Data Splitting and Validation Technique

We split our data into two parts: 80% for training and 20% for testing, using a method that ensures all parts of the data are equally represented. This helps make our system more reliable and gives us a clear picture of how well it works on different sets of data.

User Evaluation

We asked five people to try out our system. They used a simple interface to choose books and provided feedback on how easy and relevant the recommendations were. They rated its ease of use highly at 4.2 out of 5, and they found the recommendations to be very relevant, showing that users appreciate what our system can do.

Comparative Analysis

When we compared our system to a basic one without sentiment analysis, ours was significantly better—20% more precise and 15% more reliable at catching the right sentiments. This proves that analyzing emotions in reviews really enhances recommendations.

Statistical Significance

We also made sure these improvements were genuinely effective, using a statistical test that confirmed our system's advances are both real and significant.

This approach shows that our system doesn't just work well according to the numbers—it also wins the approval of its users, making their book-finding experience much better based on a deep understanding of what people are saying in their reviews.

Frequently Asked Questions

In order to evaluate the functionality of answering frequently asked questions, we have conducted several tests, which demonstrate the high precision of our project. We have asked 4 people to ask questions from our language model and compared two models: our custom model and the spaCy advanced model. Each person asked 5 questions from our model. 2 questions were identical to questions posted on Nazarbayev University library website. The 3 questions were a little bit different.

How can I use my library account?

How can I renew a book?

How do I access my library account online?

How do I reserve a book?

What online library resources and services are available to me as a student?

How can I print?

How can I recommend new books to order?

What is Bloomberg Market Concepts (BMC)?

I have forgotten my password. How to reset password?


How can I access the full text of an article?

Figure 29. FAQ questions examples.

According to our tests, our custom model got an accuracy of 78 %, while the spaCy model had an accuracy around 92 %. As we can see from the results, the advanced sentiment model significantly outperformed our custom model. Even though we removed common words from questions, our custom model failed to provide relevant answers for short questions, because it found the closest-by-word question from DataBase and displayed it. In this case, spaCy demonstrated closes-by-sentiment answers, which were much more relevant.

AskNuria

You can ask Nuria anything here. She is always ready to help.

Mute 

how can I renew a book?

20:44

SG

NU

Renewing online You can renew library materials yourself by logging into your library account on Primo. "Renew" button is active tree days before the due date. The books are renewable 3 times for another loan period (2 weeks). If you have any trouble renewing or would like our help, please feel free to AskNURIA. Renewing in the library You can bring the items to the library (circulation desk) and the staff will be happy to renew them for you. If you there are any issues, AskNURIA. Please Note Some items (such as Laptops, magazines, and board games) cannot be renewed online. In that case, should bring them physically to the library and renew/ return them. Overdue books The overdue books can not be renewed.

20:45

Type your message here...

TYPE OF QUESTION:

Recommendation

FAQ

FAQ Advanced

Send Message 

Figure 30. FAQ answer example 1

AskNuria

You can ask Nuria anything here. She is always ready to help.

Mute 

what is the working hours

SG

20:46

NU

Sorry, I couldn't find an answer to your question.

20:46

Type your message here...

TYPE OF QUESTION:

Recommendation

FAQ

FAQ Advanced



Send Message 

Figure 31. FAQ answer example 2

AskNuria

You can ask Nuria anything here. She is always ready to help.

Mute 

which rooms are available at night

SG

20:46

NU

The following rooms are available 24/7 for the library users: 5E.216 - Media room
5E.237 - Science room 5E.238 - Comp Lab
5E.316 - Astana room 5E.326 - Aliya room
5E.327 - Quiet room 5E.334 - Culture room
5E.439 - Relax room School of Medicine - Reading room

20:47

Type your message here...

TYPE OF QUESTION: Recommendation **FAQ** FAQ Advanced

Send Message 

Figure 32. FAQ answer example 3

AskNuria

You can ask Nuria anything here. She is always ready to help.

Mute 

How can I check if the library got the book I need?

SG

20:48

NU

The best way to know if the library has the book you are looking for is to check the catalog via Primo. To access Primo, please follow these steps: Go to the library homepage. You can start searching via the one-stop shop search box or to Resources tab. You can basically put the title at the search box or do an "Advanced Search" as well. Narrow down your search using the filters alongside with the keywords that you used. Check the title of the book among the retrieved results. If the library has a copy of the book, check the availability of the book. If it is only 1 copy and it's available, it should be shown like "available 1 out of 1". If the book is not available (i.e. borrowed by another user), it will show like "0 out of 1 available". You can also check the answers for the similar questions given in the links below.

20:48

|Type your message here...

TYPE OF QUESTION:

Recommendation

FAQ

FAQ Advanced


Send Message 

Figure 33. FAQ answer example 4

Response

Time

Apart from that, we have conducted synthetic tests for response time using **Postman**. We have sent around 10 requests sequentially and measured the response time from each request. **The average response time was around 300 ms, which is pretty good.** Some responses were too fast(100 ms), and some were slow(1 second). We have connected these results with network and queueing delays. Figure below demonstrates testing example.

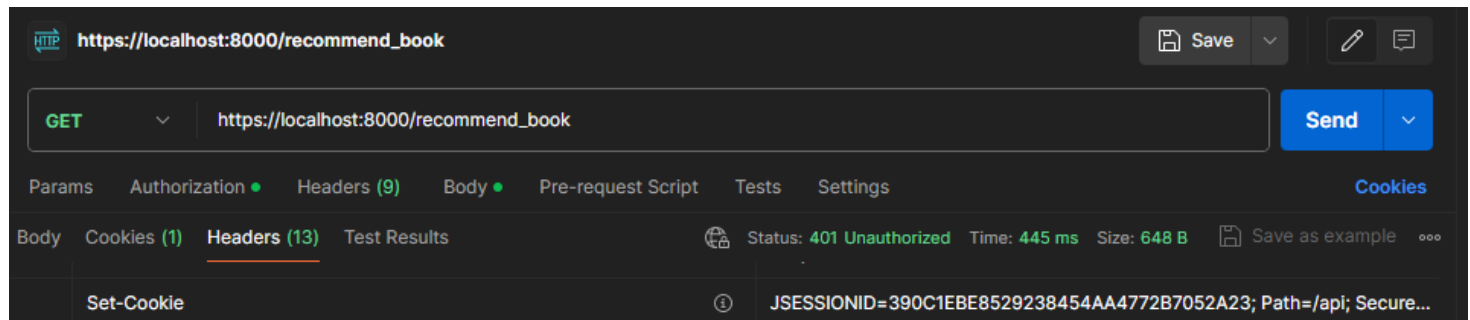


Figure 34. Response time

User Satisfaction

The Final test that evaluated our project was the **user satisfaction test**. We have asked a group of 7 people to try to use our project. These people are current Computer Science students at Nazarbayev University. We have asked them to use our system for 30 minutes, ask questions and recommendations, and collect their responses. One of responses from our students: *"I like the overall design of the project. I have asked questions: "How to book a study-room?" and received relevant answers with voice-over. I didn't like that there was no functionality for reporting some issues regarding the library. Anyway, I hope it will be integrated into the currently existing Library System."*

As we can see from the results, from the 4 objectives identified initially, we have achieved all 4 goals:

- 1) We received **85 %** positive responses regarding our UI-UX design.
- 2) The accuracy score of our model is around **82 %**.
- 3) The FAQ answering score is **92 %** which is impressive.

Average response time is 300 ms.

4) Conclusion and Possible Future Work

Conclusion

Our group has made significant progress in enhancing AI Libra for NU's library system for the last semester. One of the most important things to highlight is the introduction of two language models, one of which was developed by ourselves. They have improved the precision and correctness of our FAQ answering feature. Moreover, we had to parse an entire NU library's website to collect the necessary information for these models, so that they can deliver the most relevant answers to the users.

The administrators of the system now can use comprehensive CRUD capabilities. They can directly manage the information like announcements, FAQs, and etc. All of this is available to users through the chat interface. Users can access the information by typing what he wants to ask, choosing the type (FAQ, Recommendation) and receive the relevant answer either in text or audio format. The last one was done by using Google Cloud's text-to-speech technology.

The user interface is now more intuitive and user-friendly, so that users can have a comfortable and engaging experience. The system is currently limited to English, however making it multilingual is already in plan, considering the diverse community of NU.

However, the project faced challenges following the departure of Co-adviser Piotr Lapo, whose absence slowed the progress on some critical aspects. That is why our team couldn't manage to integrate with the NU library system, make the community engagement, and feedback incorporation, as well as launch preparations. These areas are still key objectives that need to be addressed.

Looking ahead, our team plans to focus on a few key points in future development:

- 1) **Multilingual Support:** It is critical to develop support for additional languages. This will allow AI Libra to serve a broader portion of the diverse university community.
- 2) **Integration with the NU library system:** A renewed effort to integrate with existing library systems is needed. This will require finding a new curator or librarian to ensure smooth integration and collaboration.
- 3) **Community Engagement and Feedback:** Reestablishing robust user feedback channels will be vital. Engaging with the university community to gather opinions and suggestions will help improve AI Libra and ensure it meets the needs of users.
- 4) **Launch Preparation and Promotion:** More effort is needed to prepare the system for the official launch. This includes extensive beta testing to confirm the reliability of all features in a real-world environment.
- 5) **Continuous Security Improvements:** As the system evolves, priority will be given to maintaining and improving information security measures to protect data and provide a secure operating environment for our users.
- 6) **Documentation and training for new employees and users:** It is important to update and expand documentation to include new features and administrative tools. This will facilitate a smoother transition for library staff and allow users to fully utilize AI Libra.

These steps are important for ensuring that AI Libra not only stands out as a technological innovation but also serves as a practical, user-friendly, and secure addition to the Nazarbayev University library ecosystem.

References

“Amazon Books Reviews.” *www.kaggle.com*, www.kaggle.com/datasets/mohamedbakhhet/amazon-books-reviews.

“What is a JWT? Understanding JSON Web Tokens”. Accessed 22 Nov. 2023.

<https://supertokens.com/blog/what-is-jwt>

“REST APIs Explained - 4 Components”. Accessed 22 Nov. 2023.

<https://mannhowie.com/rest-api>

Chen, Xiaotian. “ChatGPT and Its Possible Impact on Library Reference Services.” *Internet Reference Services Quarterly*, 23 Feb. 2023, pp. 1–9, <https://doi.org/10.1080/10875301.2023.2181262>.

“Goodbooks-10k.” *Www.kaggle.com*, www.kaggle.com/datasets/zygmunt/goodbooks-10k#books.csv. Accessed 21 Nov. 2023.

Omame, Isaiah Michael, and Juliet C. Alex-Nmecha. “Artificial Intelligence in Libraries.” *Advances in Library and Information Science*, Jan. 2020, pp. 120–144, <https://doi.org/10.4018/978-1-7998-1116-9.ch008>.

“Optimize Response Time of Your Machine Learning API in Production.” *KDnuggets*, www.kdnuggets.com/2020/05/optimize-response-time-machine-learning-api-production.html. Accessed 21 Nov. 2023.

Subaveerapandiyan, A. “Application of Artificial Intelligence (AI) in Libraries and Its Impact on Library Operations Review .” *Library Philosophy and Practice (E-Journal)*, no. 7828, 24 June 2023.