ДОДАТОК А

ГРАФІЧНИЙ МАТЕРІАЛ АТЕСТАЦІЙНОЇ РОБОТИ

ГЮИК.506100.009 С10

(позначення документа)

ЗАТВЕРДЖЕНО

ГЮИК.506100.009 С10 – ЛУ(ПС)

Розробка мікро-сервісної архітектури систем пошуку спеціалізованої літератури

Графічний матеріал

ГЮИК.506100.009 С10– ЛУ(ПС)

АРКУШІВ 14

2021 р.Харківський Національний Університет Радіоелектроніки

«ЗАТВЕРДЖУЮ»
керівник атестаційної
роботи проф. Ситніков Д.Е.

Розробка мікро-сервісної архітектури систем пошуку спеціалізованої літератури

Графічний матеріал

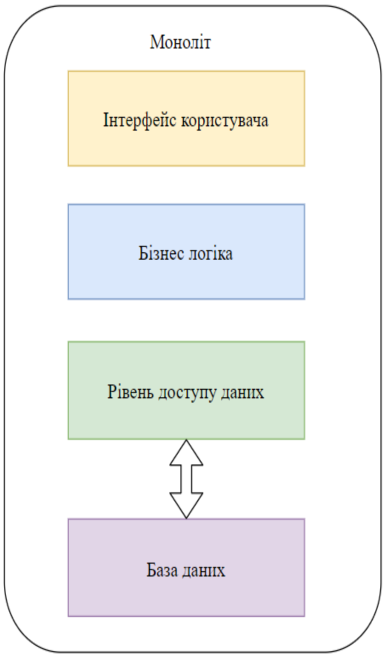
ЛИСТ ЗАТВЕРДЖЕННЯ

ГЮИК.506100.009 С10 – ЛУ(ПС)

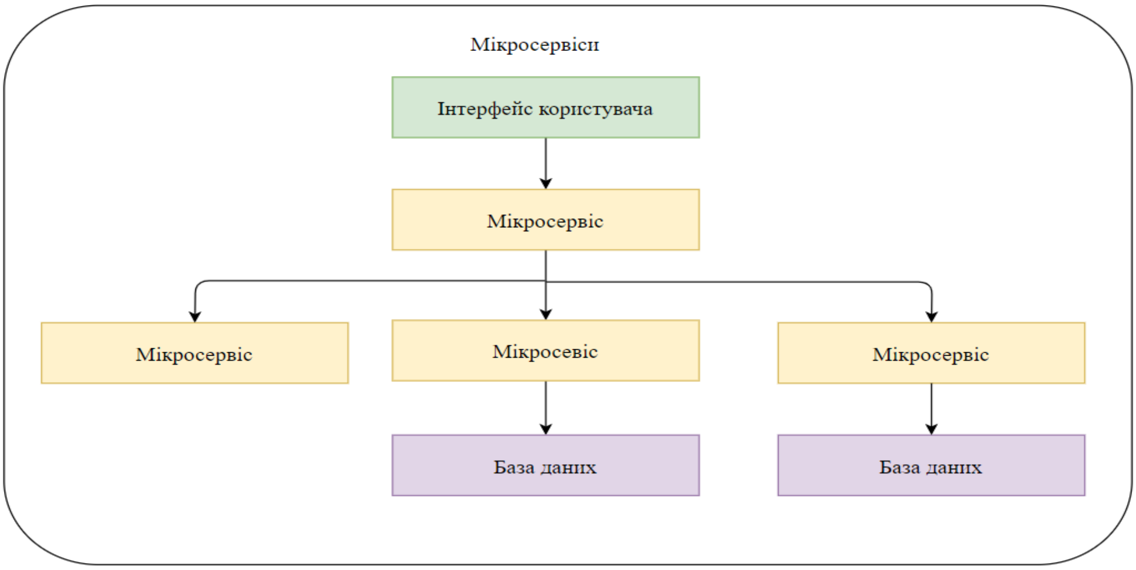
Розробив:
ст.гр. СПРм-19-2
Говдерчак О.П.

2021 р.

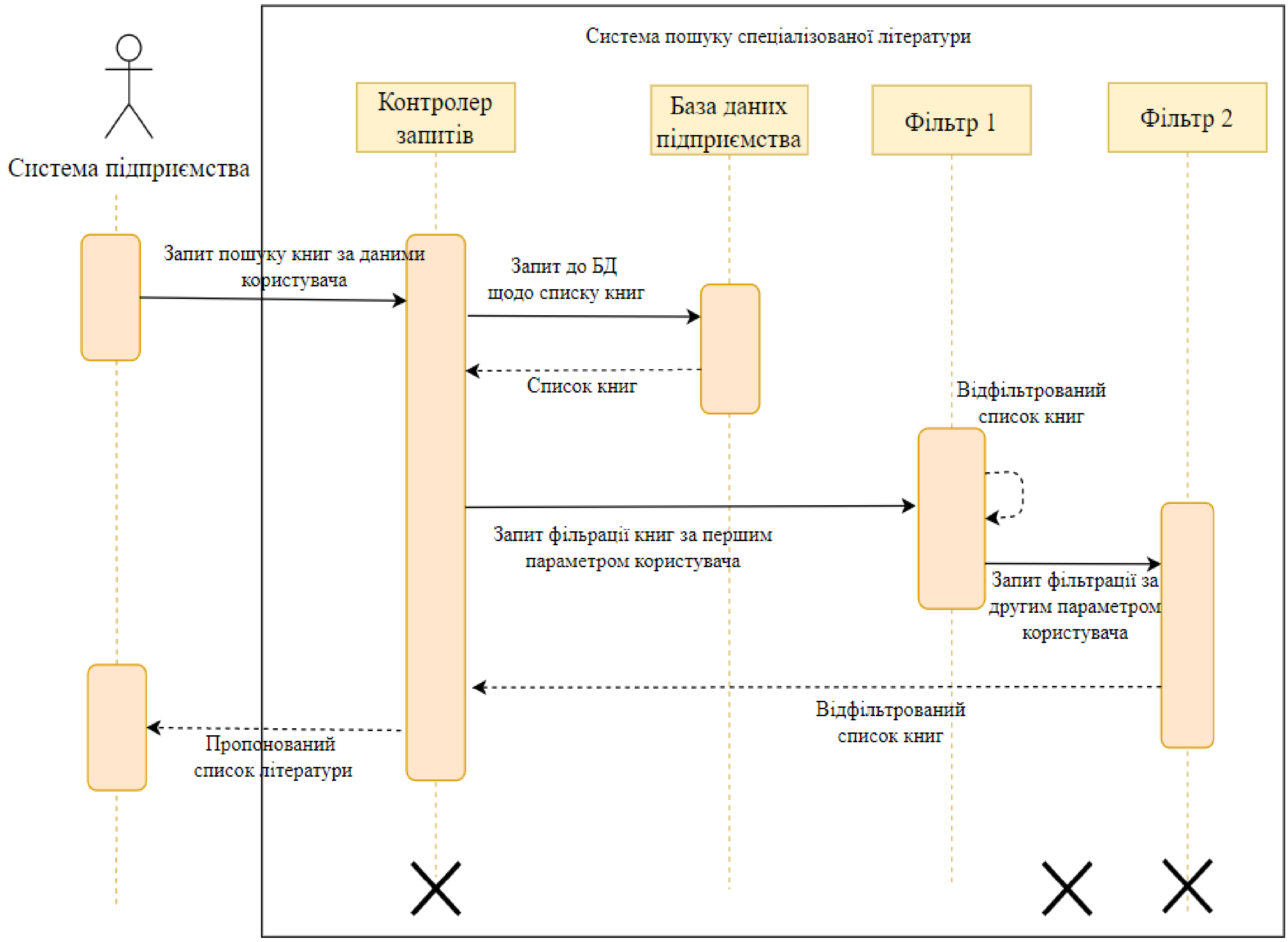
Принципи монолітної та мікросервісної архітектури



Розробив	*Говдерчак О.П.*			*Принципи монолітної та мікросервісної архітектури*		
Перевірив	*Ситніков Д.Е.*					
Н. контр.	*Ситніков Д.Е.*					
				СПРм-19-2	*Аркуш*	*1*
Затвердив	*Гребеннік І.В.*			*СТ*	*Аркушів*	*14*

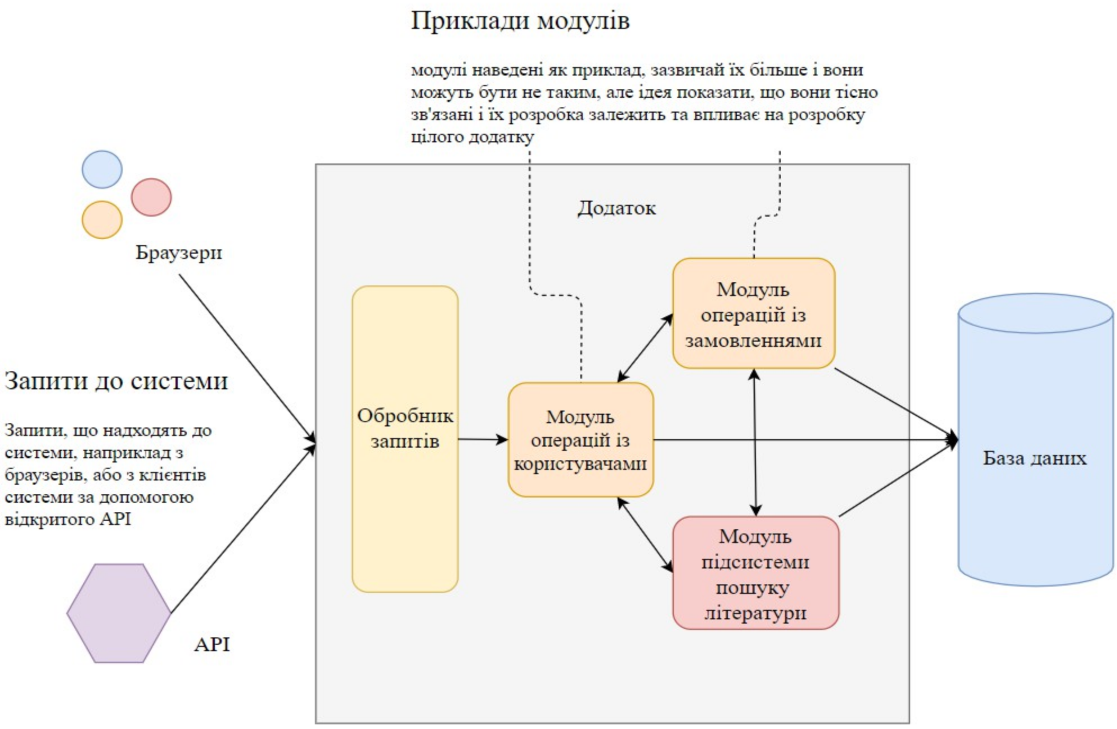


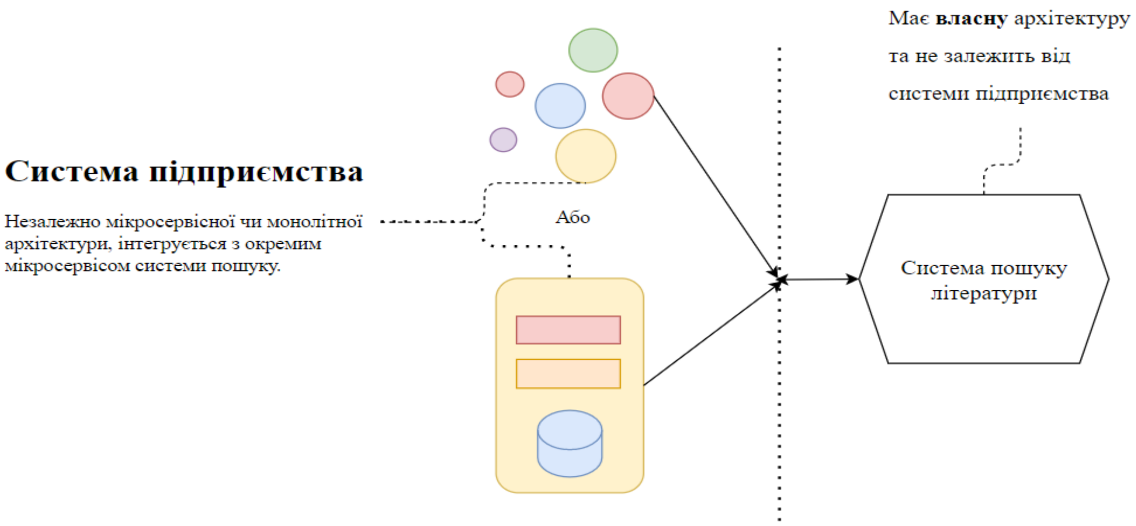
Діаграма послідовності системи пошуку спеціалізованої літератури



Розробив	*Говдерчак О.П.*			*Діаграма послідовності системи пошуку спеціалізованої літератури*
Перевірив	*Ситніков Д.Е.*			
Н. контр.	*Ситніков Д.Е.*			
				СПРм-19-2 — *Аркуш 2*
Затвердив	*Гребеннік І.В.*			*СТ* — *Аркушів 14*

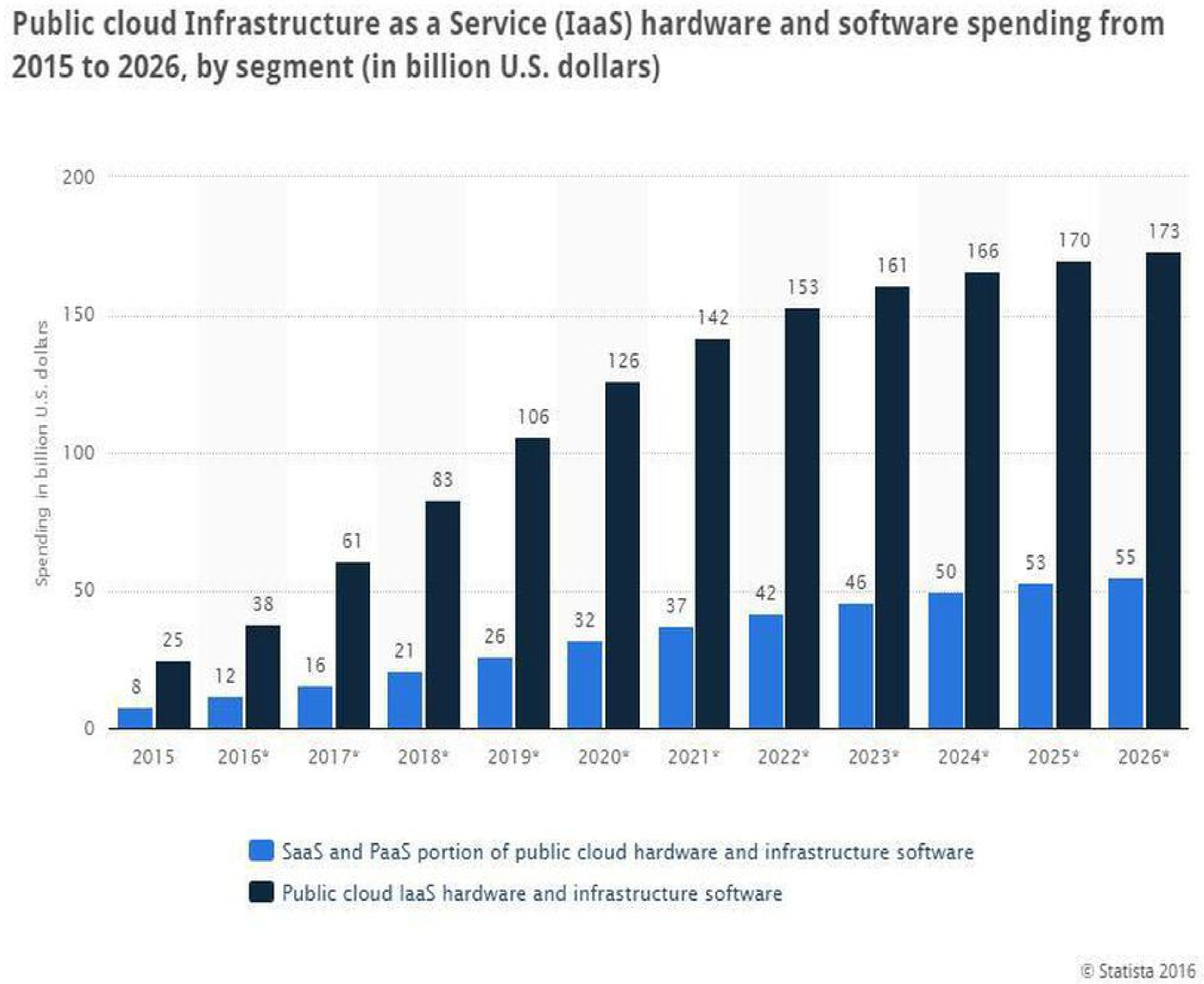
Система підприємства монолітного зразка з системою пошуку всередині та коли система пошуку – це окремий мікросервіс





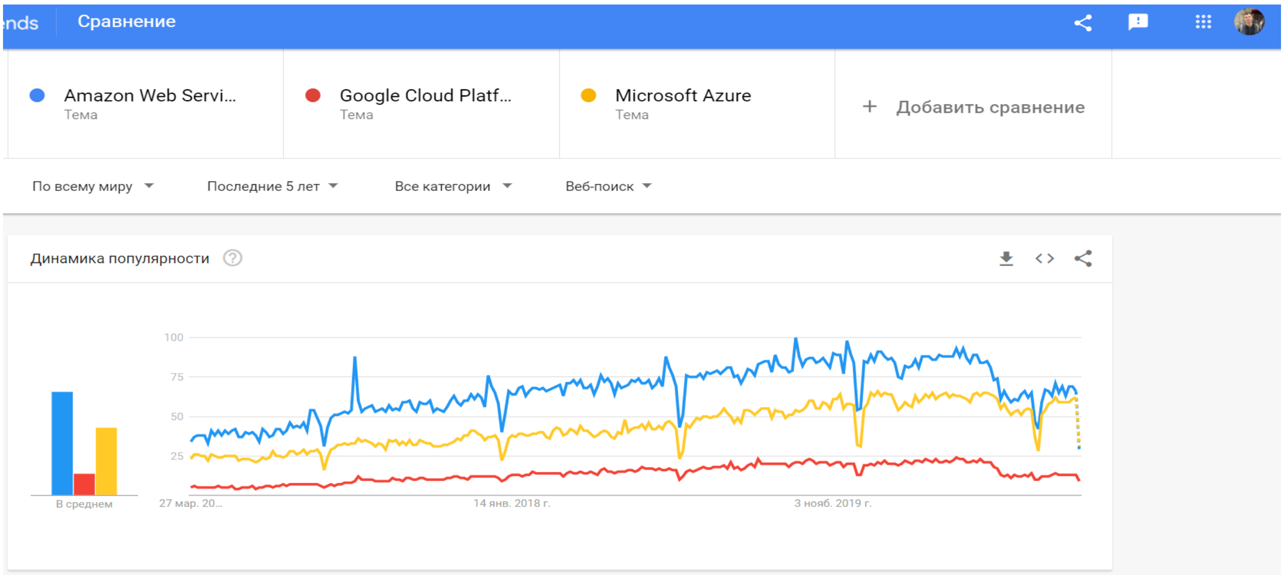
Розробив	*Говдерчак О.П.*			*Система підприємства монолітного зразка з системою пошуку всередині та коли система пошуку – це окремий мікросервіс*		
Перевірив	*Ситніков Д.Е.*					
Н. контр.	*Ситніков Д.Е.*					
				СПРм-19-2	*Аркуш*	*3*
Затвердив	*Гребеннік І.В.*			*СТ*	*Аркушів*	*14*

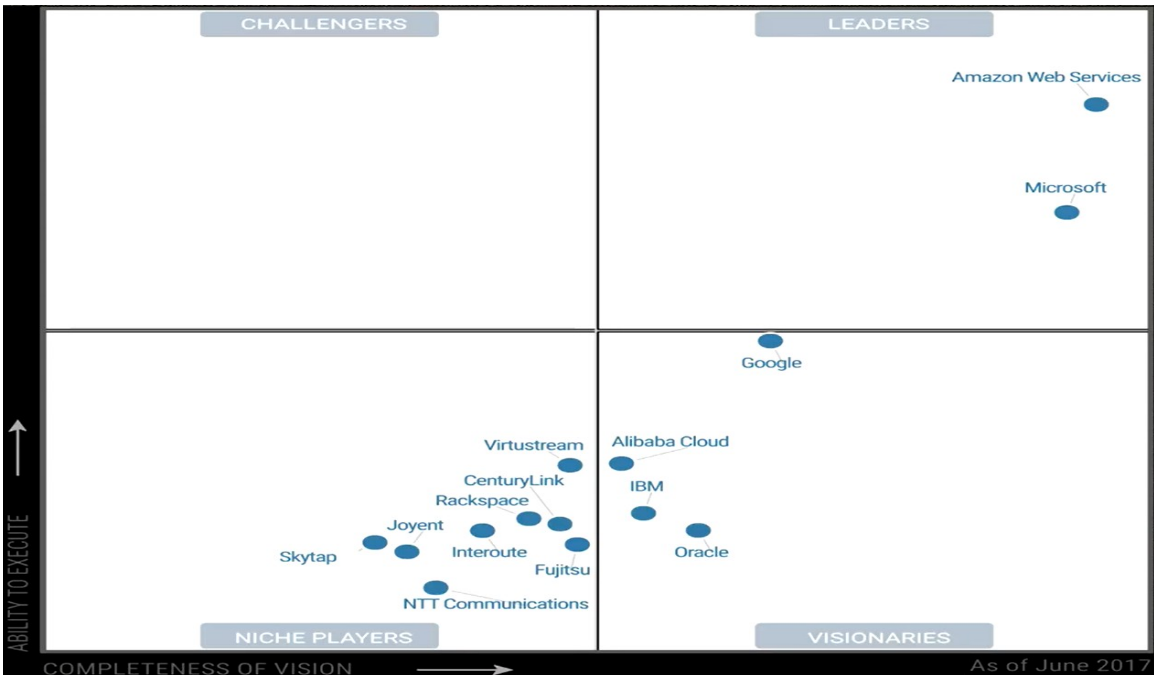
Популярність використання хмарних технологій та прогнозування прибутку провацдерів хмарних обчислень



Розробив	Говдерчак О.П.			Популярність використання хмарних технологій та прогнозування прибутку провацдерів хмарних обчислень	
Перевірив	Ситніков Д.Е.				
Н. контр.	Ситніков Д.Е.				
				СПРм-19-2	Аркуш 4
Затвердив	Гребеннік І.В.			СТ	Аркушів 14

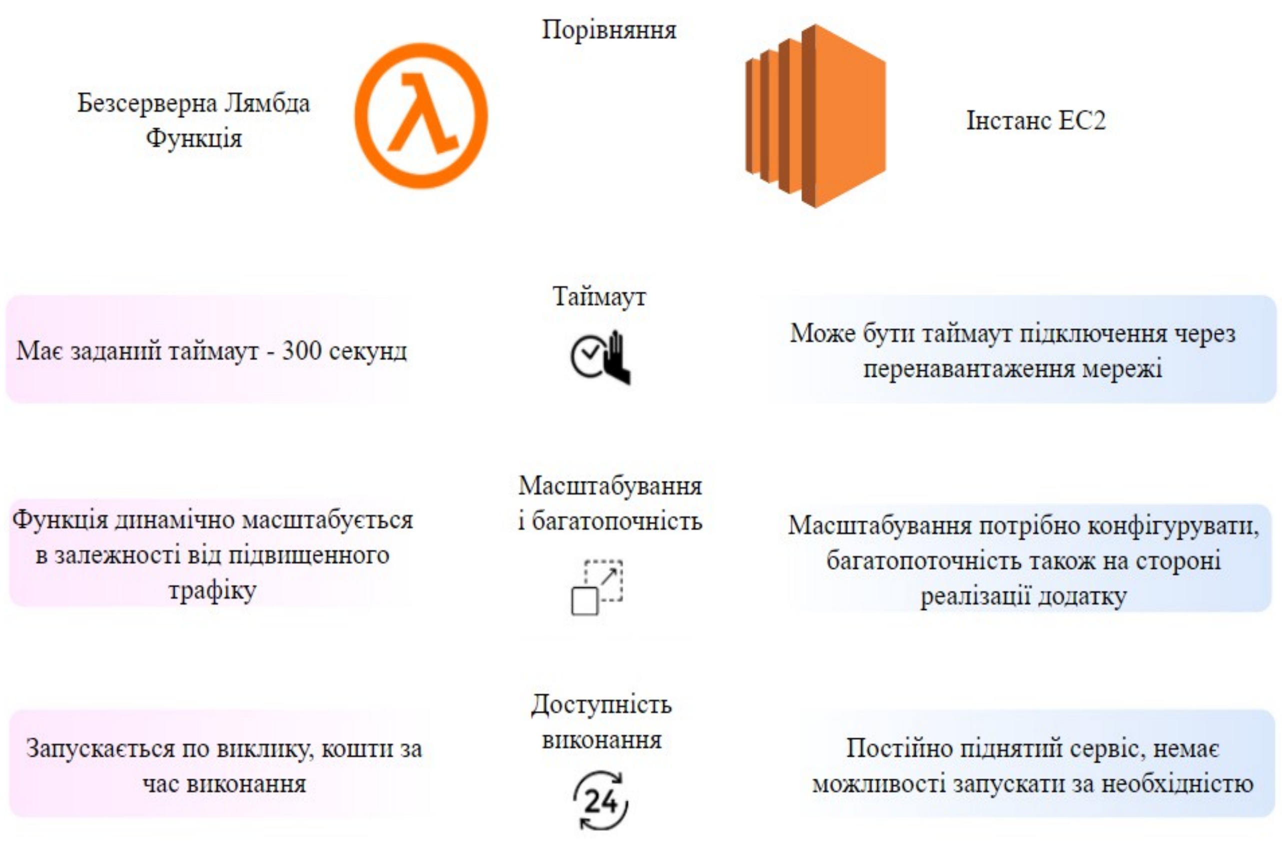
Популярність Amazon серед інших постачальників хмарних послуг





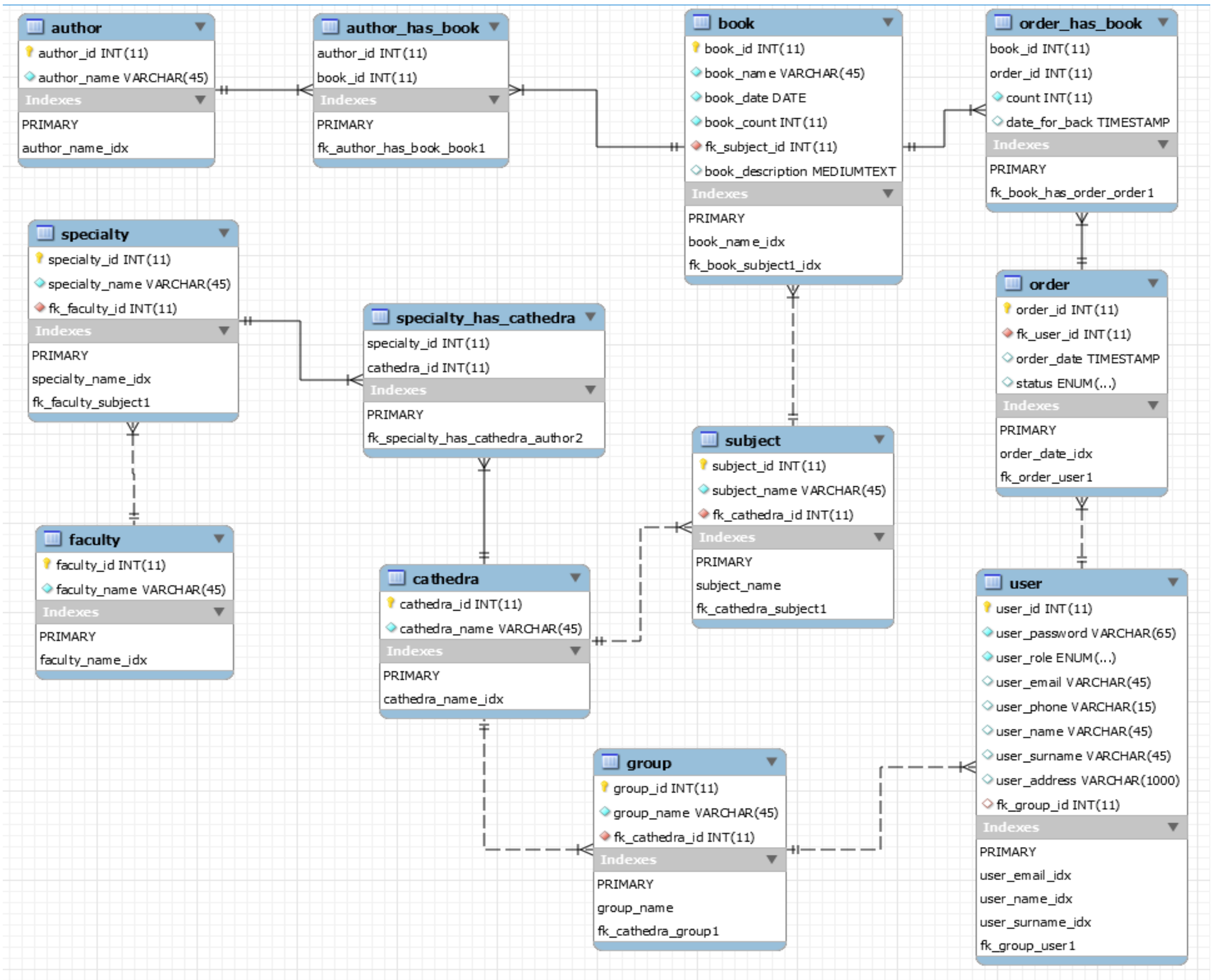
Розробив	Говдерчак О.П.			Популярність Amazon серед інших постачальників хмарних послуг	
Перевірив	Ситніков Д.Е.				
Н. контр.	Ситніков Д.Е.				
				СПРм-19-2	Аркуш 5
Затвердив	Гребеннік І.В.			СТ	Аркушів 14

Порівняння безсерверних функцій Амазону та його віртуального серверу



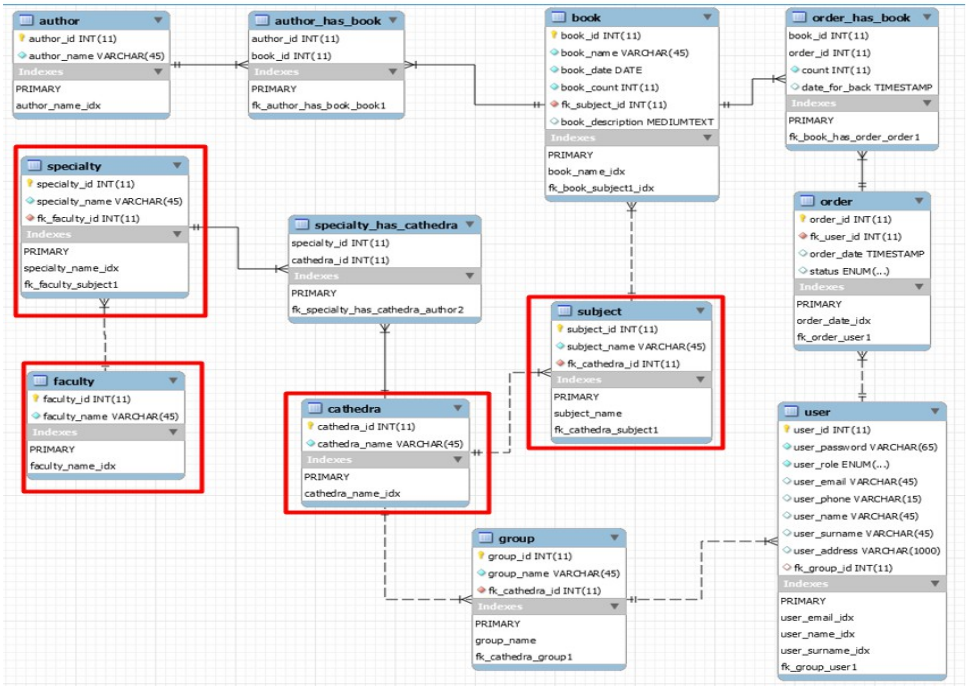
Розробив	*Говдерчак О.П.*			*Порівняння безсерверних функцій Амазону та його віртуального серверу*	
Перевірив	*Ситніков Д.Е.*				
Н. контр.	*Ситніков Д.Е.*				
				СПРм-19-2	*Аркуш 6*
Затвердив	*Гребеннік І.В.*			*СТ*	*Аркушів 14*

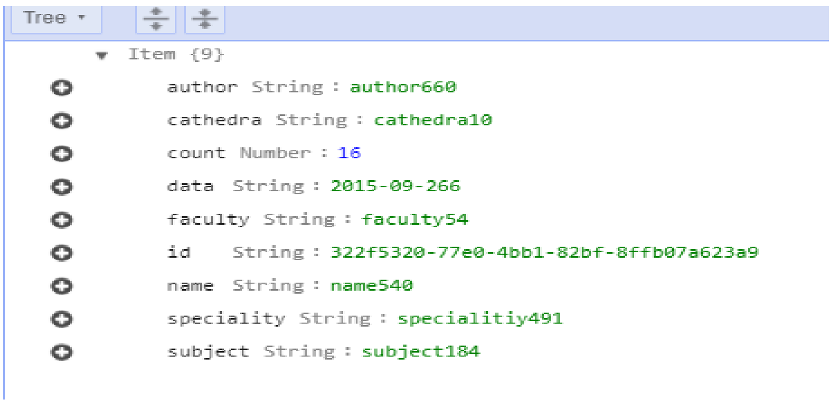
Фізична модель бази даних прикладу підприємства



Розробив	*Говдерчак О.П.*			*Фізична модель бази даних прикладу підприємства*		
Перевірив	*Ситніков Д.Е.*					
Н. контр.	*Ситніков Д.Е.*					
				СПРм-19-2	*Аркуш*	*7*
Затвердив	*Гребеннік І.В.*			*СТ*	*Аркушів*	*14*

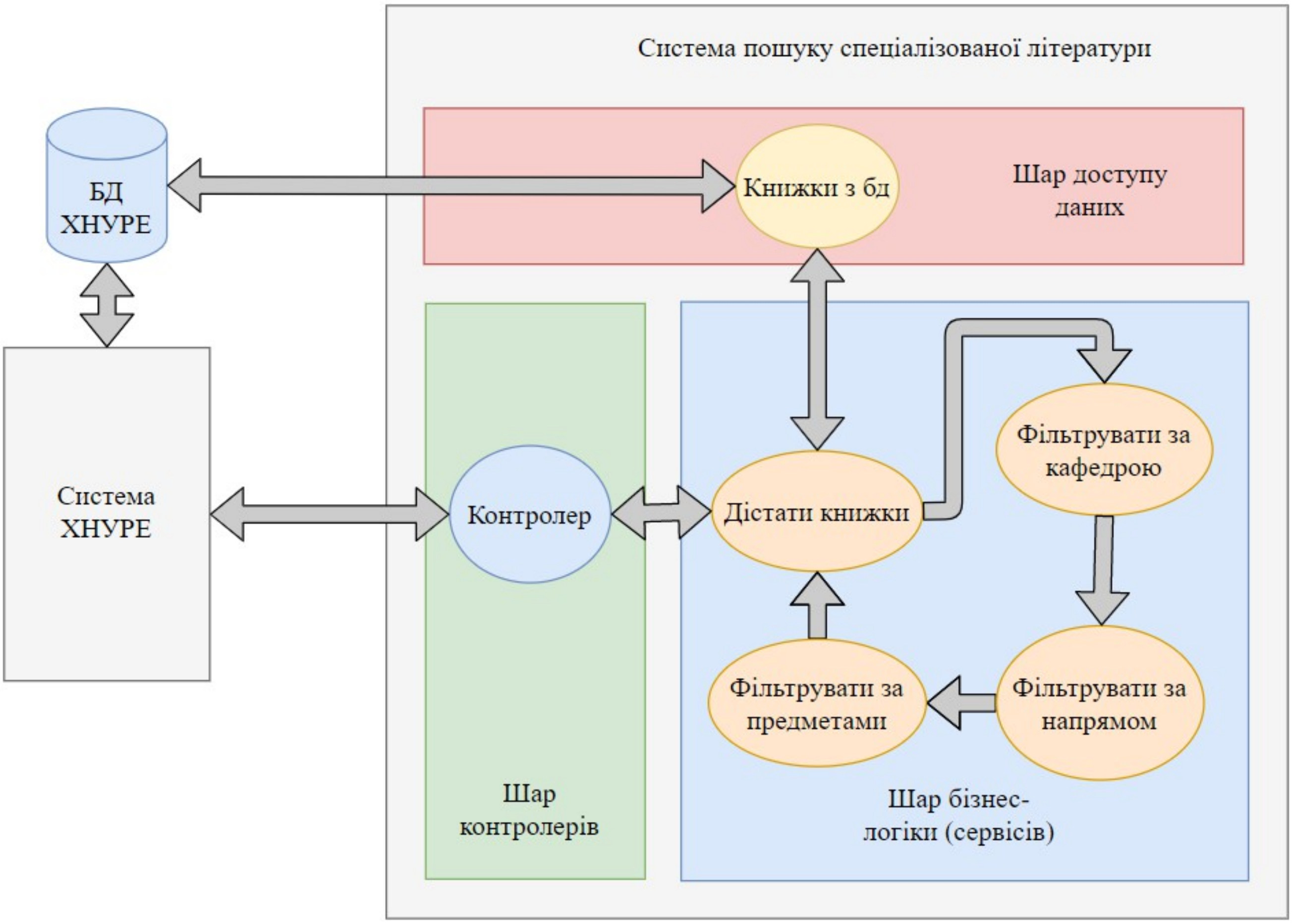
Необхідні таблиці для системи пошуку та створення прикладу реплікації їх та міграції в єдину DynamoDB таблицю





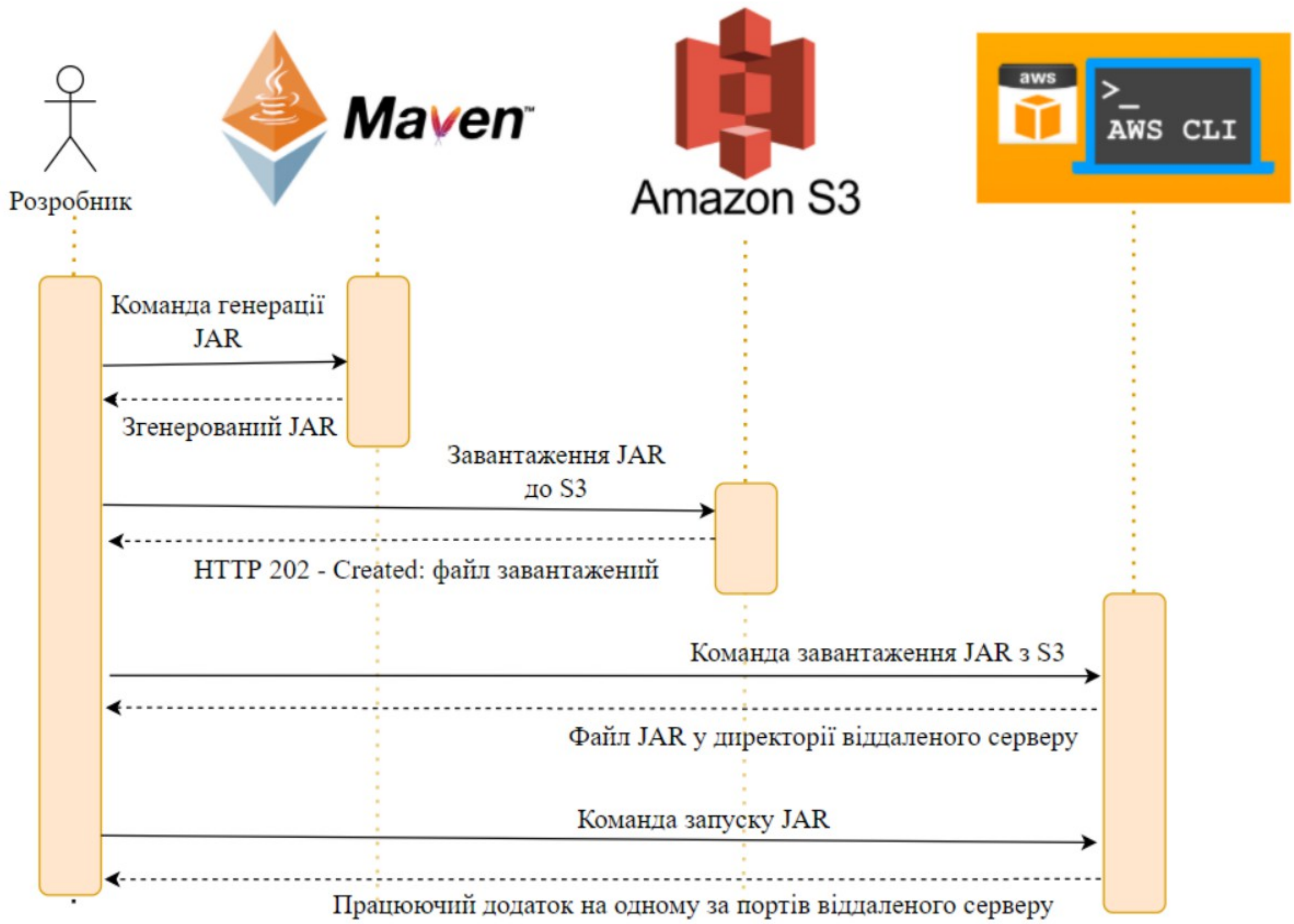
Розробив	*Говдерчак О.П.*			*Необхідні таблиці для системи пошуку та створення прикладу реплікації їх та міграції в єдину DynamoDB таблицю*		
Перевірив	*Ситніков Д.Е.*					
Н. контр.	*Ситніков Д.Е.*					
				СПРм-19-2	*Аркуш*	*8*
Затвердив	*Гребеннік І.В.*			*СТ*	*Аркушів*	*14*

Взаємодія системи підприємства (ХНУРЕ) із системою пошуку як окремим суцільним сервісом



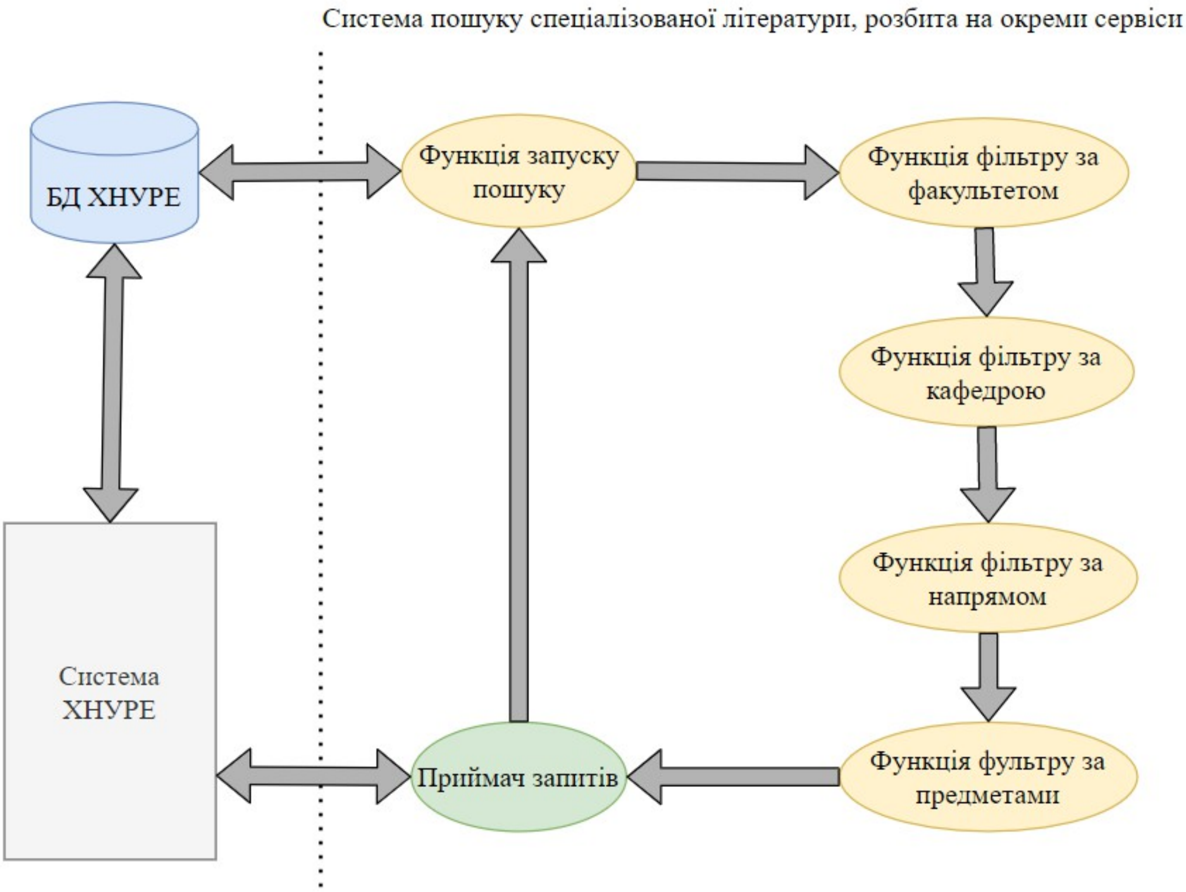
Розробив	*Говдерчак О.П.*			*Взаємодія системи підприємства (ХНУРЕ) із системою пошуку як окремим суцільним сервісом*
Перевірив	*Ситніков Д.Е.*			
Н. контр.	*Ситніков Д.Е.*			
				СПРм-19-2 — *Аркуш 9*
Затвердив	*Гребеннік І.В.*			*СТ* — *Аркушів 14*

Механізм розгортки суцільного монолітного додатку системи пошуку



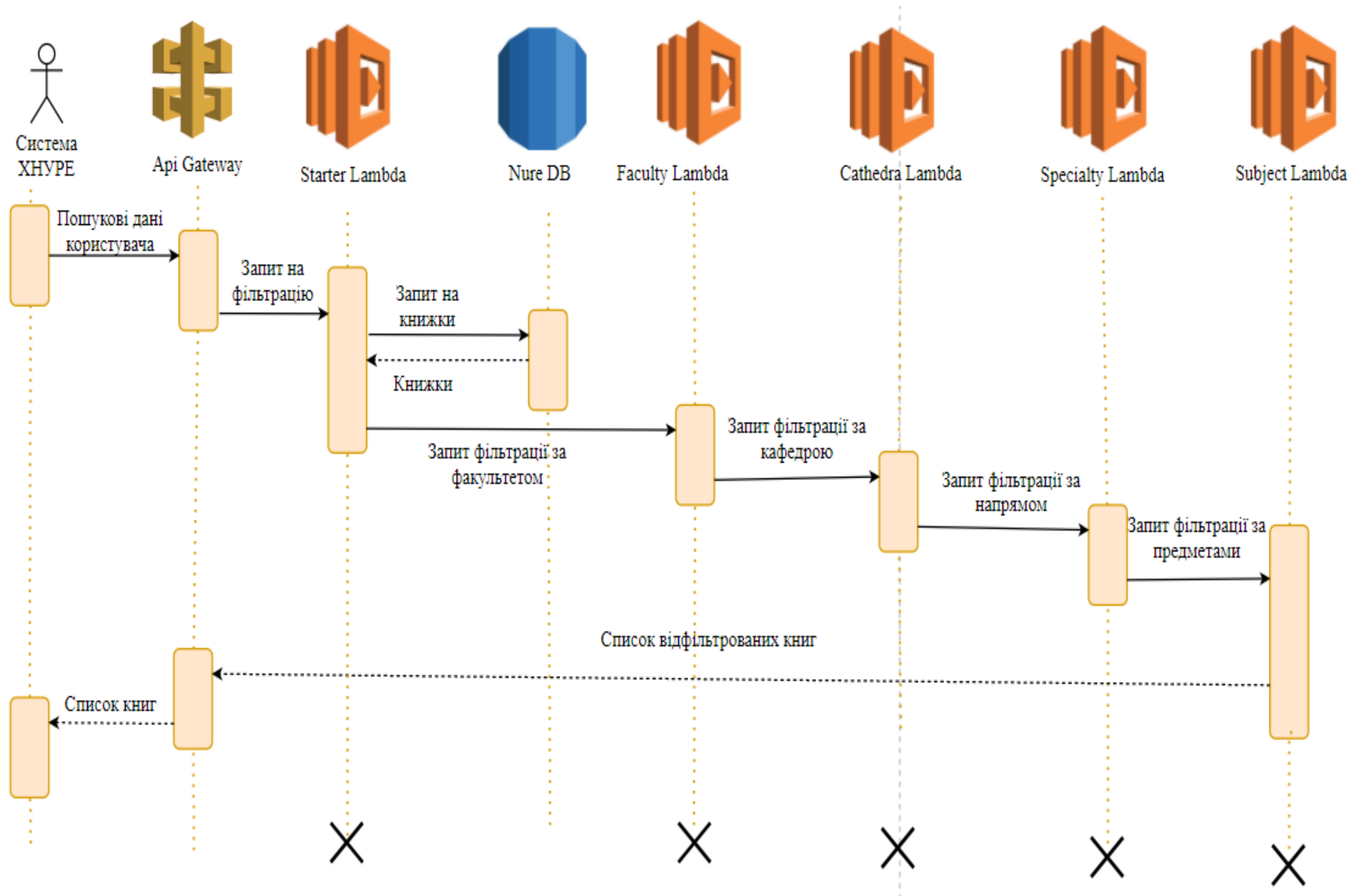
Розробив	*Говдерчак О.П.*			*Маханізм розгортки суцільного монолітного додатку системи пошуку*
Перевірив	*Ситніков Д.Е.*			
Н. контр.	*Ситніков Д.Е.*			
				СПРм-19-2 — *Аркуш* 10
Затвердив	*Гребеннік І.В.*			*СТ* — *Аркушів* 14

Взаємодія системи підприємства (ХНУРЕ) із системою пошуку як ланцюгом окремих сервісів



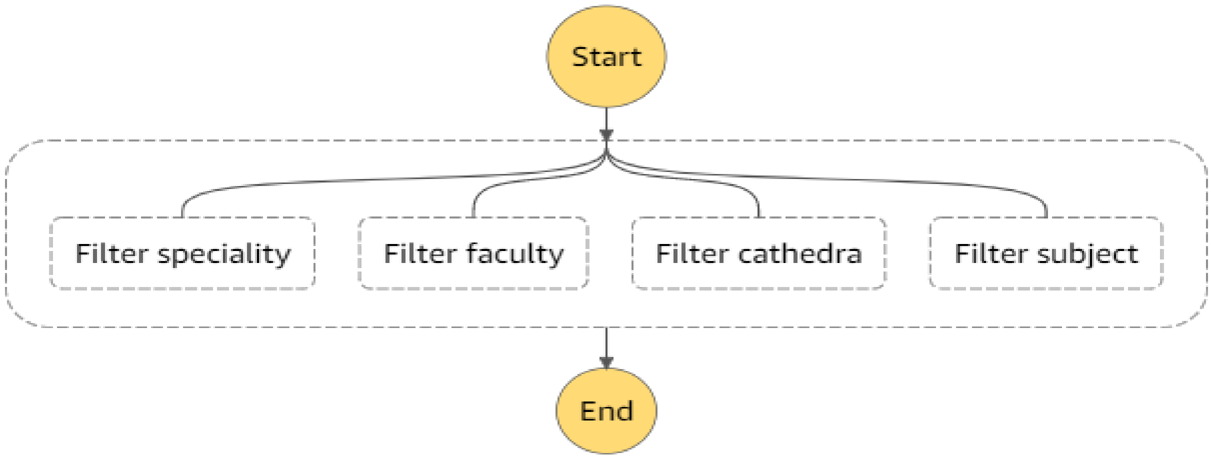
Розробив	*Говдерчак О.П.*			*Взаємодія системи підприємства (ХНУРЕ) із системою пошуку як ланцюгом окремих сервісів*	
Перевірив	*Ситніков Д.Е.*				
Н. контр.	*Ситніков Д.Е.*				
				СПРм-19-2	*Аркуш 11*
Затвердив	*Гребеннік І.В.*			*СТ*	*Аркушів 14*

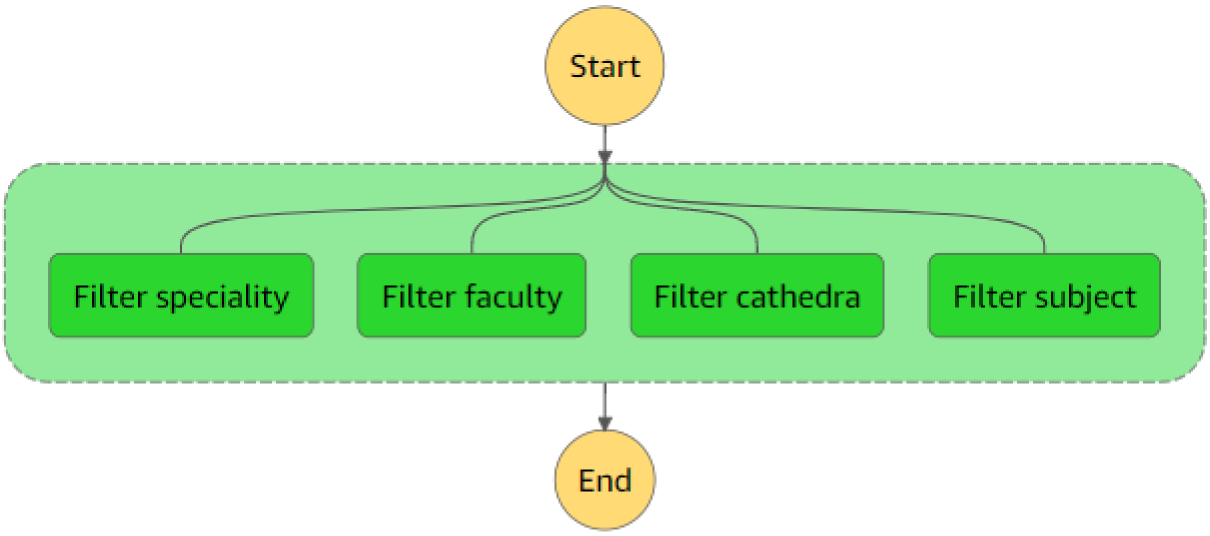
Діаграма послідовності роботи системи пошуку, розбитої на декілька сервісів (функцій)



Розробив	*Говдерчак О.П.*			*Діаграма послідовності роботи системи пошуку, розбитої на декілька сервісів (функцій)*
Перевірив	*Ситніков Д.Е.*			
Н. контр.	*Ситніков Д.Е.*			
				СПРм-19-2 — *Аркуш 12*
Затвердив	*Гребеннік І.В.*			*СТ* — *Аркушів 14*

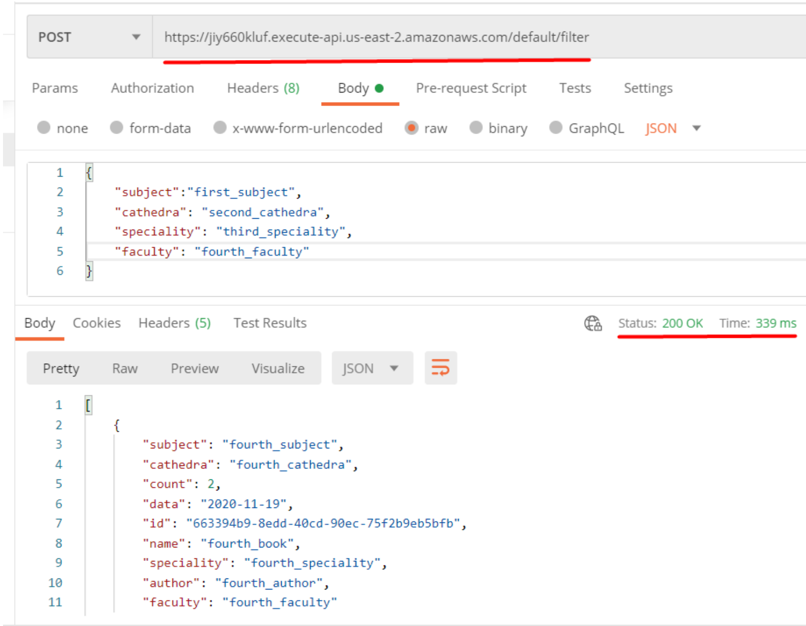
Граф роботи системи пошуку, розробленої за допомогою автомату стану Step Functions та працючий алгоритм системи

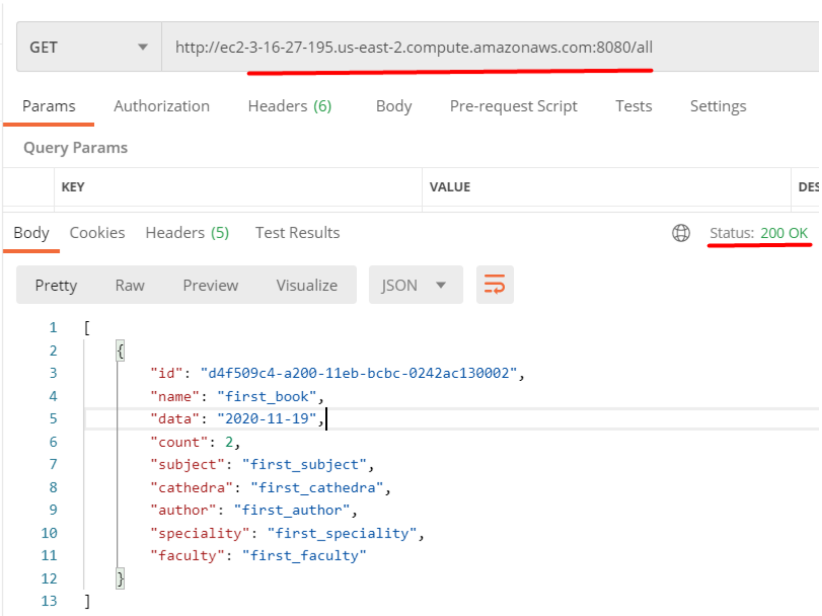




Розробив	*Говдерчак О.П.*			*Граф роботи системи пошуку, розробленої за допомогою автомату стану Step Functions та працючий алгоритм системи*	
Перевірив	*Ситніков Д.Е.*				
Н. контр.	*Ситніков Д.Е.*				
				СПРм-19-2	*Аркуш 13*
Затвердив	*Гребеннік І.В.*			*СТ*	*Аркушів 14*

Працюючі варіанти архітектур з фізичним сервером та ланцюгом безсерверних функцій





Розробив	*Говдерчак О.П.*			*Працюючі варіанти архітектур з фізичним сервером та ланцюгом безсерверних функцій*		
Перевірив	*Ситніков Д.Е.*					
Н. контр.	*Ситніков Д.Е.*					
				СПРм-19-2	*Аркуш*	*14*
Затвердив	*Гребеннік І.В.*			*СТ*	*Аркушів*	*14*

ДОДАТОК Б

ТЕКСТ ПРОГРАМИ АТЕСТАЦІЙНОЇ РОБОТИ

ГЮИК.506120.003 34 01 – 01 12 01

(позначення документа)

ЗАТВЕРДЖЕНО

ГЮИК.506120.003 34 01 – 01 12 01

Розробка мікро-сервісної архітектури систем пошуку спеціалізованої літератури

Текст програми

ГЮИК.506120.003 34 01 – 01 12 01

АРКУШІВ 11

2021 р.Харківський Національний Університет Радіоелектроніки

«ЗАТВЕРДЖУЮ»
керівник атестаційної
роботи
проф. Ситніков Д.Е.

Розробка мікро-сервісної архітектури систем пошуку спеціалізованої літератури

Текст програми

ЛИСТ ЗАТВЕРДЖЕННЯ

ГЮИК.506120.003 34 01 – 01 12 01

РОЗРОБИВ:
ст.гр. СПРМ-19-2
Говдерчак О.П.

2021 р.

ЗМІСТ

1 МОНОЛІТНА АРХІТЕКТУРА ОКРЕМОГО СЕРВІСУ СИСТЕМИ ПОШКУКУ

1.1 Розробка прошарку доступу до даних

Знадобиться сутність книжкового джерела – BookInfo.java:

```
package ua.nure.diploma.entity;

import com.amazonaws.services.dynamodbv2.datamodeling.DynamoDBAttribute;
import com.amazonaws.services.dynamodbv2.datamodeling.DynamoDBAutoGeneratedKey;
import com.amazonaws.services.dynamodbv2.datamodeling.DynamoDBHashKey;
import com.amazonaws.services.dynamodbv2.datamodeling.DynamoDBTable;
import lombok.Builder;
import lombok.Data;

import java.io.Serializable;

@Data
@Builder
@DynamoDBTable(tableName = "book")
public class BookInfo implements Serializable {

    @DynamoDBHashKey
    private String id;

    @DynamoDBAttribute
    private String name;

    @DynamoDBAttribute
    private String data;

    @DynamoDBAttribute
    private int count;

    @DynamoDBAttribute
    private String subject;

    @DynamoDBAttribute
    private String cathedra;

    @DynamoDBAttribute
    private String author;

    @DynamoDBAttribute
    private String speciality;

    @DynamoDBAttribute
    private String faculty;
}
```

Та репозиторій доступу BookInfoRepository.java:

```
package ua.nure.diploma.dao;

import org.socialsignin.spring.data.dynamodb.repository.EnableScan;
import org.springframework.data.repository.CrudRepository;
import ua.nure.diploma.entity.BookInfo;

@EnableScan
public interface  BookInfoRepository extends CrudRepository<BookInfo, String> {

}
```

1.2 Розробка прошарку бізнес-логіки

Додамо BookService.java:

```
package ua.nure.diploma.service;

import com.amazonaws.services.dynamodbv2.AmazonDynamoDB;
import com.amazonaws.services.dynamodbv2.datamodeling.DynamoDBMapper;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import ua.nure.diploma.dao.BookInfoRepository;
import ua.nure.diploma.entity.BookInfo;
import ua.nure.diploma.entity.UserInfo;

import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.function.BiPredicate;
import java.util.stream.Collectors;

@Service
public class BookService {

    private DynamoDBMapper dynamoDBMapper;

    private AmazonDynamoDB amazonDynamoDB;

    private BookInfoRepository bookInfoRepository;

    private Map<String, BiPredicate<BookInfo, UserInfo>> filters;


    @Autowired
    public BookService(AmazonDynamoDB amazonDynamoDB, BookInfoRepository bookInfoRepository,
DynamoDBMapper dynamoDBMapper) {
        this.amazonDynamoDB = amazonDynamoDB;
        this.bookInfoRepository = bookInfoRepository;
        this.dynamoDBMapper = dynamoDBMapper;
        filters = new HashMap<>();
        filters.put("cathedra", (bookInfo, userInfo) ->
bookInfo.getCathedra().equals(userInfo.getCathedra()));
        filters.put("speciality", (bookInfo, userInfo) ->
bookInfo.getSpeciality().equals(userInfo.getSpeciality()));
```

```
        filters.put("subject", (bookInfo, userInfo) ->
bookInfo.getSubject().equals(userInfo.getSubject()));
        filters.put("faculty", (bookInfo, userInfo) ->
bookInfo.getFaculty().equals(userInfo.getFaculty()));
    }

    public List<BookInfo> findAllBooks() {
        return (List<BookInfo>) bookInfoRepository.findAll();
    }

    public List<BookInfo> findFilteredBooks(UserInfo userInfo) {
        return findAllBooks().stream()
                .filter(bookInfo -> filterBook(bookInfo, userInfo))
                .collect(Collectors.toList());
    }

    public List<BookInfo> saveAll(List<BookInfo> books) {
        return (List<BookInfo>) bookInfoRepository.saveAll(books);
    }

    private boolean filterBook(BookInfo bookInfo, UserInfo userInfo) {
        return filters.values().stream()
                .anyMatch(bookInfoUserInfoBiPredicate ->
bookInfoUserInfoBiPredicate.test(bookInfo, userInfo));
    }
}
```

1.3 Розробка прошарку контролерів

Додамо BookController.java:

```
package ua.nure.diploma.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RestController;
import ua.nure.diploma.entity.BookInfo;
import ua.nure.diploma.entity.UserInfo;
import ua.nure.diploma.service.BookService;
import ua.nure.diploma.util.DbUtils;

import java.util.List;

@RestController
public class BookController {

    @Autowired
    private BookService bookService;

    @Autowired
    private DbUtils dbUtils;

    @GetMapping("/all")
```

```
    public List<BookInfo> findAllBooks() {
        return bookService.findAllBooks();
    }

    @PostMapping("/filtered")
    public List<BookInfo> filterBooks(@RequestBody UserInfo userInfo) {
        return bookService.findFilteredBooks(userInfo);
    }

    @PostMapping("/save/{count}")
    public List<BookInfo> saveAll(@PathVariable Integer count) {
        return bookService.saveAll(dbUtils.populateBooks(count));
    }
}
```

2 СИСТЕМА ПОШУКУ, РОЗБИТА НА БЕЗСЕРВЕРНІ ФУНКЦІЇ AWS LAMBDAS

2.1 Розробка функції запуску алгоритму пошуку

Функція Starter Lambda:

```
const AWS = require("aws-sdk");
const docClient = new AWS.DynamoDB.DocumentClient();
const lambdaClient = new AWS.Lambda({
  region: 'us-east-2'
});
exports.handler = async (event) => {
    console.log(event)
    const db_res = await docClient.scan({TableName: "book"}).promise();
    const payload = {
        books: db_res.Items,
        params: JSON.parse(event.body)
    }
    console.log('payload')
    console.log(JSON.stringify(payload))
    const result = await lambdaClient.invoke({
        FunctionName: 'arn:aws:lambda:us-east-2:639735456973:function:faculty_filter',
        Payload: JSON.stringify(payload)
    }).promise()
    console.log(JSON.stringify(JSON.parse(result.Payload).body))
    return {
        statusCode: 200,
        body: JSON.stringify(JSON.parse(result.Payload).body)
    };
};
```

2.2 Розробка функцій фільтрувань

Функція faculty_filter Lambda:

```
import json
import boto3

client = boto3.client('lambda')
def lambda_handler(event, context):
    req_params = event['params']
    books = event['books']
    faculty = req_params['faculty']
    result_book_list = list(filter(lambda book: book['faculty'] == faculty, books))
    left_books = [book for book in books if book not in result_book_list]
    if len(left_books) != 0:
        payload = json.dumps({
            'books': left_books,
            'params': req_params
        })
```

```
        invoke_response = client.invoke(FunctionName='arn:aws:lambda:us-east-
2:639735456973:function:cathedra_filter',
                InvocationType='RequestResponse',
                Payload=bytes(payload, encoding='utf-8')
                )
        data = json.loads(invoke_response['Payload'].read().decode())
        result_book_list.extend(data['body'])
      return {
        'statusCode': 200,
        'body': result_book_list
    }
```

Функція speciality_filter Lambda:

```
import json
import boto3
client = boto3.client('lambda')
def lambda_handler(event, context):
    req_params = event['params']
    books = event['books']
    speciality = req_params['speciality']
    result_book_list = list(filter(lambda book: book['speciality'] == speciality,
books))
    left_books = [book for book in books if book not in result_book_list]
    if len(left_books) != 0:
        payload = json.dumps({
            'books': left_books,
            'params': req_params
        })
        invoke_response = client.invoke(FunctionName='arn:aws:lambda:us-east-
2:639735456973:function:subject_filter',
                InvocationType='RequestResponse',
                Payload=bytes(payload, encoding='utf-8')
                )
        data = json.loads(invoke_response['Payload'].read().decode())
        result_book_list.extend(data['body'])
    return {
        'statusCode': 200,
        'body': result_book_list
    }
```

Функція cathedra_filter Lambda:

```
import json
import boto3

client = boto3.client('lambda')

def lambda_handler(event, context):

    req_params = event['params']
    books = event['books']
    cathedra = req_params['cathedra']

    result_book_list = list(filter(lambda book: book['cathedra'] == cathedra, books))
    left_books = [book for book in books if book not in result_book_list]

    if len(left_books) != 0:
```

```
            payload = json.dumps({
                'books': left_books,
                'params': req_params
            })

            invoke_response = client.invoke(FunctionName='arn:aws:lambda:us-east-
2:639735456973:function:speciality_filter',
                    InvocationType='RequestResponse',
                    Payload=bytes(payload, encoding='utf-8')
                    )
            data = json.loads(invoke_response['Payload'].read().decode())
            result_book_list.extend(data['body'])


        return {
            'statusCode': 200,
            'body': result_book_list
        }
```

Функція subject_filter Lambda:

```
import json
import boto3

def lambda_handler(event, context):

    req_params = event['params']
    books = event['books']
    subject = req_params['subject']

    result_book_list = list(filter(lambda book: book['subject'] == subject, books))

    return {
        'statusCode': 200,
        'body': result_book_list
    }
```

3 БЕЗСЕРВЕРНІ ФУНКЦІЇ СИСТЕМИ ПОШУКУ, ОБ'ЄДНАНІ АВТОМАТОМ СТАНУ AWS STEP FUNCTIONS

3.1 Розробка функції запуску алгоритму пошуку

Функція step_Starter Lambda:

```
const AWS = require('aws-sdk');
const stepFunctions = new AWS.StepFunctions();
const docClient = new AWS.DynamoDB.DocumentClient();

exports.handler = async (event, context, callback) => {

    const db_res = await docClient.scan({TableName: "book"}).promise();
    const payload = {
        left_books: db_res.Items,
        params: JSON.parse(event.body)
    }
    const params = {
        stateMachineArn: 'arn:aws:states:us-east-
2:639735456973:stateMachine:SearchFilterSteps',
        input: JSON.stringify(payload)
    };

 let result = stepFunctions.startExecution(params).promise().then((result) => {
        let paramsStatus = {
            executionArn: result.executionArn
        };
        let response = new Promise(function (resolve, reject) {
            let checkStatusOfStepFunction = setInterval(function () {
                //on regular interval keep checking status of step function
                stepFunctions.describeExecution(paramsStatus, function (err, data) {
                    console.debug('called describeExecution:', data.status);
                    if (err) {
                        clearInterval(checkStatusOfStepFunction);
                        reject(err);
                    } else {
                        if (data.status !== 'RUNNING') {
                            // this means step function execution is now finished and we get
result as data.output
                            clearInterval(checkStatusOfStepFunction);
                            resolve(data.output);
                        }
                    }
                });
            }, 100);
        });
        return response;
    });
    return result.then(value => JSON.parse(value));};
```

3.2 Розробка функцій фільтрувань

Функція step_faculty_filter Lambda:

```
import json
import boto3
```

```
client = boto3.client('lambda')

def lambda_handler(event, context):
    print(event)
    req_params = event['params']
    books = event['left_books']
    faculty = req_params['faculty']

    result_book_list = list(filter(lambda book: book['faculty'] == faculty, books))

    return {
        'result_faculty_books': result_book_list
    }
```

Функція step_speciality_filter Lambda:

```
import json
import boto3

client = boto3.client('lambda')

def lambda_handler(event, context):
    req_params = event['params']
    books = event['left_books']
    speciality = req_params['speciality']

    result_book_list = list(filter(lambda book: book['speciality'] == speciality,
books))

    return {
        'result_speciality_books': result_book_list
    }
```

Функція step_cathedra_filter Lambda:

```
import json
import boto3

client = boto3.client('lambda')

def lambda_handler(event, context):
    req_params = event['params']
    books = event['left_books']
    cathedra = req_params['cathedra']

    result_book_list = list(filter(lambda book: book['cathedra'] == cathedra, books))

    return {
        'result_cathedra_books': result_book_list
    }
```

Функція step_subject_filter Lambda:

```
import json
```

```python
import boto3

client = boto3.client('lambda')

def lambda_handler(event, context):
    req_params = event['params']
    books = event['left_books']
    subject = req_params['subject']

    result_book_list = list(filter(lambda book: book['subject'] == subject, books))

    return {
        'result_subject_books': result_book_list
    }
```

3.3 Розробка Step Function State Machine

JSON-подібний код автомату стану Амазона:

```json
{
  "Comment": "Diploma Step Function Example",
  "StartAt": "Parallel filtering",
  "States": {
    "Parallel filtering": {
      "Comment": "Parallel filtering.",
      "Type": "Parallel",
      "ResultPath":"$.lambda_result",
      "OutputPath":"$.lambda_result",
      "End": true,
      "Branches": [
        {
          "StartAt": "Filter speciality",
          "States": {
            "Filter speciality": {
              "Comment": "Filtering speciality.",
              "Type": "Task",
              "Resource": "arn:aws:lambda:us-east-
2:639735456973:function:step_speciality_filter",
              "InputPath": "$",
              "ResultPath": "$.lambda_result",
              "OutputPath": "$.lambda_result",
              "End": true
            }
          }
        },
        {
          "StartAt": "Filter faculty",
          "States": {
            "Filter faculty": {
              "Comment": "Filtering faculty.",
              "Type": "Task",
              "Resource": "arn:aws:lambda:us-east-
2:639735456973:function:step_faculty_filter",
              "InputPath": "$",
              "ResultPath": "$.lambda_result",
              "OutputPath": "$.lambda_result",
              "End": true
```

```
                }
              }
            },
            {
              "StartAt": "Filter cathedra",
              "States": {
                "Filter cathedra": {
                  "Comment": "Filtering cathedra.",
                  "Type": "Task",
                  "Resource": "arn:aws:lambda:us-east-
2:639735456973:function:step_cathedra_filter",
                  "InputPath": "$",
                  "ResultPath": "$.lambda_result",
                  "OutputPath": "$.lambda_result",
                  "End": true
                }
              }
            },
            {
              "StartAt": "Filter subject",
              "States": {
                "Filter subject": {
                  "Comment": "Filtering subject.",
                  "Type": "Task",
                  "Resource": "arn:aws:lambda:us-east-
2:639735456973:function:step_subject_filter",
                  "InputPath": "$",
                  "ResultPath": "$.lambda_result",
                  "OutputPath": "$.lambda_result",
                  "End": true
                }
              }
            }
          ]
        }
      }
    }
```

ДОДАТОК В

ВІДОМІСТЬ АТЕСТАЦІЙНОЇ РОБОТИ

ГЮИК.506190.040 ДЗ

(позначення документа)

ЗАТВЕРДЖЕНО
ГЮИК. ГЮИК.506190.040 ДЗ – ЛУ

Розробка мікро-сервісної архітектури систем пошуку спеціалізованої літератури

Відомість атестаційної роботи

ГЮИК.506190.040 ДЗ

АРКУШІВ 2

2021 р.Харківський Національний Університет Радіоелектроніки

«ЗАТВЕРДЖУЮ»
керівник атестаційної
роботи
проф. Ситніков Д.Е.

Розробка мікро-сервісної архітектури систем пошуку спеціалізованої літератури

Відомість атестаційної роботи

ЛИСТ ЗАТВЕРДЖЕННЯ

ГЮИК.506190.040 Д3 – ЛУ

РОЗРОБИВ:
ст.гр. СПРМ-19-2
Говдерчак О.П.

2021 р.

№	Позначення	Найменування	Додаткові. відомості

		Текстові документи	
1.	**ГЮИК.506120.003 ИЗ**	Пояснювальна записка	87 с.
2.	**ГЮИК.506120.003 34 01 – 01 12 01**	Текст програми	11 с., CD-R
3.		Графічні документи	
		Принципи монолітної	
		та мікросервісної	
		архітектури	Включено в ПЗ
		Діаграма послідовності	Включено в ПЗ
		Система підприємства	
		монолітного зразка з	
		системою пошуку	Включено в ПЗ
		Популярність	
		використання	
		хмарних технологій	Включено в ПЗ
		Популярність Amazon	
		книги читачеві» (IDEF0)	
		серед інших	Включено в ПЗ

<table>
<tr><td></td><td></td><td></td><td></td><td></td><td rowspan="2" colspan="4">ГЮИК.506190.040 ДЗ</td></tr>
<tr><td>Ізм.</td><td>Аркуш</td><td>№ докум.</td><td>Підпис</td><td>Дата</td></tr>
<tr><td colspan="2">Розробив</td><td>Говдерчак О.П.</td><td></td><td></td><td rowspan="4">Розробка мікро-сервісної архітектури систем пошуку спеціалізованої літератури</td><td></td><td>Аркуш</td><td>Аркушів</td></tr>
<tr><td colspan="2">Перевірив.</td><td>Ситніков Д.Е.</td><td></td><td></td><td></td><td>1</td><td>2</td></tr>
<tr><td colspan="2">Н. Контр.</td><td></td><td></td><td></td><td rowspan="2" colspan="3">ХНУРЕ
Кафедра СТ</td></tr>
<tr><td colspan="2">Затвердив</td><td>Гребеннік І.В.</td><td></td><td></td></tr>
</table>

		Порівняння	
		безсерверних	
		функцій Амазону	Включено в ПЗ
		Фізична модель БД	

		Прикладу підприємства	Включено в ПЗ
		Необхідні таблиці	
		для системи пошуку	
		та їх реплікація	Включено в ПЗ
		Взаємодія системи	
		підприємства із	
		системою пошуку	
		Механізм розгортки	
		суцільного додатку	
		пропонування літератури	Включено в ПЗ
		Діаграма послідовності	
		окремих функцій	
		системи пошуку	
		Працездатність кожної	
		системи пошуку	Включено в ПЗ

<table>
<tr><td></td><td></td><td></td><td></td><td></td><td rowspan="2" colspan="4">ГЮИК.506190.040 ДЗ</td></tr>
<tr><td>Ізм.</td><td>Аркуш</td><td>№ докум</td><td>Підпис</td><td>Дата</td></tr>
<tr><td colspan="2">Розробив</td><td>Говдерчак О.П.</td><td></td><td></td><td rowspan="4">Розробка мікро-сервісної архітектури систем пошуку спеціалізованої літератури</td><td></td><td>Аркуш</td><td>Аркушів</td></tr>
<tr><td colspan="2">Перевірив.</td><td>Ситніков Д.Е.</td><td></td><td></td><td></td><td>2</td><td>2</td></tr>
<tr><td colspan="2">Н. Контр.</td><td></td><td></td><td></td><td rowspan="2" colspan="3">ХНУРЕ
Кафедра СТ</td></tr>
<tr><td colspan="2">Затвердив</td><td>Гребенік І.В.</td><td></td><td></td></tr>
</table>