**Knowledge-Hub (Know-Max)**

Knowledge sharing is an extensive and important aspect of the current academic environment. The existing libraries have multiple shortcomings when it comes to availability and accessibility of books. Libraries generally have few copies of books and updated editions might not be available leading to the services being limited. The libraries might not be accessible all the time by all students and faculty. Maintenance of books also becomes an overhead as the volume keeps increasing. There are also risks and delays involved with retrieving the loaned books and accounting for lost resources.

The Universities in Missouri have collaborated to introduce an initiative named K-Hub which addresses the above problems and links all their resources and makes them available to students/faculty across the state. Resources from all the participating universities are hosted in a single repository. The books are maintained as digitized copies which ensures that the books are available and accessible all the time. Physical maintenance overhead is considerably reduced, and all the administrative actions are relatively simpler. Risks involving loss of books are almost nil.

In the proposed system, participating universities can create their profile and store their books in the given server space. The books stored in the centralized server can be managed by their respective librarians. The students and faculty should sign up in the system with their university email id. The books available in the system can be accessed only for reading purpose (can’t be downloaded or printed). The system also has options to request for new editions of e-books and students/faculty can also review and rate the books available. They can search effectively using a keyword, title, author and genre. Advanced search options using full text search, highly rated, ISBN, name of publisher, year of publishing and frequently viewed books.

Actors

1. Student
2. Librarian
3. System Administrator

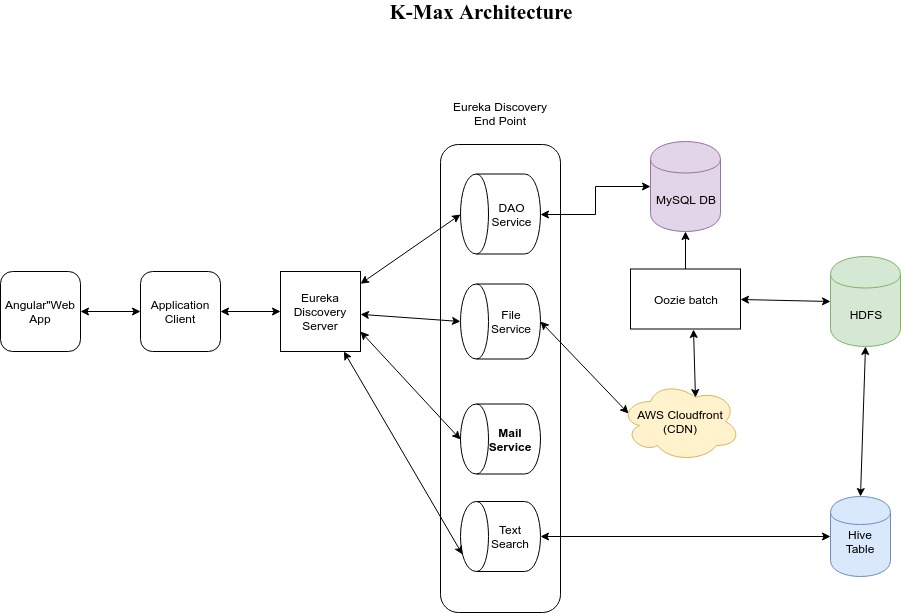
Use cases

1. Student registers, logs in, searches books and reads books in the system
2. Faculty registers, searches books and manage books in the system
3. Librarian manages their university resources in the given file server
4. Librarian views usage information related to their resources
5. System allows users to search books based on different search options
6. System administrator adds the university and sets the storage space to the university
7. System administrator adds, removes the librarians from different universities
8. System administrator views site usage information
9. Administrator manages institution
10. Batch application converts newly uploaded ebooks to text for analysis

Functional Requirements

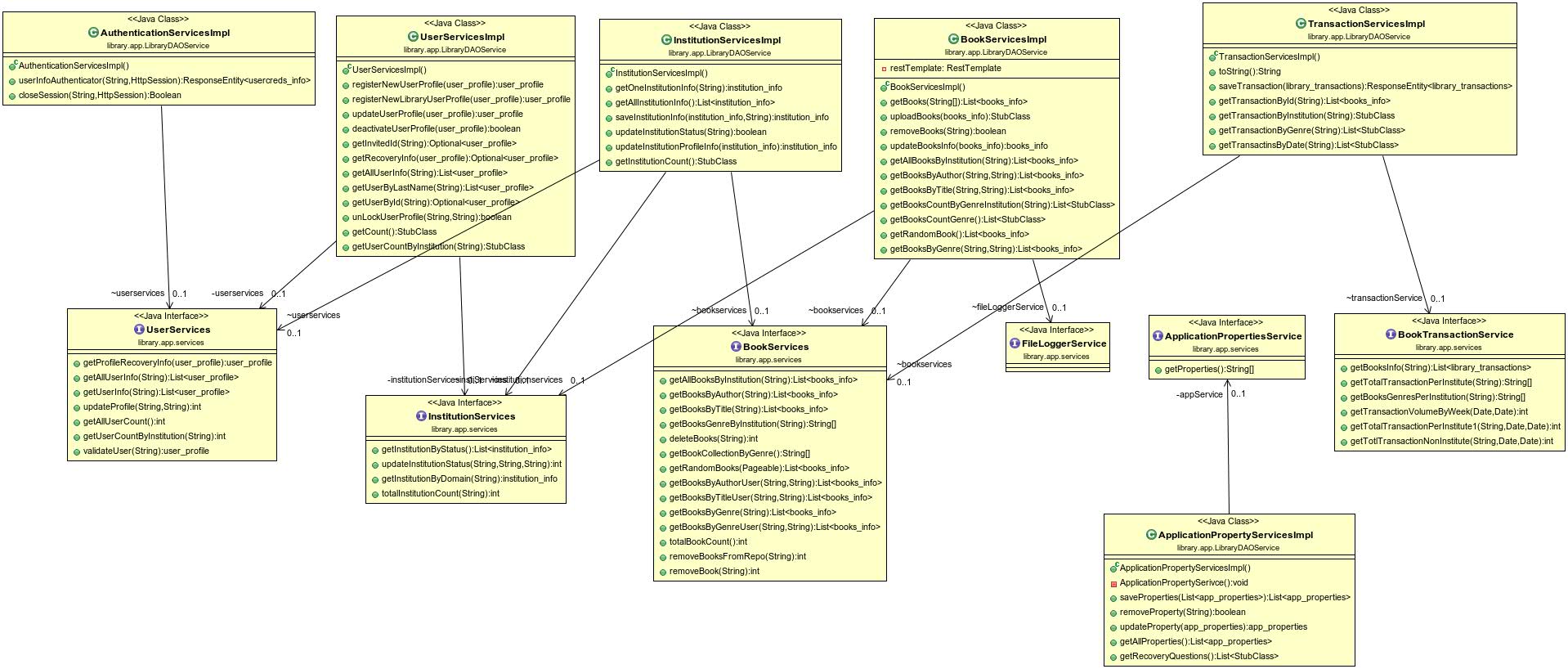
* 1. Student creates profile
  2. Student manages profile
  3. Student searches book
  4. Student reads book
  5. Administrator adds institution
  6. Administrator manages institution
  7. Administrator upload ebooks
  8. Administrator manages ebooks
  9. Administrator create librarian profile
  10. Administrator manage librarian profile
  11. Administrator views dashboard
  12. Librarian views dashboard
  13. Librarian upload books
  14. Librarian manage books
  15. Batch job converts e-books to text files for analysis
  16. Batch job (MapReduce) extracts keywords and counts from text files
  17. Batch job read the extracted keywords and store Hive DB
  18. Batch job calculates tf-idf and prepares feature data
  19. Administrator create cloud folder for Institution repository
  20. Administrator remove cloud folder for Institution repository

**Proposed system #1**

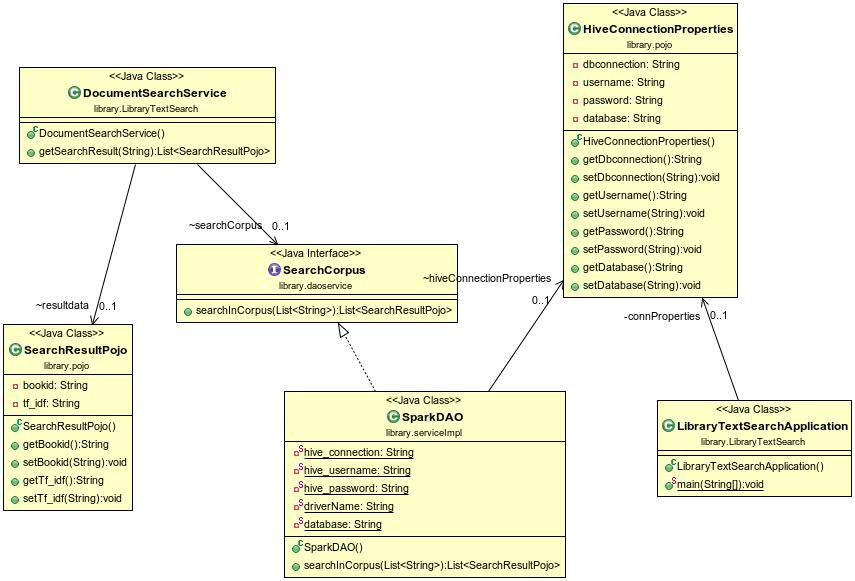


**Class Diagram**

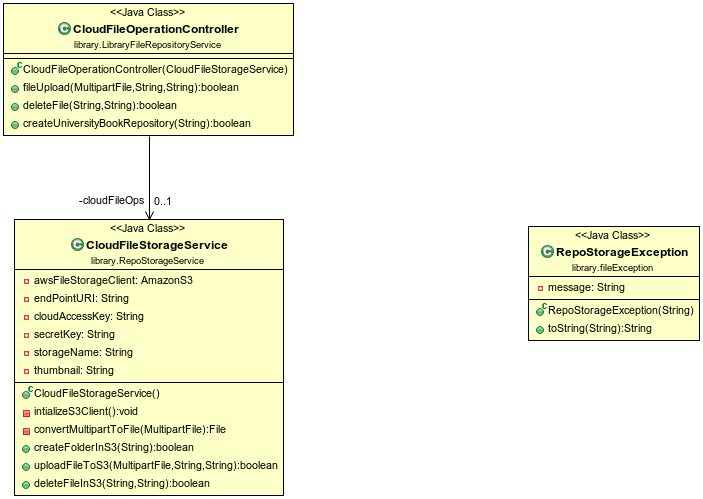
**Discovery Producer - DAO**



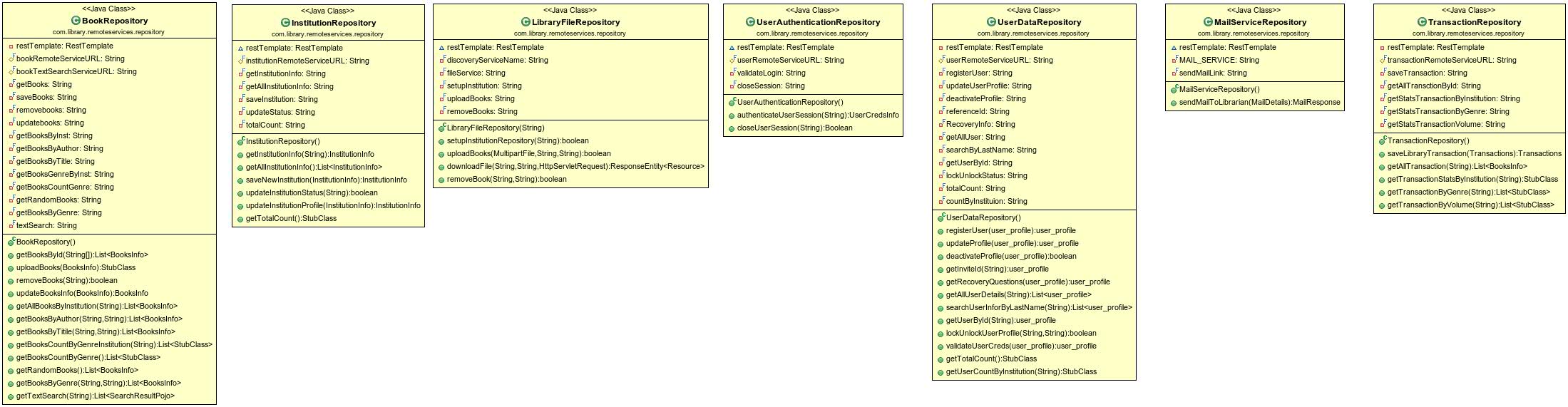
**Discovery Producer – Text Search**



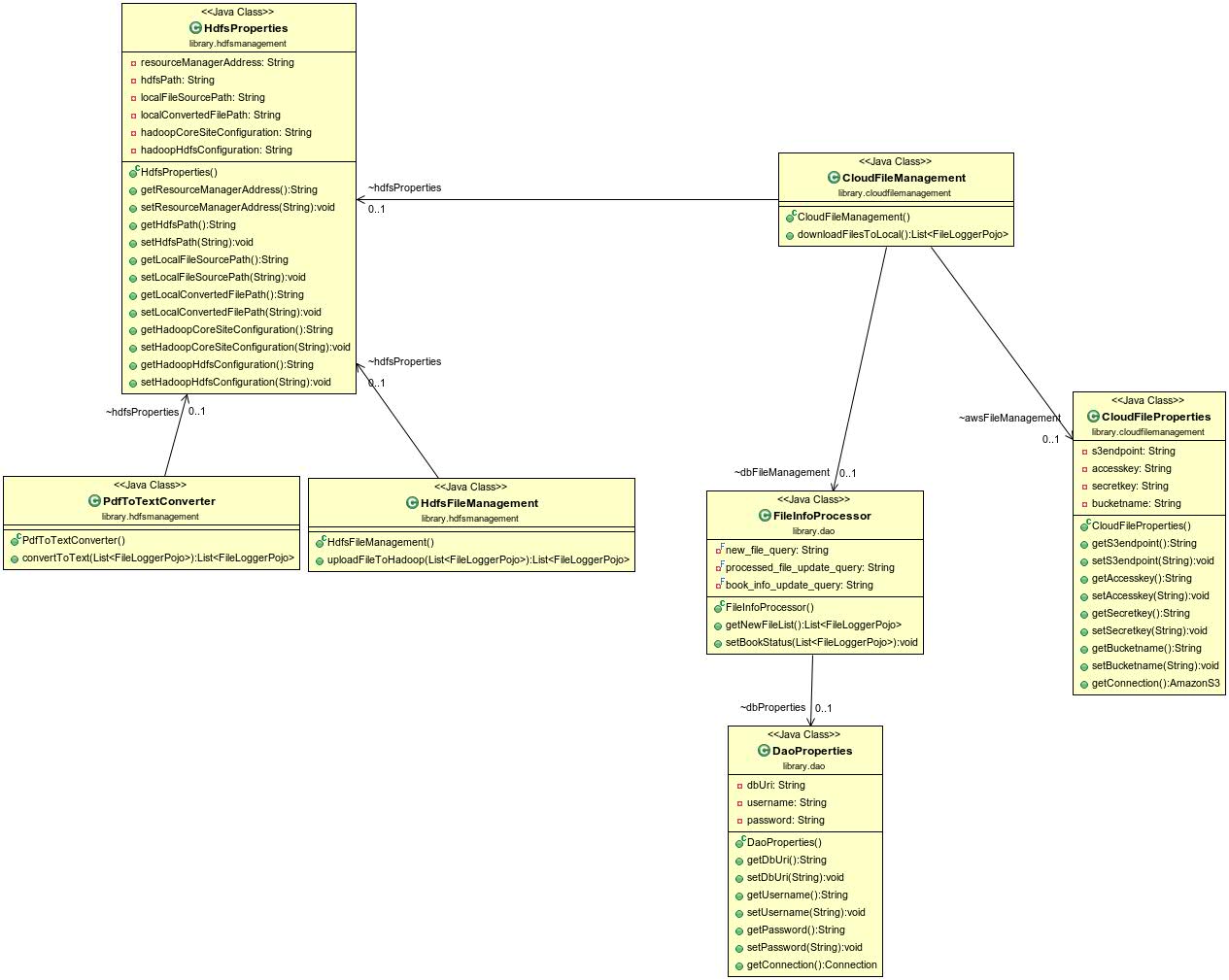
**Discovery Producer – Cloud File Management**



**Discovery - Client**

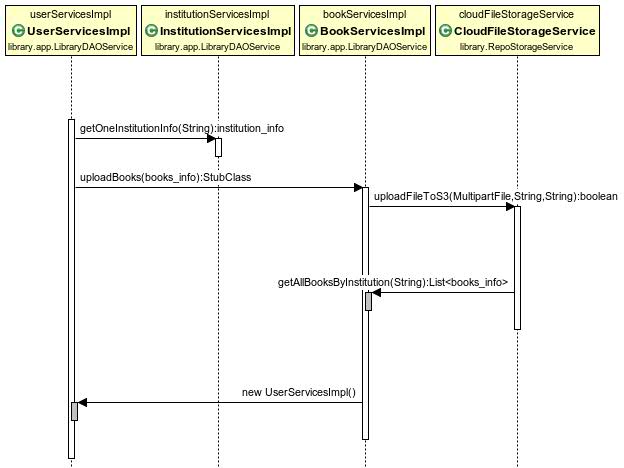


**Library App – File Processor**

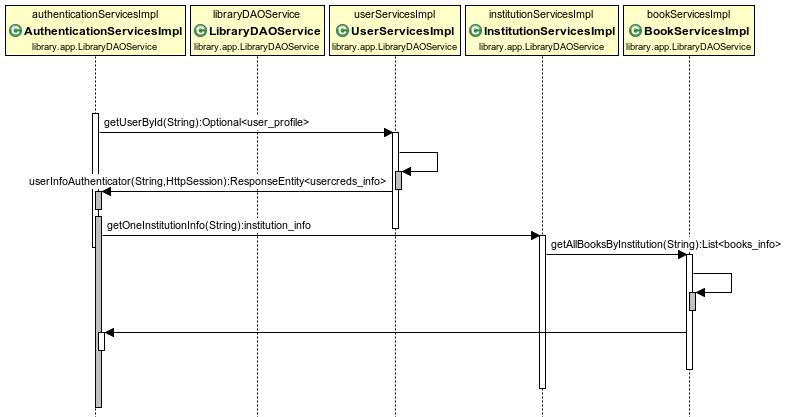


**Sequence Diagrams**

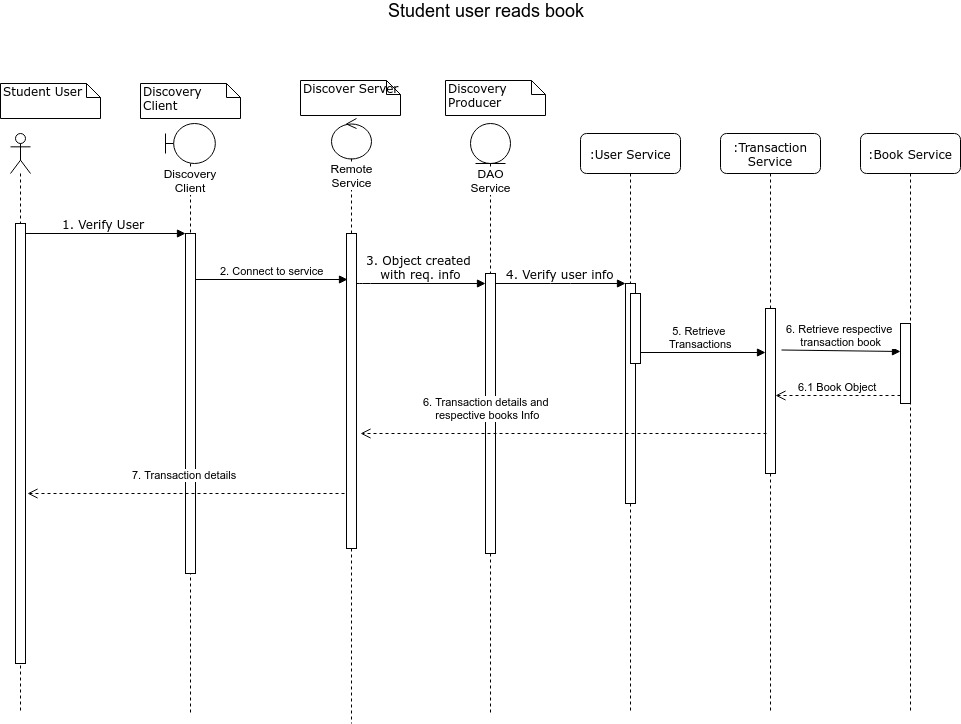
**Add library sequence**



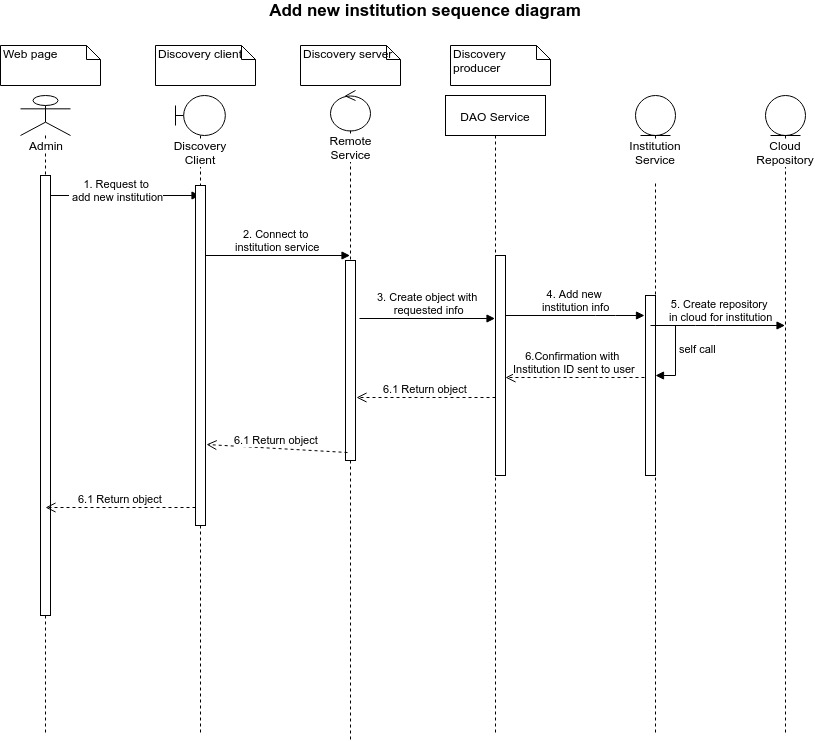
**Book search sequence – Instituion**



**Retrieve User transactions**



**Add New Institution**



**Technologies**

**Front end**

1. Angular6
2. CSS Bootstrap
3. Simple pdf viewer

**Server Side**

1. Micro services with Spring Boot 5
2. Oozie 3
3. Netflix Eureka Discovery server

**File server**

1. AWS Cloudfront CDN

**Data storage and analytic Tools**

1. Hive server2
2. Spark with MLlib for searching
3. Hadoop 3 with Yarn for storing and running analytics services

**Transactional Data storage (DBMS)**

1. MySQLTechnologies