A diagram of a software

Description automatically generated

Microservices

1. Auth Service
   * This is used as authentication for Staff, Students and university administrative staff
   * This uses Amazon Relational Database . The database stores student data which contains studentId, email,

Courses taken by student and other profile data related to student and staff

1. Upload Courses

* This service is responsible for storing data related to courses
* The all the metadata related to like course professor,

Course Id, course code , course name is stored in DocumentDB(Mongo Db)

* The course related details will also be stored in ElasticSearch which will power our fuzzy search
* Amazon S3 will be used to store course syllabus, and other blob related items which are require heavy memory.
* Amazon S3 can be later used to store assignments related to coursework , midterm exams and other books related to course

1. Search Courses and Register Courses

* Searching of Courses will be powered by ElasticSearch which will allow fuzzy search
* Registration will store data in 2 tables one will be Relational Database and Cassandra
* Relational Database will update student profile and store courses registered by the student , this will power queries like – Find all the courses taken by the student
* Cassandra – This will store all the students registered in a courses . Since this an ever-increasing data and Cassandra is read and update optimized and a distributed storage.

This will power queries – Find all the students registered in the course, courseId will be the key and value can be studentId’s

* Cache can persist frequent data like course details which will improve performance during peak registration time
* It can also store results data related to all students registered in a course , during registration time for frequent access courses

1. Notification Service

* Used to send notifications to students or staff.

Consistency Model

* Eventual Consistency – The whole architecture is a event driven architecture where each microservice communicates with other through events which are passed in through kafka