DATA MIGRATOR

A PROJECT REPORT

Submitted by

BRAHMADUTTAN S (TKM20MCA-2015)

to

The APJ Abdul Kalam Technological University

In partial fulfillment for the award of the degree of

MASTER OF COMPUTER APPLICATIONS

Thangal Kunju Musaliar College of Engineering
Kerala

DEPARTMENT OF COMPUTER APPLICATIONS

JULY 2022

DECLARATION

I undersigned hereby declare that the project report on **DATA MIGRATOR**, submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of Prof. Vaheetha Salam. This submission represents my ideas in my own words and where ideas or words of others have been included,I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University..

Kollam

18-07-2022

BRAHMADUTTAN S

DEPT. OF COMPUTER APPLICATIONS TKM COLLEGE OF ENGINEERING

KOLLAM

2021 - 22

CERTIFICATE

This is to certify that the report entitled **DATA MIGRATOR** submitted by **BRAHMADUTTAN S** (TKM20MCA2015) to the APJ Abdul Kalam Technological University in partial fulfillment of the Masters degree in Computer Applications is a bonafide record of the project work carried out by him under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Internal Supervisor Head of the Department External Examiner

Knowledge Lens Pvt. Ltd
74/A, Keonics, Hosur road,
Electronic City Phase 1 - Bengaluru - 560100
CIN No: U72200KA2013PTC072670

Date:19-07-2022

TO WHOM IT MAY CONCERN

This is to certify that Mr. Brahmaduttan S from TKM College of Engineering, Kollam is pursuing the internship on **"DATA MIGRATOR"** project at **Knowledge Lens Pvt Ltd** starting from 2nd May 2022 to till 22nd July 2022. During the period of his training with us he is found punctual, hardworking, and inquisitive.

We wish him all the success in future endeavors.

Yours Truly,

For **Knowledge Lens Pvt Ltd**.

Jayashree S
HR Manager
hr@knowledgelens.com

Acknowledgement

First and foremost I thank GOD almighty and my parents for the success of this project. I owe sincere gratitude and heart full thanks to everyone who shared their precious time and knowledge for the successful completion of my project.

I am extremely grateful to **Dr. Fousia M Shamsudeen**, Head of the Department, Department of Computer Applications, for providing me with best facilities.

I would like to thank my coordinator and project guide **Prof. Vaheetha Salam**, Department of Computer Applications, who motivated me throughout the project .

I would like to thank my external coordinator **Mr. Amal G Jose** and **Mr. Chandrakanth Gottam**, Knowledge Lens, who guided me throughout my work.

I profusely thank all other faculty members in the department and all other members of TKM College of Engineering, for their guidance and inspirations throughout my course of study.

I owe my thanks to my friends and all others who have directly or indirectly helped me in the successful completion of this project.

BRAHMADUTTAN S

ABSTRACT

DATA MIGRATOR,is a web platform that allows data to be moved quickly from one database to another. It facilitates the migration of several collections either in their entirety as-is or by specifically selecting one or more in a single click. One such tool that businesses employ to move data from one database to another is database migration. Database migrations, also known as schema migrations, can assist with programmatically managing incremental changes to data structures. Database migration's main objective is to make database changes repeatable, testable, and shareable without erasing any data.

The primary goal of development is to execute database migrations in order to reduce the risk of downtime and data corruption. Businesses can save money on infrastructure as well as the expertise and manpower required for ongoing support. Professional database migration services are highly adaptable and have been shown to improve system security, reliability, speed, and disaster recovery planning. Database migration can also reduce the likelihood of costly service disruption.

Despite their importance, data migration projects can be extremely complex. Data migration necessitates downtime, which may disrupt data management operations. This is why it is critical to understand the risks of DB migration, as well as best practises and tools that can aid in the process.

Contents

List of Figures

Chapter 1

Introduction

"**DATA MIGRATOR**" is a web application that allows users to quickly move their data between databases only a single click. It facilitates the migration of several collections either in their entirety as-is or by specifically selecting one or more in a single click. One such tool that businesses employ to move data from one DB to another is database migration. Database migrations are, also known as schema migrations, can assist with programmatically managing incremental changes to data structures. Database migration's main objective is to make database changes repeatable, testable, and shareable without erasing any data. Development's main objective is to execute database migration in a way that reduces the risk of downtime and the potential for data corruption.Businesses can save money on infrastructure as well as the expertise and manpower required for ongoing support. Professional database migration services are highly adaptable and have been demonstrated to improve system security, dependability, performance, and disaster recovery planning. Furthermore, database migration may reduce the likelihood of costly service interruption. Data migration projects can be quite challenging, despite the fact that they are crucial. Data management processes may be interrupted if data migration necessitates downtime. This is why it's crucial to comprehend the hazards associated with DB migration, as well as best practises and the tools that can make the process go smoothly.

1.1 Objective

The goal is to accomplish the following:

- Save On Overhead Expenses
- Enhance Speed Performance
- Reduce Redundant Data
- Create Reliable Backups
- Schema Build
- Move data from an outdated legacy system to a modernized software
- Migrating data in 4 Ways:
  - Mongo DB to Cosmos DB
  - Cosmos DB to Mongo DB
  - Mongo DB to Mongo DB
  - Cosmos DB to Cosmos DB

1.2 Company Profile

The Lens product line from Knowledge Lens automates and simplifies the discovery of hidden insights within Big Data. The objective is to convert unstructured data into actionable business insights. We are big data technology nerds with a wide range of big data projects, including big data engineering, data science, and industry expertise.

1.2.1 Products

- **iLens(Intelligent Lens)**
  A platform called iLens offers smart integration with numerous devices or sensors in big businesses, the manufacturing sector, homes, and other places.
  In order to record time series data in real time, iLens offers a MQTT interface for smooth integration of various field sensor devices. ILens has the capacity to produce alerts and alarms depending on pre-configured rules.

- **Mlens**

  MLens is a one-step solution for managing disaster recovery for big data and platforms.

  Features of MLens :

  1. Big Data Backup Migration
  2. Automated Disaster Recovery
  3. Data Encryption, compression Archival
  4. High Speed Batch Data Ingestion
  5. Monitoring Scheduling
  6. Secured Access controls

- **AiLens**

  Next-generation AI platform with a collaborative workspace, experiment designer, engineering workbench for modelling features, and integrations for enterprise security and DevOps with the repository of AI/ML assets.

  With a single graphical user interface for creating Data Engineering and AI/ML pipelines, AiLens is an intelligent AI helper. The unified AI orchestrator in AiLens allows users to launch model executions on any runtime, including Tensorflow, SparkML, H2O, MxNet, Theano, PyTorch, and AWS / Azure, via a console. Given that every user will have the same experience, AiLens is extremely adaptable.

  regardless of any future technology developments due to the platform's meta model-driven design. Our software stands out from the competition by a wide margin thanks to its intuitive job submission and monitoring structure, secure integration with external entities, built-in encryption, and support for role-based access control.

  Key features are:

  1. Any AI Stack, Any AI Algorithm, Anywhere
  2. Unified AI Orchestrator
  3. Simplified User Experience
  4. Intelligent Assistant for AI
  5. Integrated Data Preparation AI Modelling Environment
  6. Seamless Enterprise Security Integration

- **GLens**

  The GLens product package for monitoring ambient air quality, industrial emissions, and effluent discharges uses real-time data acquisition, monitoring, and analytics. GLens DAS Software, GLens Server Platform, and GLens Environ Data Logger provide a comprehensive answer to all industrial environmental requirements. With a plug-and-play paradigm, the platform may be connected to any analyzer, sensor, or device to acquire data in real time.

  The key features of GLens are:

  1. Rest based open protocol for multi-client deployment.
  2. Real time alerts and alarms with SMS and Email integration.
  3. Remote calibration and configuration of analyzers.
  4. Plug and play complete protocol integration with any analyzer make and model. – Integrated and data quality codes as per ISO 7168.
  5. Integrated analytics and predictive models for effective pollution control.
  6. Live consolidated industry dashboards.

1.2.2 Services

- **Big Data Engineering Services**

  We offer comprehensive Big Data Protects Architecture, Design, Development, Testing, and Deployment.

- **Big Data Security Services**

  We are among the specialist consulting firms that offer specialised Big Data Services.

- **Big Data Analytics Services**

  Using our pre-built analytical Lens, we uncover hidden insights from a vast array of data sources.

- **Big Data Competency Development**

  We provide one of the top enterprise-wide Big Data Competency Development programmes despite our lack of specialised knowledge.

Chapter 2

Literature Survey

A literature review is an in-depth examination and interpretation of relevant literature. A literature review identifies research questions and then seeks answers by searching for and analysing relevant literature. Re-analyzing study results can yield new insights, which is why literature reviews are essential.A literature review summarises and discusses the current level of knowledge on a subject, as found in scholarly publications and journal articles. At university, you can write one of two sorts of literature reviews: one as an independent course assignment, and the other as an introduction to or preparation for a longer piece, such as a thesis or research report. The type of review you write will define the focus and perspective of your review, as well as the type of hypothesis or thesis argument you present. Reading published literature reviews or the introductory chapters of theses and dissertations in your topic area is one approach to comprehend the differences between these two forms. Analyze their argumentation strategy and approach to problem-solving.

2.1 Purpose of the Literature Review

1. It makes research on a certain issue accessible to readers by selecting and summarising relevant, significant, important, and valid papers or studies.

2. This is an ideal beginning place for new researchers in a new topic, as it demands them to summarise, evaluate, and compare the original research in that field.

3. It discourages researchers from replicating work that has already been accomplished.

4. It can point the way forward for future research or suggest areas to focus on.

5. It highlights the key findings.

6. Examining the literature for discrepancies, omissions, and contradictions.

7. It offers a critical assessment of other researchers' methodology and approaches.

2.2 Related Works

The process of moving data from one database to another is known as data migration. Companies are shifting their databases from one database (e.g., RDBMS) to a new database for a number of reasons, such as the fact that the new database can handle more data, is quicker and more scalable, etc (e.g., NoSQL). There are several tools and techniques accessible, including sqoop, mongoimport, and mongify. These may be utilised to convert RDBMS databases to NoSQL databases. In contrast to RDBMS databases, NoSQL databases employ document models, graph models, key-value stores, and other technologies. NoSQL was created for the primary purpose of storing and retrieving massive amounts of data. The team offers in this study a database migration methodology that can migrate both real-time and historical data.

The recommended approach is implemented in Java for the MySQL (RDBMS) to MongoDB document database. At a predefined time, the prototype simultaneously migrates live data and a database instance (database snapshot). Over the past three to four decades, relational databases have been utilised by several organisations to store and analyse corporate databases. Relational databases store information in a relational or structural fashion because they adhere to the relational model. A relational database is utilised when data has some structure and grows slowly. Multiple sources are currently producing data at an exponential rate.

In this study, the Snapshot-Live Stream Db Migration Model is proposed as a hybrid migration model for document databases, along with a parallel approach for an automatic migration solution. [1]

Since its debut, the NoSQL database has become increasingly popular. In terms of database technology, it is still a relatively new idea with a lot of open questions, notably around data transfer from one NoSQL database to another. Due to this flaw, a comprehensive migration architecture that covers all phases of data migration must be developed. The data migration issue has given rise to a number of frameworks, each with its unique approach, traits, and features. In order to build a generic framework, this study combines all the frameworks

by contrasting them and focusing on a key variable.Included in the proposed framework to facilitate data movement are algorithm migration, model migration, and migration schemes.

NoSQL-database.org reports that there are now over 225 NoSQL databases. On the basis of how they store and handle data, NoSQL databases may be classified as 1) key-value, 2) document, 3) column, and 4) graph. Since NoSQL employs a distinct data storage, data management approach, programming language, and implementation, it might complicate data movement.

There are eight main steps in the proposed framework: Determination, preparation, extraction, transformation, loading, validation, testing, and documentation are the first six steps. Regarding the data migration that will be implemented, each stage may consist of a number of tasks. The framework also contains migration models, methodologies, and schemes. In order to test the framework in the real world, a specific amount of data is moved across datastores (i.e. Document to Column). This framework describes and accounts for each phase of the migration process. This framework was created by combining the variety of earlier frameworks with DSRM (Design Science Research Methodology). It is envisaged that the established framework would be ubiquitous, flexible, and of high quality. [2]

NoSQL databases are developed to overcome problems associated with massive data processing. Data migration from an original database to a NoSQL database involves the creation of novel ideas and techniques due to our ever-increasing data volume. This article discusses a plan for migrating from a relational database to a document-oriented NoSQL database, as well as many migration methodologies and database reverse engineering (MongoDB). The approach focuses on establishing a data model, from which the schema is migrated and the data is mapped while conforming to collection integrity restrictions. Relational database modelling is used to prevent duplication and build a tree data structure for a collection of related tables. Transactions that enable actions to be taken to establish a healthy state increase the data's integrity. Communication between two separate DBMS types is required for database migration.

The method for transitioning from the relational model to the document-oriented paradigm is examined in this paper. outlines a migration strategy to MongoDB, beginning with a method for utilising metadata to extract a relational database's data structure. Then, move the schema and data according to the data model while respecting the integrity requirements. By linking collections and combining data through foreign keys, migration eliminates joins, solves the

issue of duplicated data, and increases the capacity of relational databases (ensureIndex and UniqueIndexes). [3]

The operation of the information system may be affected by database migration. Integrity, truthfulness, and business continuity are the process's challenging issues. We covered business continuity in this article. A new and old concurrent network environment is being constructed at the same time. The ocr (Oracle Cluster Registry), voting disc, and ASM data are being brought online. With no downtime, we successfully migrated the university's essential databases that enabled teaching, research, IC cards, and other operations. Additionally, the method did not interfere with or negatively impact the information system.

Database migration can be described in two different ways. To upgrade to a more recent release is the first and most widely used definition of database migration. Another definition of database migration states that it is the process of transferring data from one database management system to another, whether it be in the same physical location or not.

Database migration technologies come in a variety of flavours. The impact on business is classified into three types: the first is that the database is only read during the migration process; The second is that the database can be written throughout the migration process, but the company may be affected during the database's final switching process; and the third is that the migration and switching processes have no effect on the business.

- Read Only Migration
- Almost Online Migration
- Completely Online Migration

This project will also include the core of the data centre and the migration of the ORACLE RAC database. And this migration process should strive to move as much as feasible online; otherwise, it will impede regular school instruction, scientific research, and other normal development activities. In this regard, the following database online migration technique is proposed.[4]

- Environment Design
- Migration Design
- Implement

Multi-database systems are similar to distributed database systems. It consists of many separate database management systems. In contrast, the multi-database system lacks a centralised method for managing all data. A database system running on a single computer is used to store and manage all data. Although the multi-database system does not necessarily require all of the database system's attributes, each database system must have them. Several multi-database systems only permit data access in dire circumstances. Then, using existing database systems, the multi-database system may be built. As a result, building a multi-database system is simpler than building a distributed database system with central management.

In a multi-database system, an incremental data migration scheme based on an incremental online rearrangement technique is one approach for data movement. It is adaptive to data migration in multi-database systems since it does not use data snapshots or versions.[5]

Without physical restructuring, a multi-database system unites many database management systems across a network. Users can access the data in the multiple database system as if it were a single database. Each component database is able to deliver its own service while working as an integrated database system.

This strategy partitions a huge data migration into many smaller data migrations. By adding more inquiries between modest data migrations, response times for additional queries can be improved. Nevertheless, certain modest data migrations are repeated until all target data have been migrated. This strategy increases the amount of exclusive controls, including locking operations. By separating the data migration procedure, however, these locking zones should shrink and the total cost of these locking actions should not increase significantly.

The approach for incremental data migration on a multi-database system based on MySQL and the SPIDER storage engine. Our testing results indicate that, with the exception of a few circumstances, our incremental data transfer method is effective at reducing execution time.[6]

Relational database management solutions, according to some experts, are no longer able to handle the data management challenges posed by many contemporary applications. NoSQL databases also provide a more effective way of storing data as compared to traditional databases, including easy scaling, quick I/O, and low cost. In order to keep up with demand, more and more applications must migrate data from relational databases to NoSQL databases. To complete the job, it is vital to use efficient tools. The lack of standardised or consistent models or methods for completing the process, however, prevents NoSQL databases from being widely used. A common mid-model must be provided in order to maximise the efficacy of model

transition and data migration across relational databases and numerous NoSQLs.

Identifying the model differences between conventional relational databases and NoSQL databases is the most challenging component of data conversion. If we wish to develop a model that is consistent across relational and NoSQL databases, we must first understand the differing physical models of these two kinds of databases.

- Relational data model
- Document data model

The mid-model introduces the data feature and query feature. Using these abstract principles and many established tactics, we design a full process that can be implemented in the real world.Additionally, this approach is evaluated. Based on this technique, a consistent mid-model is created to ease model transition and data migration between relational and NoSQL databases. As there is no generally applicable consistent and complete model, it is unusual that this model is compatible with a number of major NoSQL databases.[7]

Chapter 3

Methodology

DATA MIGRATOR is a web-based software designed to migrate data from one database to another. However, a monitoring and analysis dashboard is included, which allows you to see the status of Migration information in a single window. The project has a dashboard in Data Migrator that displays a graphical representation of overall Job status, including total Job success rate, failure rate, and amount pending. It provides information on the migration count, migration status in percentage, and the average migration time required to complete the migration. In the dashboard, display the most recent jobs completed by the user, along with the job name, database information, and the level at which the user stopped the job. Also displays the number of Completed Jobs and the number of Jobs In Progress, as well as how many data will be successfully migrated from the source to the destination, how many are pending, and how many failed, all of which information is provided in the Dashboard. The focus here is primarily on data migration from source to destination, schema creation, and job status monitoring. This application has multiple jobs added at the same time.

Because this is the first version of the application, it focuses on Data Migration on MongoDB and Azure Cosmos DB. Here, it acquires data from the source database using the REST API and stores it in our PostgreSQL, and when the user proceeds with the migration, it migrates data from PostgreSQL to the destination database, as shown in Figure 3.1, and which receives an exact and detailed Job status. In this case, employ charts in the user interface to display Job status, and the echarts package is used to do so.

Migrating data in 4 Ways:

- Mongo DB to Cosmos DB
- Cosmos DB to Mongo DB
- Mongo DB to Mongo DB
- Cosmos DB to Cosmos DB

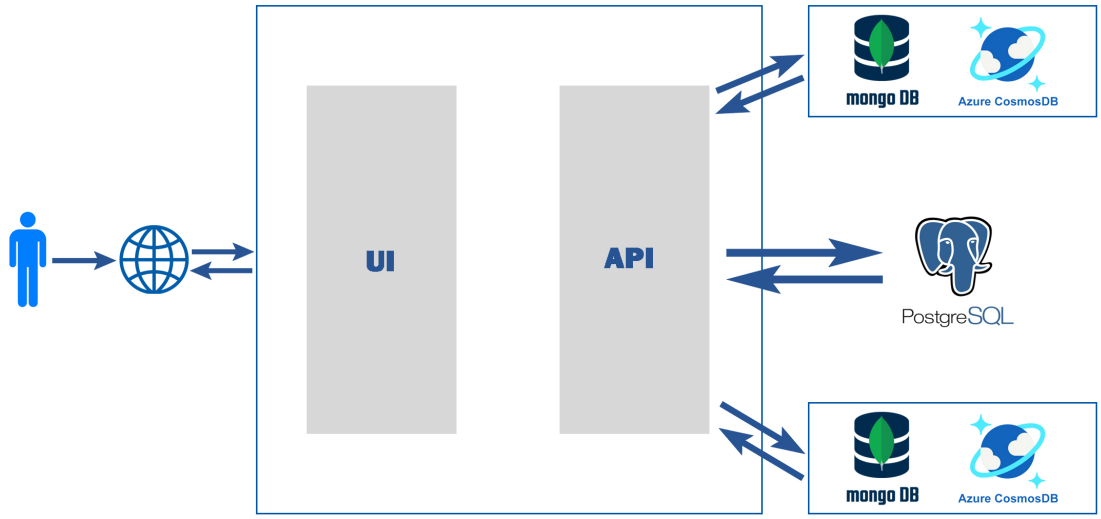


Figure 3.1: Work flow

3.1 Mapping the Databases

Here when credentials passed on both source and target databases, then fetching the source database data begins and the data first store it in the our postgreSQL database and after fetching all the data from source then only the migration to the target database begins.

3.2 Post-migration optimization

After you migrate the data stored in source database to the target DB, you can connect to target DB and manage the data. You can also perform other post-migration optimization steps. These might include optimizing the indexing policy, updating the default consistency level, or configuring global distribution for database account.

3.3 System Specifications

This section describes the application development architecture identified for this project based on needs.

3.3.1 Software Specification

- Programming Language : Python
- Designing tools : Angular
- Web server : Nginx
- Web Browser : Any web browser
- Database : PostgreSQL, MongoDB, Cosmos DB
- Backend: Python FastAPI

3.3.2 Software Description

- **Python**

  A well-liked general-purpose high-level programming language is Python. Because of its emphasis on code readability and simpler syntax than in languages like C++ or Java, it enables programmers to express concepts in smaller chunks of code. The language contains constructions that enable both small- and large-scale programmes to be clearly expressed. Python supports programming paradigms including object-oriented, imperative, functional, and procedural. It has a dynamic type system, an extensive standard library, automatic memory management, and all of these things.

- **Angular**

  Angular is a TypeScript-based, open-source framework for developing web-based applications. It is currently one of the most popular UI frameworks. With Angular, it is simple to create faster web applications, and Angular is a platform that makes it simple to create web applications. Angular provides a simple and powerful template syntax for rapidly generating UI views.The main features of Angular are ,It is component

based,Using TypeScript,Using concepts of Services that could be used across multiple applications

- **PostgreSQL**

  PostgreSQL is powerful object relational database.It has a powerful community support, because it is developing since last 30 years with a powerful open source community.Nowadays most of the companies like Uber,Netflix,Spotify,Instgram are using Postgres in their infrastructure.Postgres can be accessed by command line tool psql and or postgres client PgAdmin

- **MongoDB**

  MongoDB is an open-source, cross-platform document-oriented database application. A NoSQL database management system, MongoDB, leverages JSON-like documents with optional schemas. MongoDB was developed by MongoDB Inc. and is licenced under the Server Side Public License (SSPL), which is considered non-free by a number of distributions.

  MongoDB Atlas adds additional robust capabilities, such as continuous backups and point-in-time recovery, to increase the availability of your mission-critical production databases. MongoDB Atlas simplifies database access management. Your database instances are installed in a separate Virtual Private Cloud to provide network isolation (VPC). IP whitelisting or VPC peering, always-on authentication, encryption at rest and in transit, and sophisticated role-based access control are further security features.

  Features:

  - **Ad-hoc queries:** Field, range, and regular-expression searches are supported by MongoDB. In addition to returning specified document fields, queries may also return user-defined JavaScript functions. It is also possible to arrange queries to return a random sample of a given number of results.
  - **Indexing:** The fields of a MongoDB document may be indexed using main and secondary indices or index.
  - **Replication:** Services are a collection of code that may be used by many application components. Therefore, if you had a data component that retrieved

data from a database, you could implement it as a shared service that could be used by various apps. Environment setup.

- **Editor:** Through the use of replica sets, MongoDB offers high availability. A replica set contains two or more copies of identical data. Any member of the replica-set may assume the position of main or secondary replica at any moment.
- **Load balancing:** Sharding enables horizontal scalability in MongoDB. The user specifies a shard key, which determines the distribution of data inside a collection. Based on the shard key, the data is separated into ranges and spread over several shards. A shard is composed of a master and one or more copies.
- **File storage:** MongoDB may be used to store files as a file system known as GridFS. The GridFS function is available with the MongoDB drivers. Developers may manipulate files and information using MongoDB's provided functionalities.

- **Cosmos DB**

  Cosmos Database (DB) is a multi-model, low-latency, worldwide distributed database built for large-scale data management. It is a PaaS-based cloud-based NoSQL database offered by Microsoft Azure (Platform as a Service). It is a highly available, high throughput, and dependable serverless database. The Azure Document DB is included inside the Cosmos database, which is accessible from anywhere. Azure Cosmos DB is extensively used in online and mobile apps and is great for modelling social interactions, connecting with third-party services, and delivering personalised experiences with rich content. The Cosmos DB SDKs may be used to develop sophisticated iOS and Android apps using the popular Xamarin framework.

Chapter 4

RESULT AND DISCUSSION

Existing data are exported from the current DBMS and imported into the new DBMS throughout this procedure. Even though the same data may be present in the migrated database, interactions with the database are likely to vary, particularly when migrations occur from extremely old systems, those created with different technologies, or when the new database was not designed with the intended users in mind. The database inside an organisation facilitates organisational activities and keeps data across functions. These functionalities that assist user tasks are provided through a user interface. A comprehensive data transfer plan avoids a substandard experience that causes more issues than it solves. In addition to missing deadlines and over budgets, insufficient preparations might result in the total failure of relocation initiatives. Teams must devote their complete focus to planning and strategizing migrations, rather than subordinating them to another project with a larger scope.

4.1 Testing

A software product or programme is tested to determine whether it functions as intended. Testing has benefits such as preventing flaws, cutting costs associated with development, and improving performance. examining the product requirements for accuracy and completeness in a variety of contexts, such as the infrastructure, the industry, the business, the implementation feasibility and viability, the performance, the security, and the usability.

4.1.1 Unit Testing

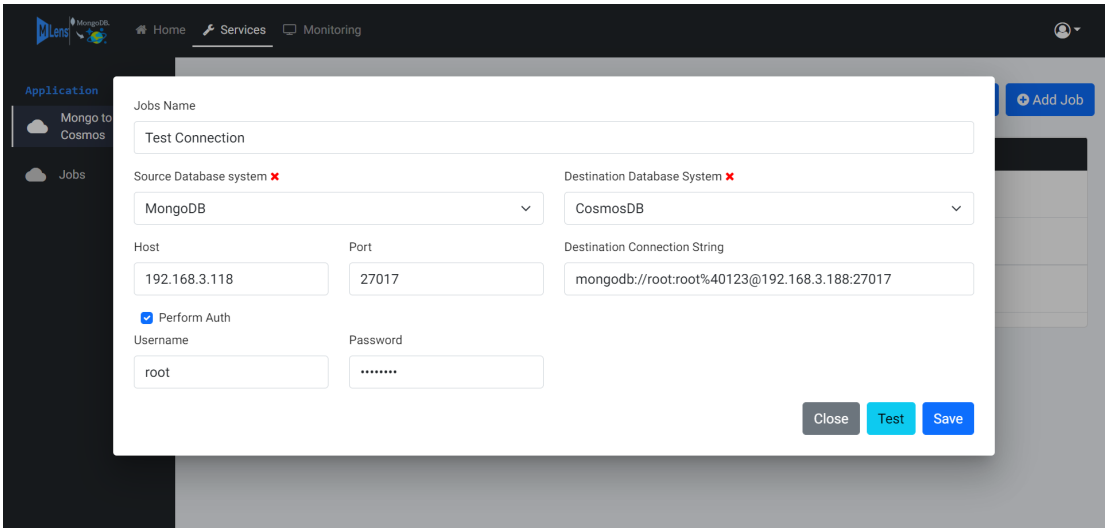
The smallest testable parts of a programme, known as units, are separately and independently examined for proper operation during the unit testing phase of software development. This testing strategy is used by software developers and occasionally by QA staff throughout the development process. The main goal of unit testing is to test separately developed code and see if it works as expected.

4.1.2 Integration Testing

The several parts, modules, and/or components of a software application are checked in concert during integration testing, a type of software testing. Testing the interfaces between modules is done to check for potential problems that can arise when these components are integrated and have to communicate.

4.1.3 Validation error occurs

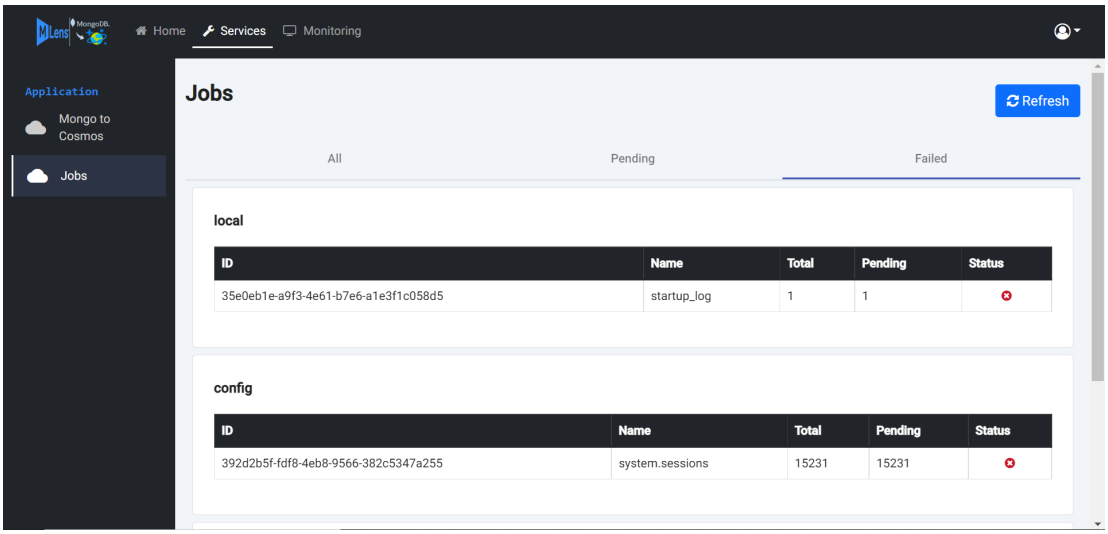
When the application is tested to make sure it works right, the validation error happens when the source database is not properly validated. This was fixed by adding columns for the source and destination username and password and validating the database before migration.



4.1.4 Migration error occurs

After the completion of migration from source to destination, some data was not migrated and failed, which is shown in the jobs summary. Actually, the local and config type data are

unique for every collection of data. Some are pending, and after hitting refresh, all data will be migrated into the destination database.



4.2 Output Screens and Results

1. Login page

   When user hits the Login button this page is loaded

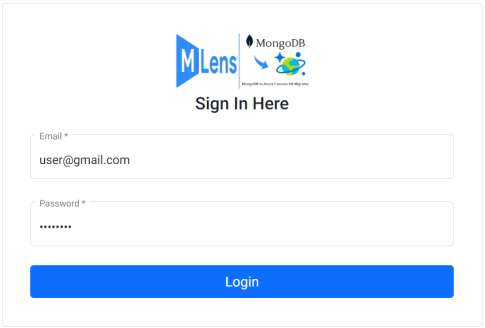


Figure 3.1: Login Page

2. Dashboard page

   Dashboard page is divided into Three parts:

   - Overall cost Dashboard

- Bar chart with filters usage Cost Dashboard
- Daily usage Cost Dashboard
- Total Accumulated Cost Dashboard

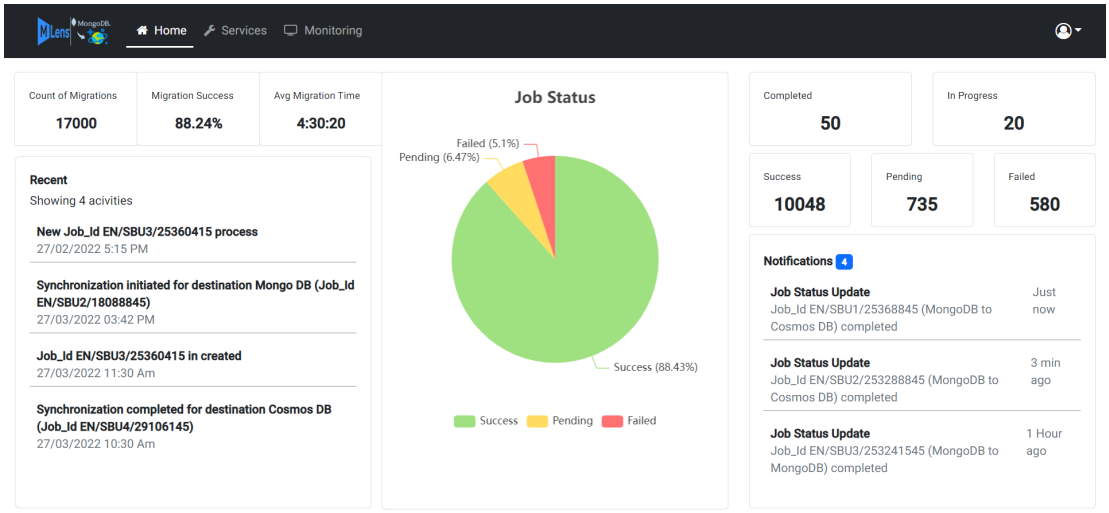


Figure 3.2: Dashboard Page

The dashboard lists the status of all the job's done by the user , and there is a top navbar including the pages Services and Monitoring.

It uses three filters in the charts: month, date, and day, number of weeks, number of months, and custom date range. After that, sort by resource.

3. Services Page

- The Services page is where you can perform add a job, sync a job, connect to that job, refresh the services to get the sync status previously done.

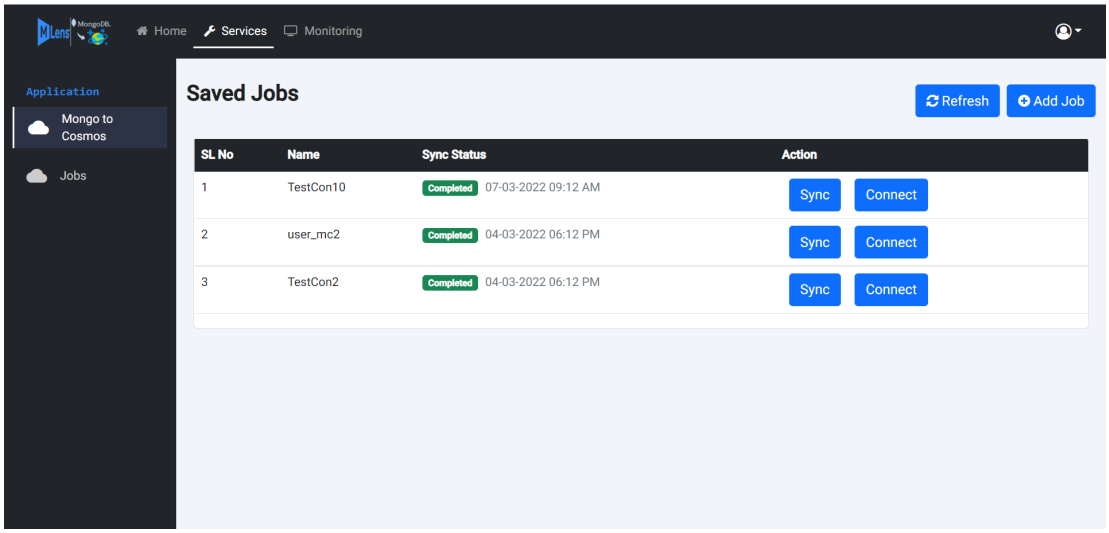


Figure 3.3: Services Page

4. Add Job

- Add Job is a pop-up dialogue to add a new Job for migration.
- Users can add the database details with there corresponding source DB and destination DB. The user can add a name to the job, also user will perform authentication with username and password, if the database they have selected to migration.

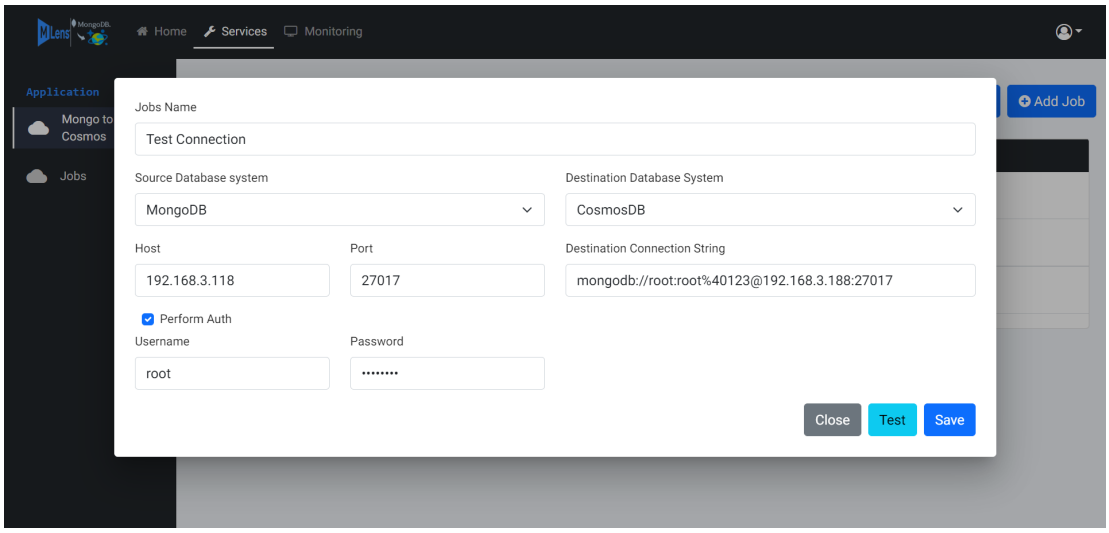


Figure 3.4: Add Job

5. Job Connection Checking

- After hits ”save” then assuring the credentials entered in the source and destination DB details

- There is a loader performs when checking connection.

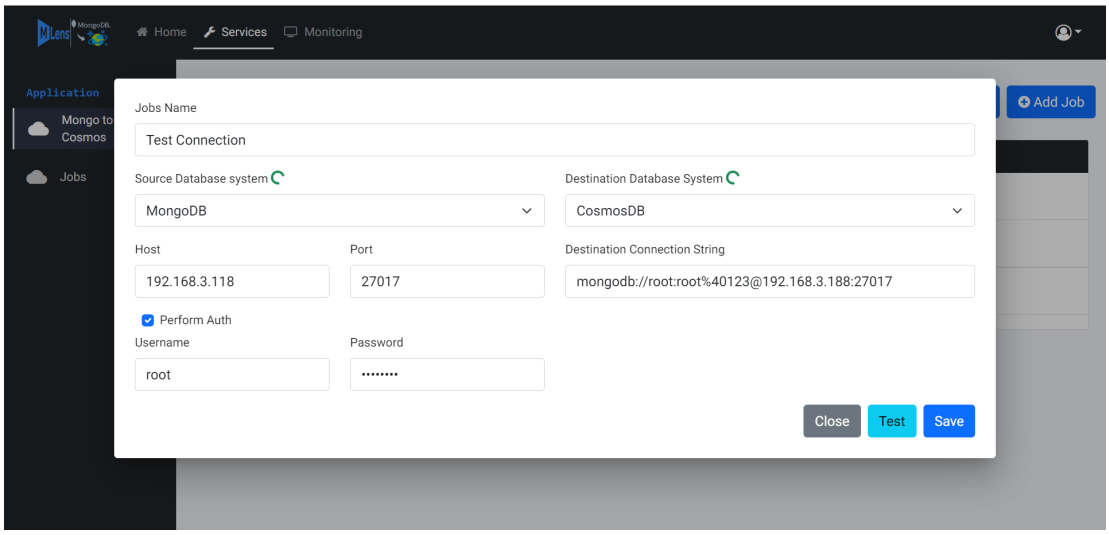


Figure 3.5: Connection checking

6. Database listing

- When the user hits the sync button the database will be listed as this
- The user can select by specific one or by migrate all

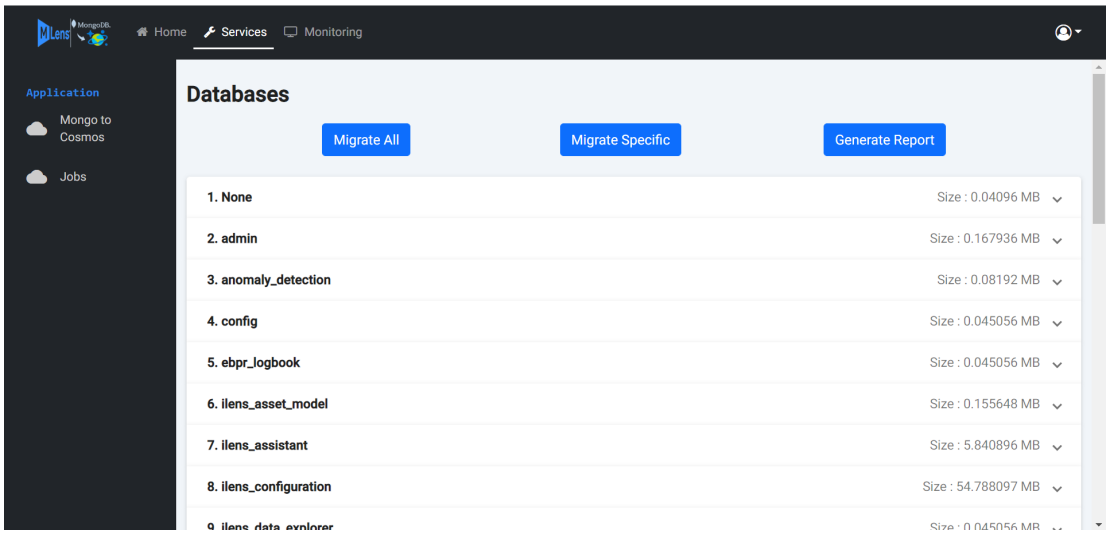


Figure 3.6: Database listing

7. Collection Schema

- User can see the collection schema when the user check by one by one through the Dropdown option when the Database is listing.

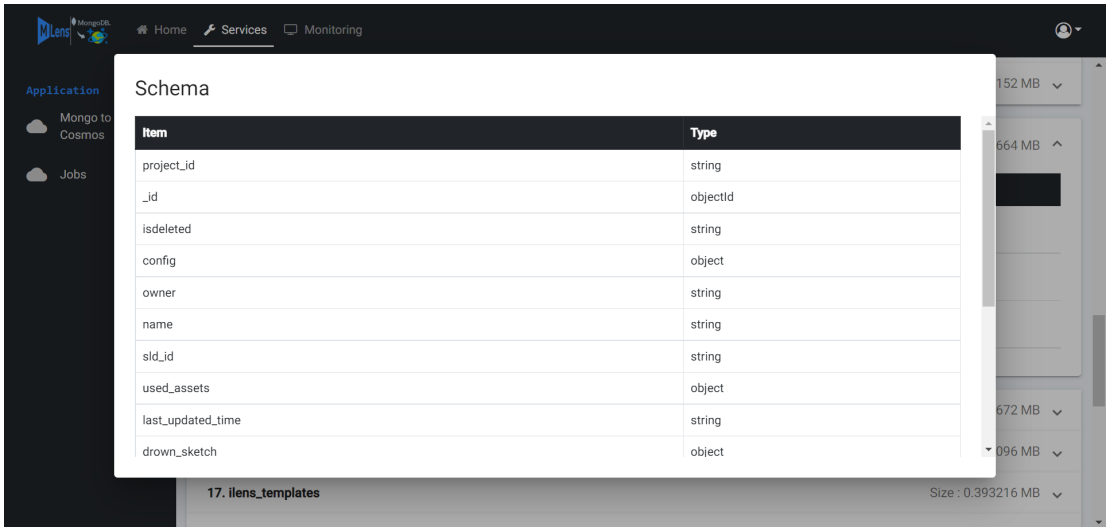


Figure 3.7: Collection Schema

8. Jobs Status

- Were the Migration processing the Report will see with the Job status details as completed, pending and failed
- The user can reinitialise the failed jobs again

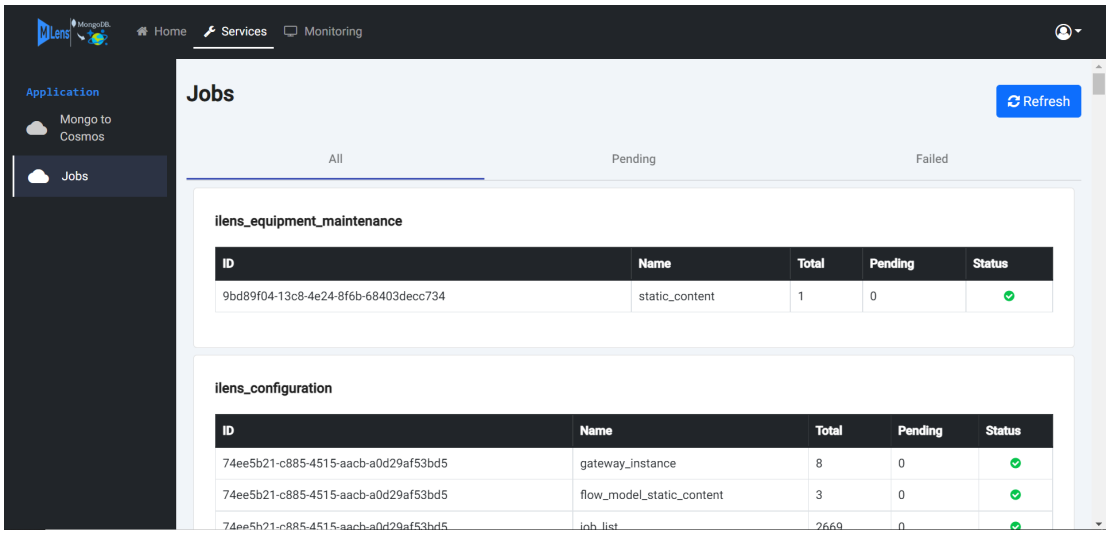


Figure 3.8: Job listing

Chapter 5

CONCLUSION

The project aimed to reduce user effort in monitoring and controlling pollution in accordance with industrial rules and regulations. The first attempt was made to determine the need for the system. To meet the needs of the users, a detailed study was created in such a way that it is user friendly and simple to use. This particular system has been designed in such a way that even a user with limited knowledge can easily operate it.

5.1 Future Enhancement

The system is designed in such a way that adding new modules is relatively simple. The system's flexibility will be increased as the system is rebuilt. Keeping in mind the advanced features of this technology, the system has been designed to be as versatile and user friendly as possible. All of the SRS requirements have now been implemented in the system.

Data migration between RDBMS and NoSQL databases is a difficult task because data is constantly updated and growing in size. In the future, more database options for migration will be added, including SQL and NoSQL.

References

[1] Basant Namdeo, Ugrasen Suman - *Á Model for Relational to NoSQL database Migration: Snapshot-Live Stream Db Migration Model* - 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)

[2] Yansyah Saputra Wijaya, Arry Akhmad Arman - *A Framework for Data Migration Between Different Datastore of NoSQL Database* - 2018 International Conference on ICT for Smart Society (ICISS)

[3] Alae EL ALAMI, Mohamed BAHAJ - *Migration of a Relational Databases to NoSQL: The Way Forward* - 2016 5th International Conference on Multimedia Computing and Systems (ICMCS)

[4] Zhao Du, Qian Wang, Naijia Liu - *Design and Realization of Database Online Migration* - 2012 2nd International Conference on Computer Science and Network Technology

[5] Ken Higuchi, Wenqian Wang, Tatsuo Tsuji - *Incremental Data Migration for Multi-database Systems* - 2012 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing

[6] Naoyuki MIYAMOTO, Ken HIGUCHI, Tatsuo TSUJI - *Incremental Data Migration for Multi-Database Systems Based on MySQL with SPIDER Storage Engine* - 2014 IIAI 3rd International Conference on Advanced Applied Informatics

[7] Dongzhao Liang, Yunzhen Lin, Guiguang Ding - *Mid-model Design Used in Model Transition and Data Migration between Relational Databases and NoSQL Databases* - 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity).

[8] Alae El Alami, Mohamed Bahaj - *Framework for a complete migration of relational databases to other types of databases(object oriented OO, object-relational OR, XML)*

- 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)

[9] Nisreen I. Abo Dabowsa, Abdelsalam M. Maatuk, Salwa M. Elakeili, M. Akhtar Ali - *Converting Relational Database to Document-Oriented NoSQL Cloud Database* - 2021 IEEE 1st International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering MI-STA.

[10] K. Munir, M. Waseem Hassan, A. Ali, R. McClatchey, I. Willers - *Database independent migration of objects into an object-relational database* - The 2nd International Workshop on Autonomous Decentralized System, 2002.

[11] Qian Wang; Naijia Liu; Chun Yu - *Research and Practice of university database Migration* - 2012 International Symposium on Information Technologies in Medicine and Education

APPENDIX

Screenshots

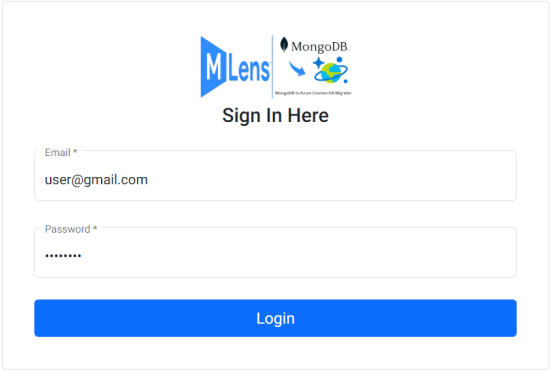


Figure A.1: Login Page

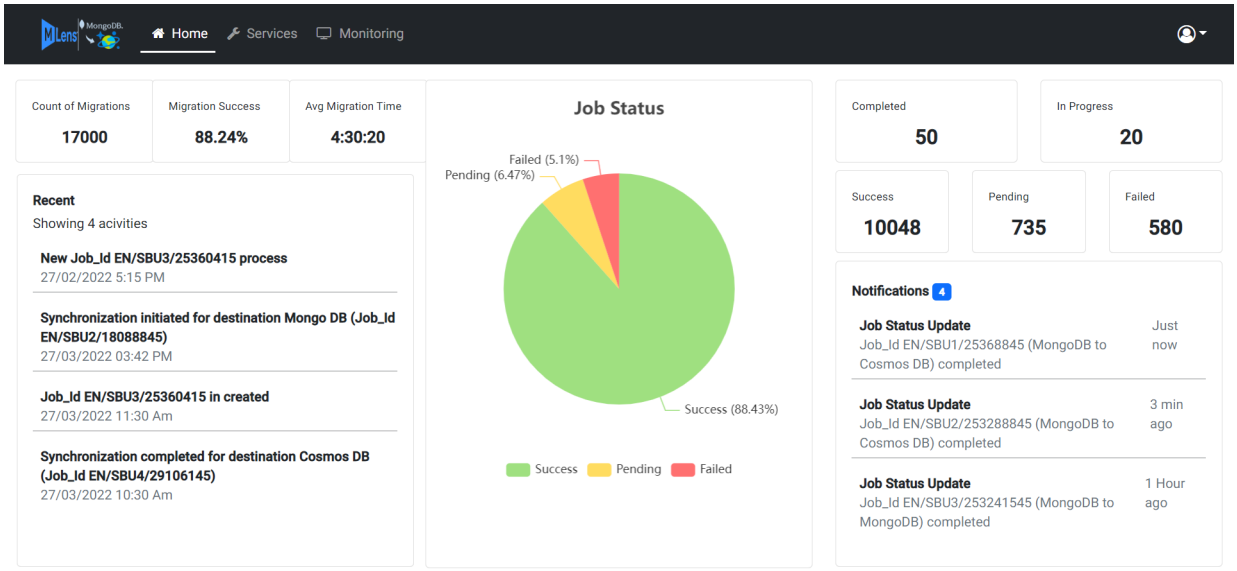


Figure A.2: Dashboard

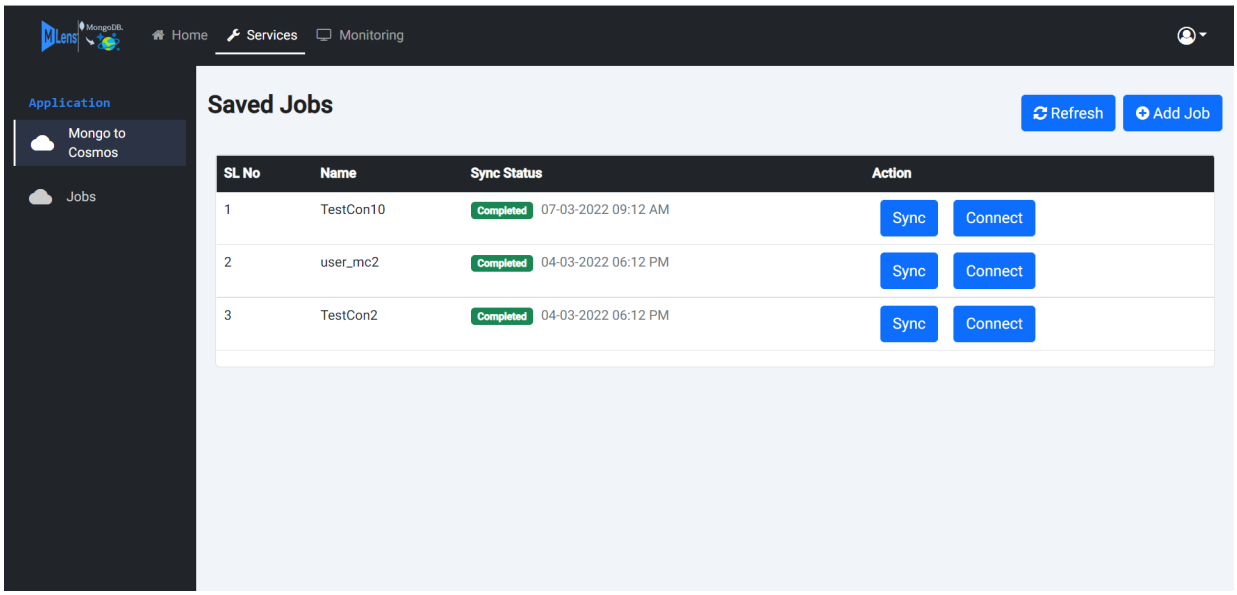


Figure A.3: Services

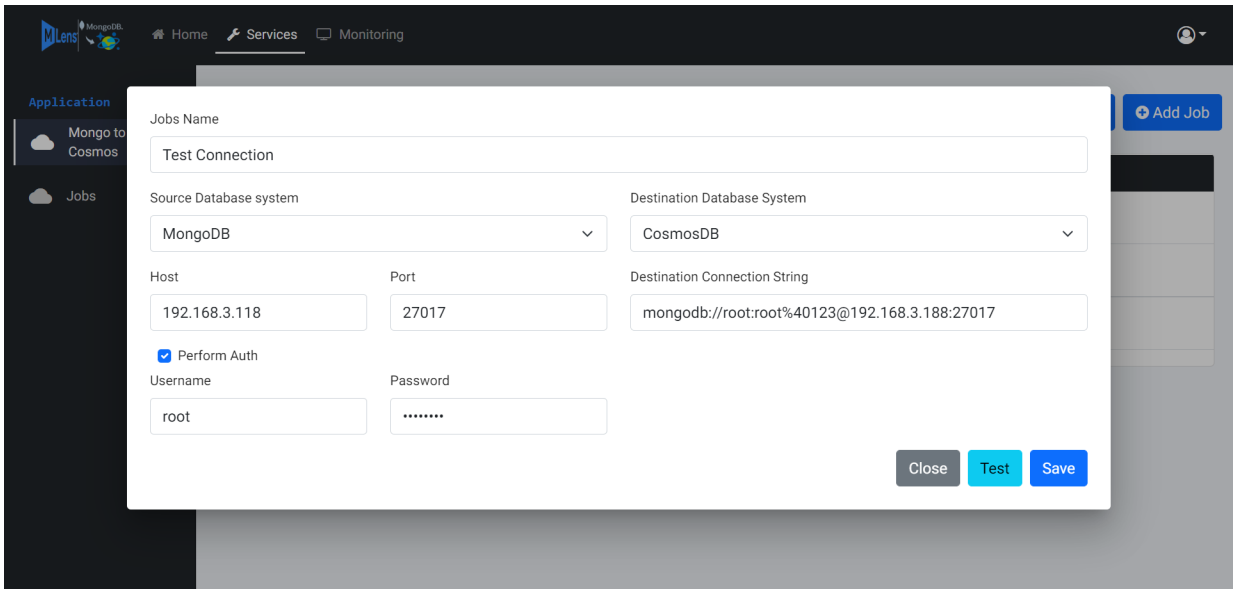


Figure A.4: Add Job

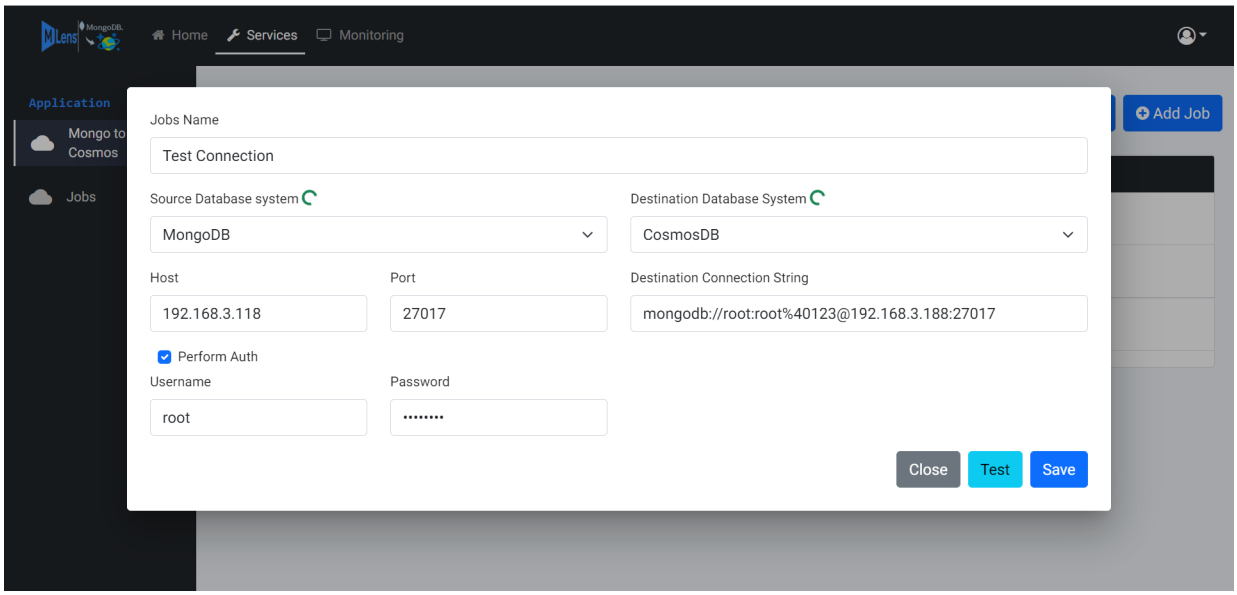


Figure A.5: DB Connection Checking

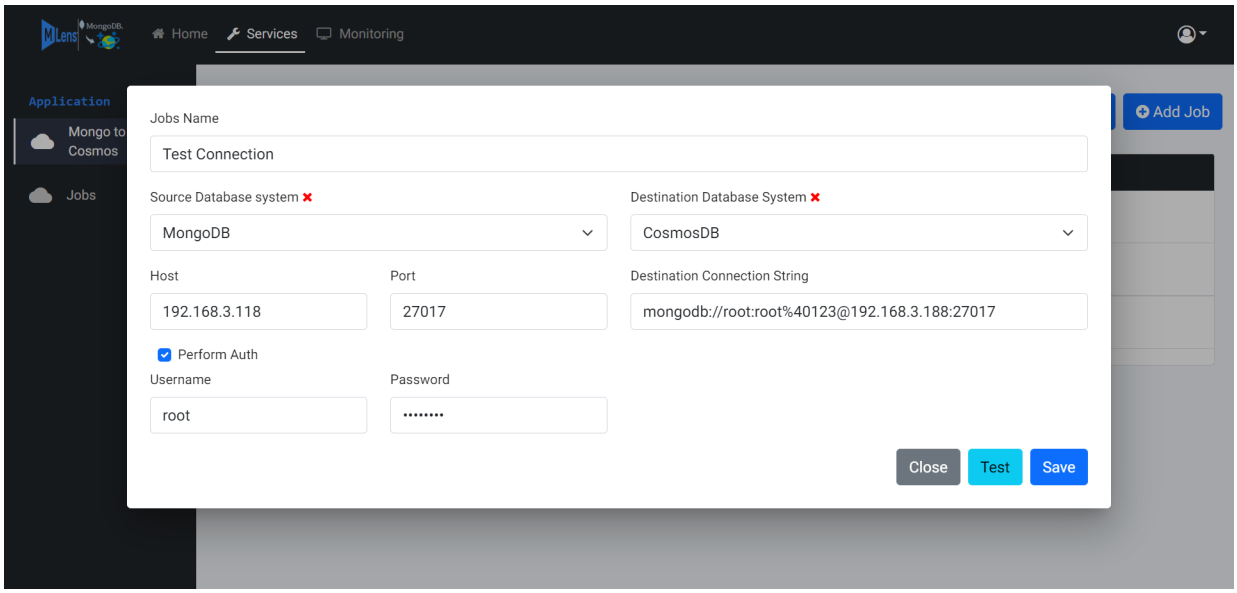


Figure A.6: DB connection failed

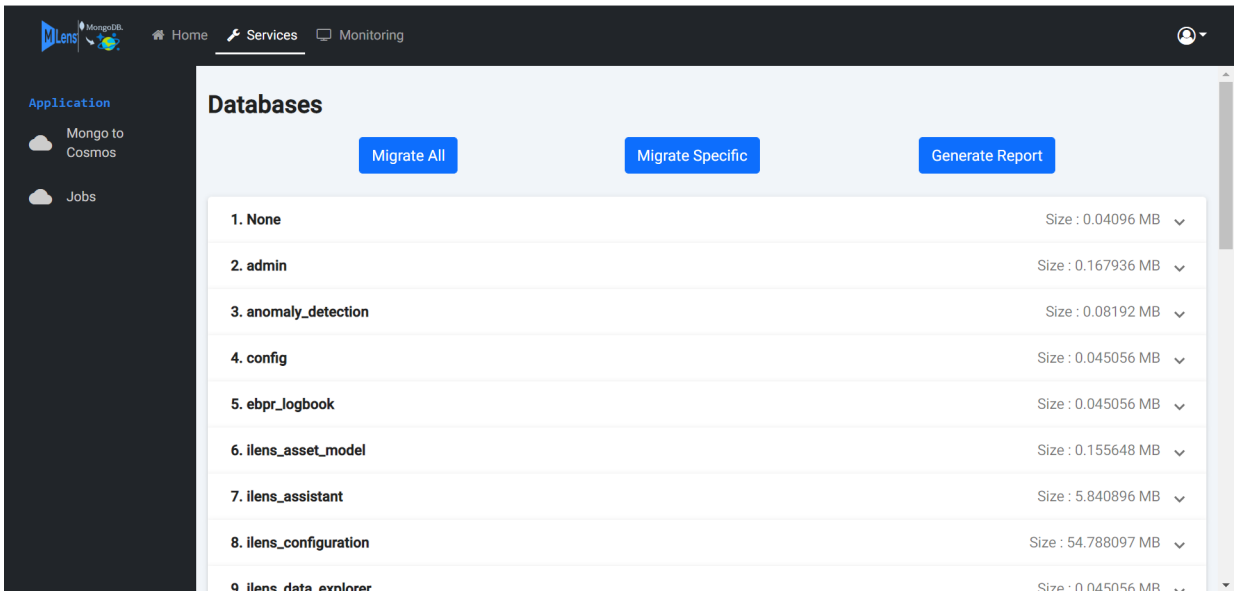


Figure A.7: Database Listing

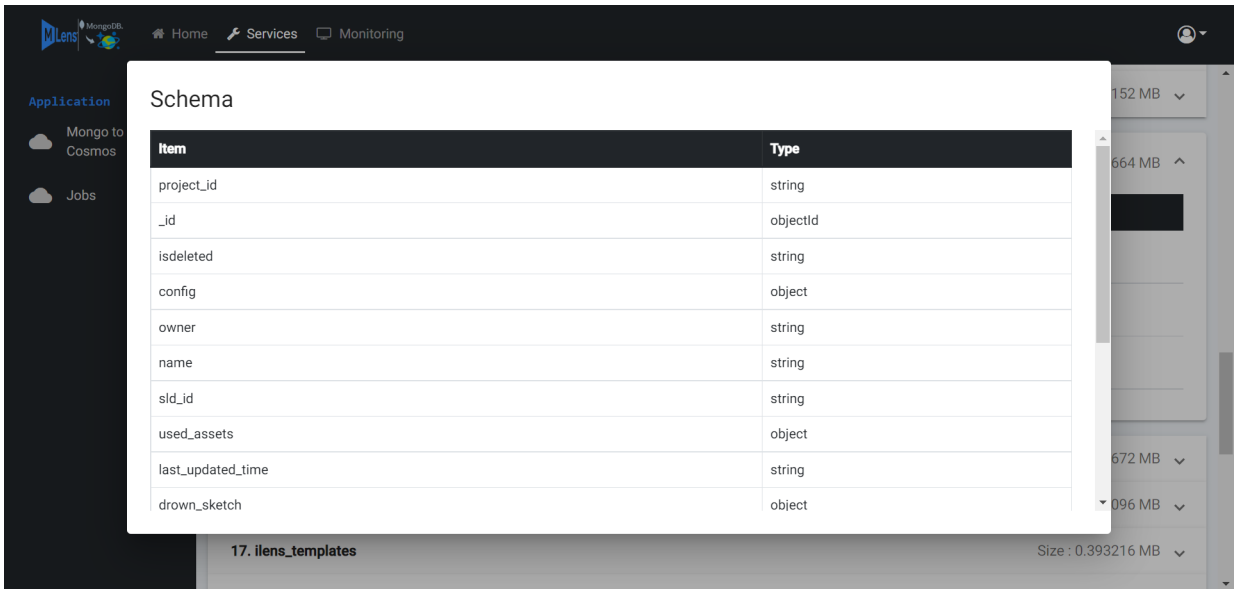


Figure A.8: Collection Schema

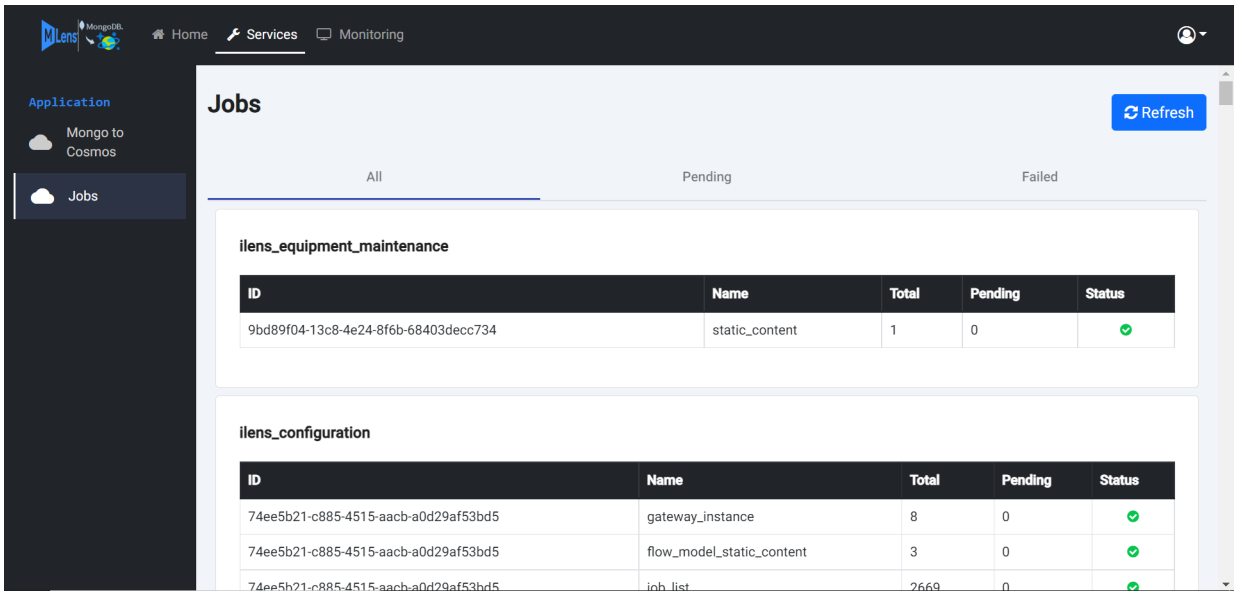


Figure A.9: Jobs Status

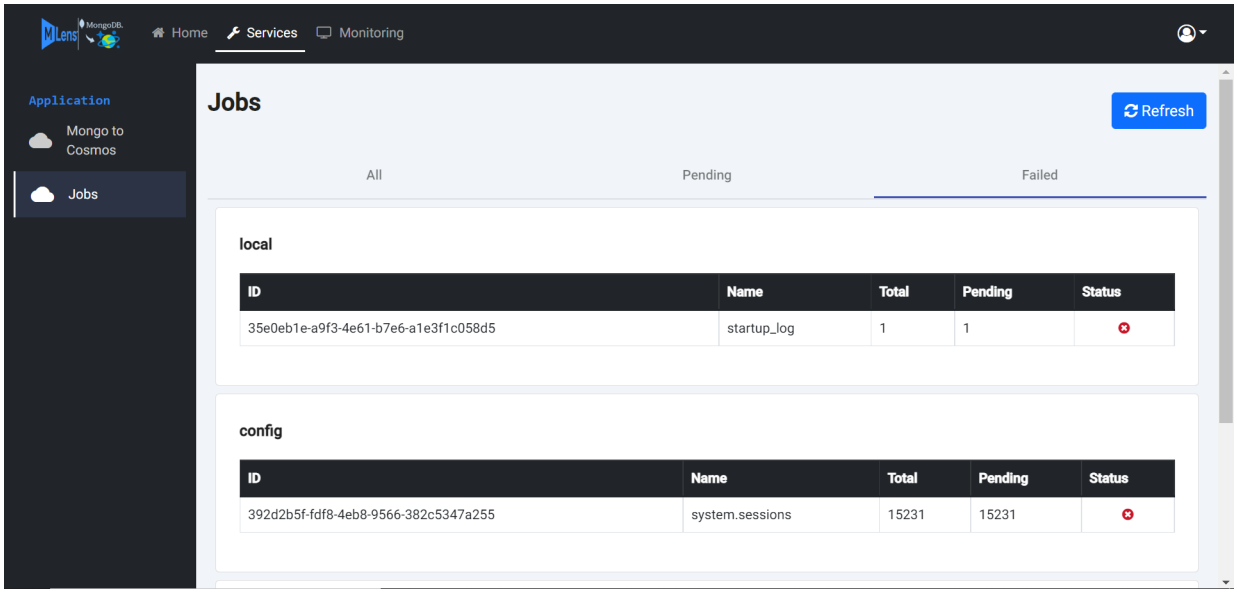


Figure A.10: Jobs failed