NextGen Data Management Information System

Comprehensive Technical Documentation

EMMANUEL EDUBE GULOBA
S23B38/005
b24690@students.ucu.ac.ug

ASINGWIIRE ENOCH
M23B38/027
b22983@students.ucu.ac.ug

EMMANUEL NSUBUGA
M23B38/005
b20811@students.ucu.ac.ug

Uganda Christian University

Faculty of Engineering, Design and Technology
Department of Computing
NextGen Data Architects

Last Updated: November 2025

Contents

1 System Overview **3**

2 System Architecture **3**
- 2.1 High-Level Architecture 3
- 2.2 Data Flow 4

3 Key Features **4**

4 Machine Learning Models **4**
- 4.1 Random Forest Regressor 4
- 4.2 Gradient Boosting Regressor 4
- 4.3 Neural Network (MLPRegressor) 4
- 4.4 Enhanced Models 5
- 4.5 Training Pipeline 5

5 Technology Stack **5**
- 5.1 Backend 5
- 5.2 Frontend 5
- 5.3 Data Warehouse 5
- 5.4 ETL 5

6 Installation & Setup **5**
- 6.1 Backend 5
- 6.2 Frontend 5

7 API Documentation **6**
- 7.1 Authentication 6
- 7.2 Prediction API 6

8 Data Warehouse Schema **6**
- 8.1 Dimension Tables 6
- 8.2 Fact Tables 6

9 ETL Pipeline **6**

10 Security & RBAC **7**

11 Troubleshooting **7**

12 References **8**

1 System Overview

The **NextGen Data Management Information System** is a full-stack analytics platform designed to support data-driven decision-making at Uganda Christian University (UCU). The system integrates:

- A **multi-layer ETL pipeline** (Bronze–Silver–Gold)
- A **PostgreSQL star-schema data warehouse**
- A **Flask API backend**
- A modern **React.js frontend**
- **Machine learning models** for performance prediction
- **Role-Based Access Control (RBAC)**
- **High-performance dashboards and analytics**

The platform unifies academic, financial, attendance, and demographic data to provide real-time, predictive and actionable insights for students, lecturers, Heads of Departments (HODs), Deans, Senate, and Finance.

2 System Architecture

2.1 High-Level Architecture

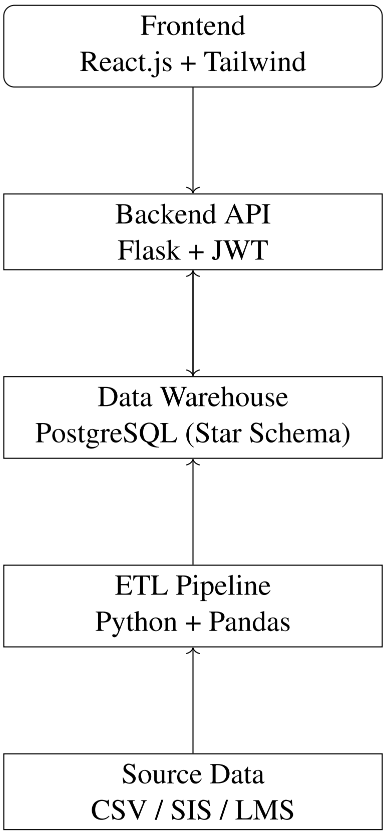

Figure 1: System Architecture

2.2 Data Flow

1. ETL extracts raw data into the Bronze layer.
2. Cleaning, validation, normalization occur at Silver.
3. Aggregations, fact tables, and dimensions generated for Gold.
4. Backend API queries the warehouse.
5. Frontend visualizes analytics.

3 Key Features

Multi-role dashboards:

- Student: personal analytics
- Staff: class analytics
- HOD: department insights
- Dean: faculty analytics
- Senate: university-wide analytics
- Finance: payment dashboards

Predictive analytics:

- Machine learning predictions for academic performance
- Tuition-attendance-performance correlation
- Scenario-based simulations

Data Export: Excel, CSV, PDF (ReportLab).

4 Machine Learning Models

Three primary regression models are implemented:

4.1 Random Forest Regressor

Robust to noise, handles nonlinear patterns, ideal baseline model.

4.2 Gradient Boosting Regressor

Most accurate model (RMSE = 0.59). Best for predictive stability.

4.3 Neural Network (MLPRegressor)

Learns complex non-linear relationships.

4.4 Enhanced Models

- Tuition–Attendance–Performance Model
- Enrollment Trend Model
- Foundational Course Performance Model

4.5 Training Pipeline

1. Feature engineering
2. Scaling
3. Cross-validation
4. Model persistence (pickle)

5 Technology Stack

5.1 Backend

Flask, SQLAlchemy, pandas, NumPy, scikit-learn, JWT, ReportLab.

5.2 Frontend

React.js, Tailwind CSS, SciChart.js, Axios.

5.3 Data Warehouse

PostgreSQL 8.0, star-schema dimensional modelling.

5.4 ETL

Python, Pandas, PyArrow, SQLAlchemy.

6 Installation & Setup

6.1 Backend

```
cd backend
python -m venv .venv
source .venv/bin/activate
pip install -r requirements.txt
python setup_databases.py
python etl_pipeline.py
python train_models.py
python app.py
```

6.2 Frontend

```
cd frontend
npm install
npm start
```

7 API Documentation

7.1 Authentication

```
POST /api/auth/login
{
 ”identifier”: ”dean”,
 ”password”: ”dean123”
}
```

7.2 Prediction API

```
POST /api/predictions/predict
{
 ”student_id”: ”J21B05/001”,
 ”model_type”: ”ensemble”
}
```

8 Data Warehouse Schema

8.1 Dimension Tables

Student, Course, Program, Faculty, Time, Semester.

8.2 Fact Tables

Enrollment, Attendance, Payment, Grade.

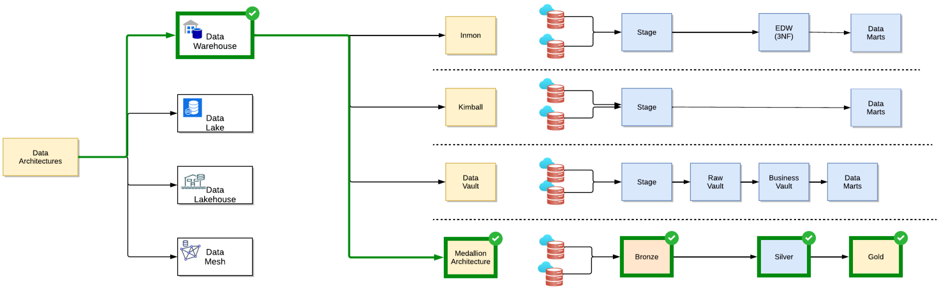

Figure 2: Star Schema Overview

9 ETL Pipeline

Flow:

1. Extract (CSV)
2. Transform (cleaning, normalization)
3. Load (PostgreSQL Star Schema)

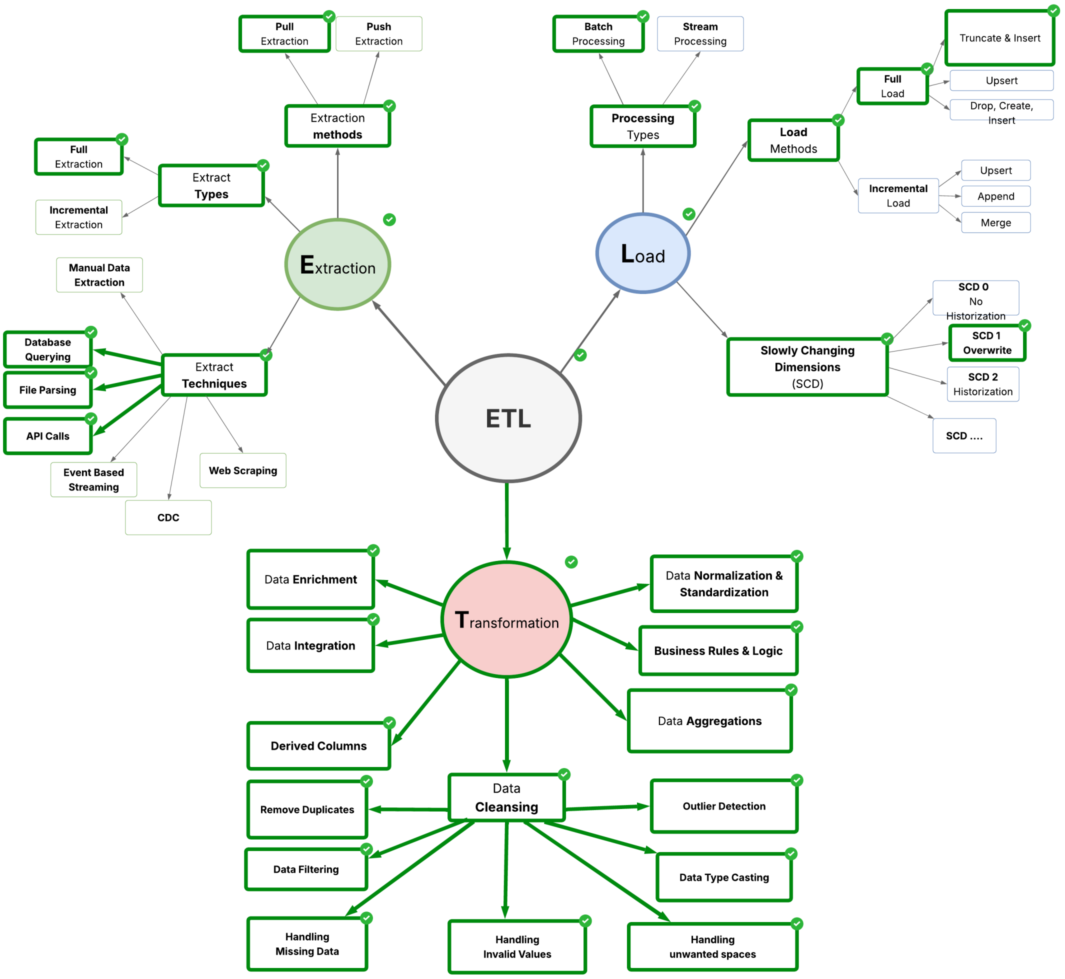

Figure 3: ETL Pipeline

10 Security & RBAC

Roles: Student, Staff, HOD, Dean, Senate, Finance.

Mechanisms:

- JWT authentication
- Password hashing (bcrypt)
- Role-based data scoping
- CORS

11 Troubleshooting

Model not trained:

```
python train_models.py
```

Database errors: Check PostgreSQL credentials and running status.

12 References

- Flask Documentation — https://flask.palletsprojects.com
- React.js — https://react.dev
- scikit-learn — https://scikit-learn.org
- MySQL — https://mysql.com
- Pandas — https://pandas.pydata.org
- SciChart.js — https://www.scichart.com
- Postgresql - https://www.postgresql.org/docs/