

USE CASE STUDY REPORT

Group No.: 2

Student Names: Justin Kim and Abhishek Jami

Executive Summary:

This study presents SmartRefi, a data driven system designed to support student loan refinancing that incorporates academic performance into interest rate determination. Traditional refinancing platforms rely mainly on credit based financial indicators and treat student borrowers as a uniform group. This approach overlooks academic performance, which can reflect responsibility and future earning potential. SmartRefi addresses this limitation by integrating GPA history, lender rate tiers, loan products, applications, offers, and refinance contracts into a unified database that supports more personalized pricing.

The project applies the full data management workflow required for the course. It begins with the development of conceptual data models using both EER and UML approaches. These models capture the entities involved in the refinance process and the relationships among them. The models are then mapped into a relational schema that defines tables, keys, constraints, and cardinalities. The schema is implemented in MySQL and populated with realistic sample data that reflects common refinance scenarios. The system supports SQL queries that retrieve student information, analyze GPA based pricing, trace refinance contract lineages, and measure lender performance.

A partial NoSQL implementation is demonstrated in MongoDB to evaluate how the SmartRefi structure behaves in a document based environment. Although the relational model is more suitable for enforcing constraints and maintaining consistent relationships across the refinance pipeline, the NoSQL prototype provides insight into flexible data representation.

The project concludes with a Python application that connects to the MySQL database, retrieves data, and performs simple analytics. These analytics include GPA and pricing comparisons, weighted interest rate improvements from refinancing, and lender level performance summaries. The results show how the SmartRefi system can provide meaningful insights for students and lenders.

SmartRefi demonstrates that academic performance can be incorporated into refinance pricing in a structured and reliable way. The completed system provides a foundation for future improvements, such as integrating more advanced analytics, expanding the NoSQL implementation, and strengthening data governance measures. Overall, the study achieves the goal of designing a complete end to end data management solution for a GPA based refinancing service.

I. Introduction

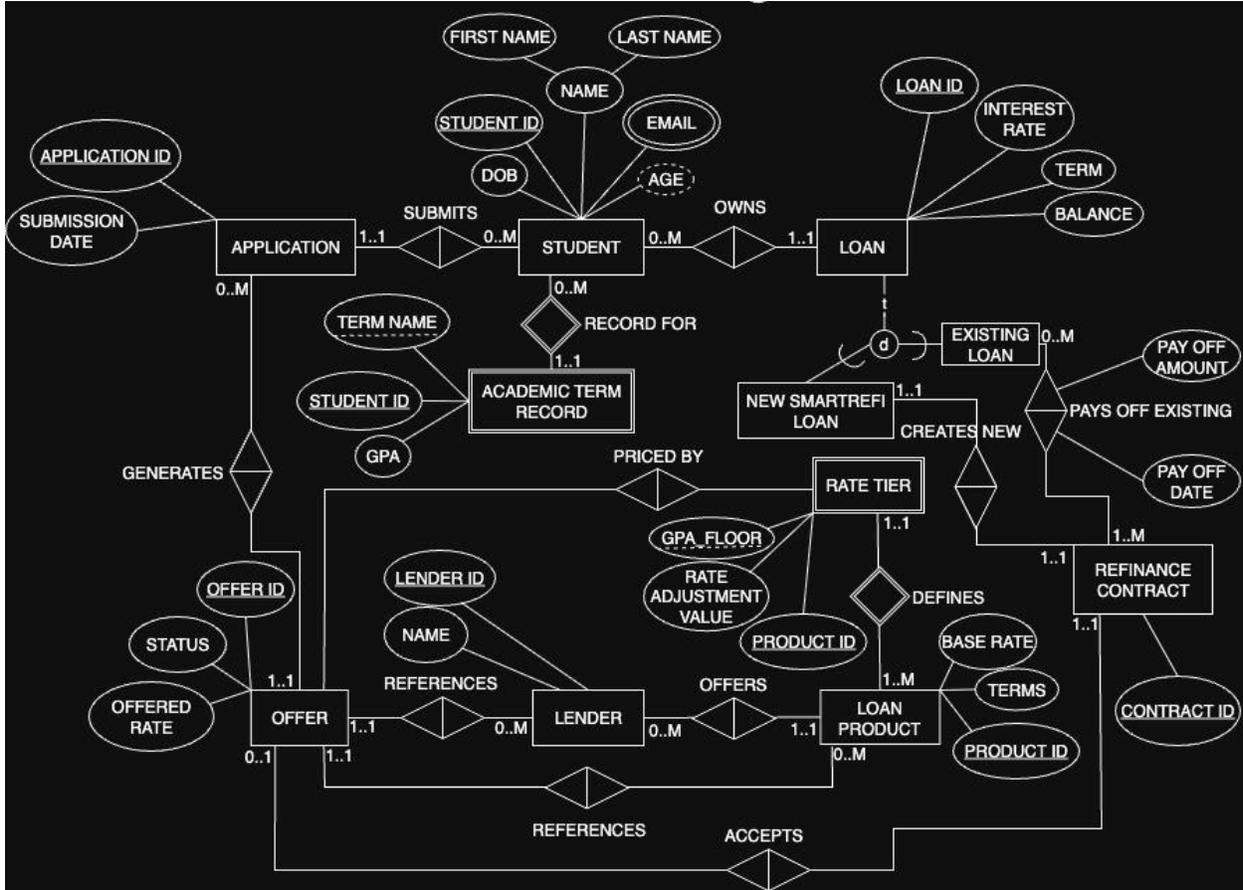
Student loan refinancing has become an important tool for graduates who want to lower their interest rates and reduce long term repayment costs. Private lenders rely on credit based financial indicators to assess risk, but these indicators overlook academic performance, which can reflect responsibility and future earning potential. As a result, students with strong academic records often receive the same refinancing offers as students with weaker performance, and lenders miss an opportunity to improve how they segment borrowers.

SmartRefi addresses this gap by building a data driven system that incorporates academic performance into the refinance process. The goal of this study is to design and implement a complete data management solution that supports GPA based pricing. This includes creating conceptual models, mapping those models into relational structures, implementing the database in MySQL, testing a NoSQL alternative, and developing a Python application for data retrieval and basic analytics. The system must track students, loans, GPA history, applications, lender products, rate tiers, offers, and refinance contracts in a consistent and integrated way.

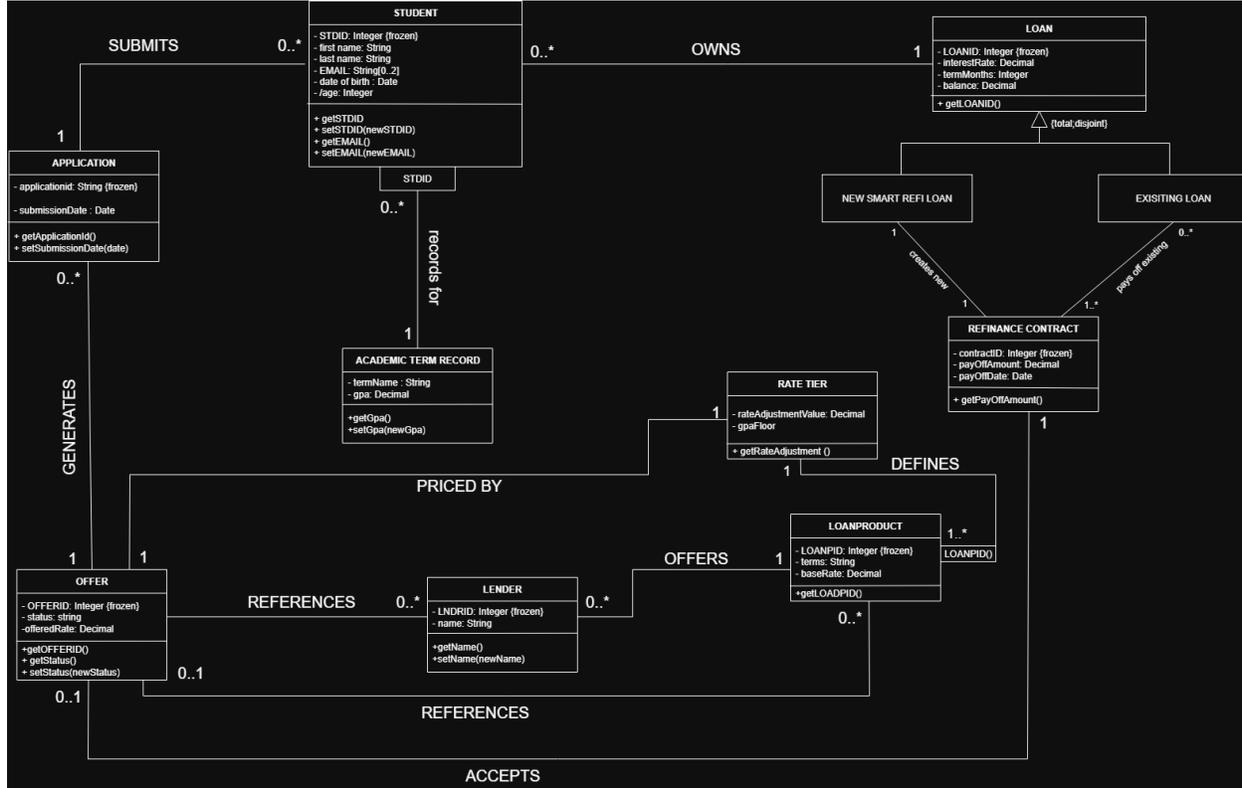
To accomplish this goal, the project follows the requirements defined in the initial proposal. A student may hold multiple loans and submit multiple applications. Each application must reference at least one academic record used for pricing. Lenders provide loan products and GPA based rate tiers. Applications may receive multiple offers, but only one offer may be accepted and converted into a refinance contract. A contract produces a new loan and can pay off several existing loans. These requirements guide the data modeling and ensure that the system can support accurate pricing, full lineage of the refinance process, and meaningful analytics for both lenders and students.

II. Conceptual Data Modeling

1. EER Diagram



2. UML Diagram



III. Mapping Conceptual Model to Relational Model

STUDENT(Student_ID, First_Name, Last_Name, Email, DOB)
 Primary key: Student_ID

LOAN(Loan_ID, *Student_ID*, *Lender_ID*, *Product_ID*, Interest_Rate, Term_Months, Balance)
 Primary key: Loan_ID
 Student_ID foreign key → STUDENT(Student_ID), NOT NULL
 Lender_ID foreign key → LENDER(Lender_ID), NULL ALLOWED
 Product_ID foreign key → LOAN_PRODUCT(Product_ID), NULL ALLOWED

APPLICATION(Application_ID, *Student_ID*, Submission_Date, Status)
 Primary key: Application_ID
 Student_ID foreign key → STUDENT(Student_ID), NOT NULL

ACADEMIC_TERM_RECORD(*Student_ID*, Term_Name, GPA)
 Primary key (composite): (Student_ID, Term_Name)
 Student_ID foreign key → STUDENT(Student_ID), NOT NULL

LENDER(Lender_ID, Name)

Primary key: Lender_ID

LOAN_PRODUCT(Product_ID, Lender_ID, Name, Base_Rate, Terms)

Primary key: Product_ID

Lender_ID foreign key → LENDER(Lender_ID), NOT NULL

RATE_TIER(GPA_Floor, Product_ID, Rate_Adjustment_Value)

Primary key (composite): (GPA_Floor, Product_ID)

Product_ID foreign key → LOAN_PRODUCT(Product_ID), NOT NULL

OFFER(Offer_ID, Application_ID, Lender_ID, Product_ID, Offered_Rate, Status)

Primary key: Offer_ID

Application_ID foreign key → APPLICATION(Application_ID), NOT NULL

Lender_ID foreign key → LENDER(Lender_ID), NOT NULL

Product_ID foreign key → LOAN_PRODUCT(Product_ID), NOT NULL

REFINANCE_CONTRACT(Contract_ID, Offer_ID, Signed_Date, Disbursed_Amount, New_Loan_ID)

Primary key: Contract_ID

Offer_ID foreign key → OFFER(Offer_ID), NOT NULL, UNIQUE (1:1)

New_Loan_ID foreign key → LOAN(Loan_ID), NOT NULL, UNIQUE

REFINANCE_ITEM(Contract_ID, Old_Loan_ID, Payoff_Amount, Payoff_Date)

Primary key (composite): (Contract_ID, Old_Loan_ID)

Contract_ID foreign key → REFINANCE_CONTRACT(Contract_ID), NOT NULL

Old_Loan_ID foreign key → LOAN(Loan_ID), NOT NULL

IV. Implementation of Relation Model via MySQL and NoSQL

1. MySQL

-- Show all refinance contracts with full lineage

```
SELECT
    rc.Contract_ID,
    s.First_Name,
    s.Last_Name,
    o.Offer_ID,
    ldr.Name AS Lender,
    lp.Name AS Product,
    rc.Signed_Date,
    rc.Disbursed_Amount,
    newl.Interest_Rate AS New_Rate,
    newl.Balance AS New_Balance
```

```

FROM REFINANCE_CONTRACT rc
JOIN OFFER o          ON rc.Offer_ID = o.Offer_ID
JOIN APPLICATION a    ON o.Application_ID = a.Application_ID
JOIN STUDENT s        ON a.Student_ID = s.Student_ID
JOIN LOAN newl        ON rc.New_Loan_ID = newl.Loan_ID
JOIN LENDER ldr        ON newl.Lender_ID = ldr.Lender_ID
JOIN LOAN_PRODUCT lp  ON newl.Product_ID = lp.Product_ID
ORDER BY rc.Contract_ID;

```

Contract_ID	First_Name	Last_Name	Offer_ID	Lender	Product	Signed_Date	Disbursed_Amou...	New_Rate	New_Balance
1	Aiden	Park	1	Evergreen Federal	Evergreen Refi 5yr Fixed	2025-03-01	14000.00	4.650	14000.00
2	Maya	Chen	3	Cascade Student Finance	Cascade Smart 5yr Fixed	2025-03-05	23000.00	4.800	23000.00
3	Ethan	Patel	8	NorthStar Bank	NorthStar Choice 15yr	2025-03-12	31000.00	5.200	31000.00
4	Liam	Martinez	11	Cascade Student Finance	Cascade Smart 5yr Fixed	2025-03-20	25000.00	4.700	25000.00
5	Emma	Lopez	16	Evergreen Federal	Evergreen Refi 10yr Fixed	2025-03-25	26000.00	4.950	26000.00
6	Charlotte	Baker	20	Evergreen Federal	Evergreen Refi 5yr Fixed	2025-03-28	12000.00	4.550	12000.00
7	Benjamin	Diaz	21	NorthStar Bank	NorthStar Choice 10yr	2025-04-01	20000.00	4.900	20000.00
8	Henry	Lee	24	NorthStar Bank	NorthStar Choice 15yr	2025-04-08	27000.00	5.100	27000.00
9	Isabella	Thomas	26	Harbor Capital	Harbor Flex 10yr	2025-04-12	19000.00	5.350	19000.00
10	Zoe	Morales	31	Harbor Capital	Harbor Flex 15yr	2025-04-18	33000.00	5.400	33000.00

2. NoSQL - MongoDB

```

-- Analyze GPA impact on offered rates
-- Join academic records with offers and rate tiers to see pricing
alignment.
SELECT
    s.Student_ID,
    s.First_Name,
    s.Last_Name,
    atr.GPA,
    lp.Name AS Product,
    lp.Base_Rate,
    o.Offered_Rate,
    (o.Offered_Rate - lp.Base_Rate) AS Rate_Diff
FROM ACADEMIC_TERM_RECORD atr
JOIN STUDENT s          ON atr.Student_ID = s.Student_ID
JOIN APPLICATION a      ON s.Student_ID = a.Student_ID
JOIN OFFER o            ON a.Application_ID = o.Application_ID
JOIN LOAN_PRODUCT lp    ON o.Product_ID = lp.Product_ID
WHERE atr.Term_Name = 'Spring2025'
ORDER BY atr.GPA DESC;

```

Student_ID	First_Name	Last_Name	GPA	Product	Base_Rate	Offered_Rate	Rate_Diff
6	Olivia	Johnson	3.85	Evergreen Refi 5yr Fixed	4.250	4.800	0.550
16	Isabella	Thomas	3.80	Harbor Flex 10yr	4.800	4.950	0.150
20	Zoe	Morales	3.75	Harbor Flex 15yr	5.100	5.400	0.300
20	Zoe	Morales	3.75	Pioneer Advantage 10yr	4.900	4.950	0.050
3	Noah	Garcia	3.70	Pioneer Advantage 5yr	4.400	4.750	0.350
3	Noah	Garcia	3.70	Pioneer Advantage 10yr	4.900	5.050	0.150
18	Mila	Rodriguez	3.65	Cascade Smart 15yr Fixed	5.000	5.300	0.300
9	Lucas	Rivera	3.60	Harbor Flex 10yr	4.800	5.050	0.250
1	Aiden	Park	3.55	Evergreen Refi 5yr Fixed	4.250	4.650	0.400
1	Aiden	Park	3.55	NorthStar Choice 10yr	4.650	4.950	0.300
12	Charlotte	Baker	3.55	Evergreen Refi 5yr Fixed	4.250	4.550	0.300
13	Benjamin	Diaz	3.50	NorthStar Choice 10yr	4.650	4.900	0.250
13	Benjamin	Diaz	3.50	Cascade Smart 5yr Fixed	4.500	4.750	0.250
5	Ethan	Patel	3.45	NorthStar Choice 15yr	4.950	5.200	0.250
5	Ethan	Patel	3.45	Harbor Flex 10yr	4.800	5.100	0.300
8	Ava	Kim	3.40	Pioneer Advantage 5yr	4.400	4.650	0.250
8	Ava	Kim	3.40	Pioneer Advantage 10yr	4.900	4.950	0.050
14	Amelia	Gonzalez	3.35	Pioneer Advantage 10yr	4.900	5.000	0.100
2	Maya	Chen	3.25	Cascade Smart 5yr Fixed	4.500	4.800	0.300
2	Maya	Chen	3.25	Cascade Smart 15yr Fixed	5.000	5.300	0.300
15	Henry	Lee	3.20	NorthStar Choice 15yr	4.950	5.100	0.150
15	Henry	Lee	3.20	Pioneer Advantage 5yr	4.400	4.700	0.300
10	Emma	Lopez	3.15	Evergreen Refi 10yr Fixed	4.750	4.950	0.200
10	Emma	Lopez	3.15	Cascade Smart 15yr Fixed	5.000	5.250	0.250
17	Jacob	Hernandez	3.10	Evergreen Refi 10yr Fixed	4.750	5.000	0.250
17	Jacob	Hernandez	3.10	NorthStar Choice 15yr	4.950	5.200	0.250
19	Daniel	Wright	3.10	Evergreen Refi 5yr Fixed	4.250	4.700	0.450
4	Sophia	Nguyen	3.05	Evergreen Refi 10yr Fixed	4.750	5.050	0.300
11	James	Singh	3.00	Harbor Flex 15yr	5.100	5.550	0.450
11	James	Singh	3.00	Evergreen Refi 5yr Fixed	4.250	4.850	0.600
7	Liam	Martinez	2.95	Cascade Smart 5yr Fixed	4.500	4.700	0.200
7	Liam	Martinez	2.95	NorthStar Choice 15yr	4.950	5.150	0.200

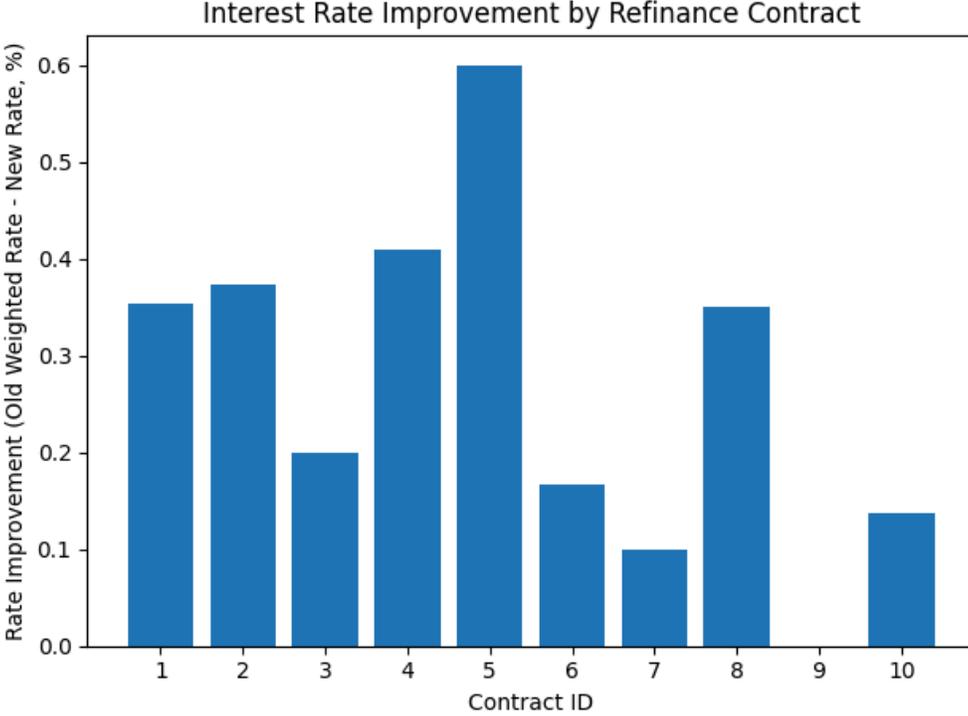
V. Database Access via R or Python

The Python implementation demonstrates how the SmartRefi database can be accessed programmatically to support data retrieval and basic analytics. Using a simple MySQL connector and structured query functions, the application reads data from the relational schema and performs analyses that highlight GPA based pricing, refinance savings, and lender performance. This section shows how the database can be integrated into an external application environment, validating that the system supports practical analytical workflows beyond direct SQL execution.

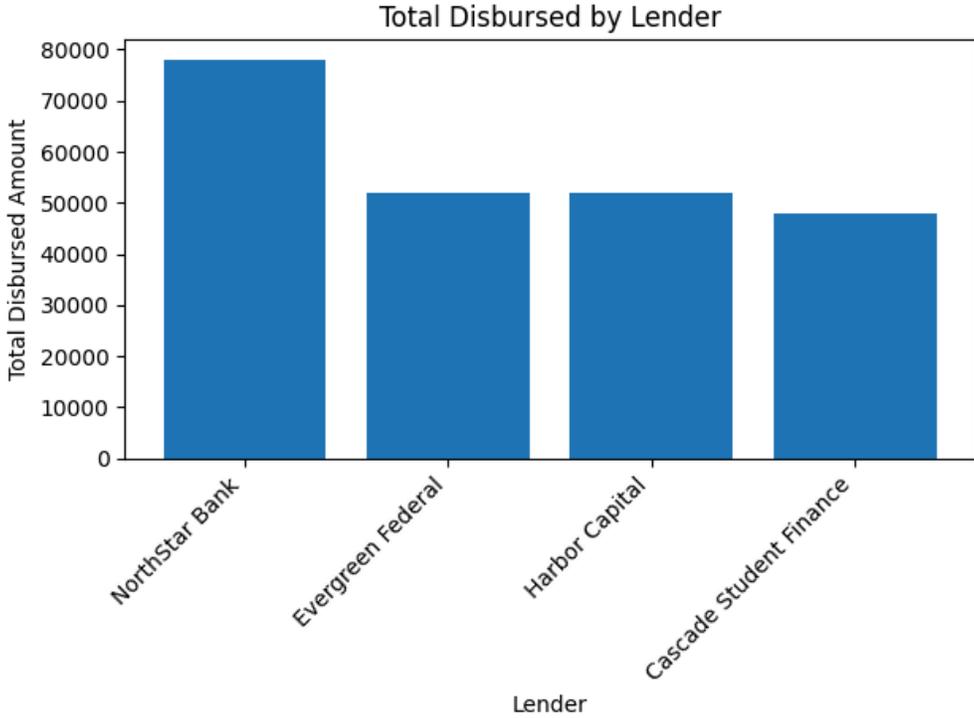
Visualization 1 - GPA Pricing



Visualization 2 - Refinance Savings



Visualization 3 - Lender Performance



VII. Summary and Recommendation

This study designed and implemented SmartRefi, a complete data management system that supports a GPA based student loan refinancing model. The project covered the full pipeline from business problem definition to conceptual data modeling, relational mapping, database implementation, NoSQL prototyping, and Python based analytics. The results show that academic performance can be integrated into a refinancing workflow in a structured and reliable way, and that the system can support the pricing logic and analytics required for lenders and students.

The relational model proved to be the strongest fit for this use case. It preserved referential integrity across students, applications, offers, contracts, new loans, and payoff items. MySQL allowed accurate joins, lineage tracking, and analysis of GPA based pricing. The analytics produced through the Python application demonstrate the value of the system by showing how GPA affects rates, how contract level savings can be measured, and how lender performance can be compared through disbursement and contract volume.

The NoSQL implementation was useful for exploring alternative data representations, but it highlighted the challenges of enforcing structured relationships in a document based environment. SmartRefi depends on consistent links between many entities in a refinance pipeline, and the relational database supported these requirements more effectively.

Future improvements should focus on expanding analytics capabilities, improving data governance, and preparing the system for real world use. Examples include adding more detailed rate tier logic, implementing audit controls, and developing a front end interface that would allow students and lenders to interact with the system directly. The project can also be extended by applying machine learning to predict rate adjustments or default risk using academic trends.

Overall, SmartRefi meets the project goals and provides a solid foundation for a GPA informed refinancing platform that can support better pricing decisions and a more equitable lending process.