Software Development and Implementation

by

Noshin Tabassum (221014032)

Internship report (CSE4099) submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science in Computer Science and Engineering

Under the supervision of

Raihan Kibria
Senior Lecturer, Department of Computer Science and Engineering

Md.Zakir Hossain
Senior Software Engineer, Synesis IT PLC

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

UNIVERSITY OF LIBERAL ARTS BANGLADESH

SUMMER 26

© *Author*
All rights reserved

DECLARATION

Internship Title	Software Development and Implementation
Author	*Noshin Tabassum*
Student ID	221014032
Supervisor (Academic)	Raihan Kibria
Supervisor (Industry)	Md.Zakir Hossain

I declare that this internship report entitled *Software Development and Implementation* is the result of my own work except as cited in the references. The internship report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Noshin Tabassum
221014032

Department of Computer Science and Engineering
University of Liberal Arts Bangladesh

Date: April 21,2026

Department of Computer Science and Engineering
University of Liberal Arts Bangladesh
Mohammadpur, Dhaka - 1207

CERTIFICATE

This is to certify that the internship report entitled **Software Development and Implementation**, submitted by **Noshin Tabassum** (Student ID: 221014032) is a undergraduate student of the **Department of Computer Science and Engineering** has been examined. Upon recommendation by the examination committee, we hereby accord our approval of it as the presented work and submitted report fulfill the requirements for its acceptance in partial fulfillment for the degree of *Bachelor of Science in Computer Science and Engineering*.

Raihan Kibria, Senior Lecturer
University of Liberal Arts Bangladesh

Md.Zakir Hossain, Senior Software Engineer
Syensis IT PLC

Place: Dhaka
Date: April 21,2026

ACKNOWLEDGEMENTS

I would like to express my deep and sincere gratitude to my academic supervisor Raihan Kibria and my industry supervisor Md.Zakir Hossain, for giving me the opportunity to conduct this internship and providing invaluable guidance throughout this work. His dynamism, vision, sincerity and motivation have deeply inspired me. He has taught me the methodology to carry out the work and to present the works as clearly as possible. It was a great privilege and honor to work and study under his guidance. Also, I would like to thank the other employees of *Synesis IT PLC* for their help and cooperation during my practice time.

I am greatly indebted to our honorable teachers of the Department of Computer Science and Engineering at the University of Liberal Arts Bangladesh who taught me during the course of our study. Without any doubt, their teaching and guidance have completely transformed us to the persons that we are today.

I am extremely thankful to my parents for their unconditional love, endless prayers and caring, and immense sacrifices for educating and preparing me for the future. I would like to say thanks to my friends and relatives for their kind support and care.

Finally, I would like to thank all the people who have supported me to complete the Internship directly or indirectly.

Noshin Tabassum
University of Liberal Arts Bangladesh
Date: April 21,2026

Dedicated to my loving
parents

– *Noshin Tabassum*

EXECUTIVE SUMMARY

This internship was taken to close the gap between the knowledge gained and the experience in the field of software development. It gave me a chance to put theoretical knowledge into practice in a real-life setting and obtain practical experience in developing a backend system. Internship was carried out in Synesis IT PLC, a multinational ICT firm based in Bangladesh which specializes in providing large-scale government and enterprise solutions especially in the area of e-Governance. The main goals of this internship were to gain real-world experience in backend development with Java and Spring Boot, learn the workflow of a real-world software development process, and learn problem-solving skills. The internship scope was centered around the backend development, database operations and API implementation. The field I was primarily involved with was the backend development, where I was involved in query handling in databases, service layer development, and API response handling. The internship period enabled me to gradually shift between my knowledge of the system and involvement in the development processes. Through the internship, I was able to implement native SQL queries, projection techniques to retrieve data more efficiently, and develop repository and service layer techniques. I also verified Postman API using Postman to make sure that the responses are correct. Such contributions enhanced efficiency in handling data and performance of the system. Various difficulties were witnessed such as dealing with sophisticated queries, solving data mapping problems, and modifying client demands. These difficulties were solved with the help of debugging, thorough examination and constant education. The internship has helped me greatly in improving my technical skills in Java, Spring Boot, and database management, as well as my analytical thinking and debugging skills. Moreover, it also helped me to grow professionally by learning how to work in a team, communicate, and be flexible. In general, the experience of this internship helped me to get a lot of insights of the real-world software development and made me ready to get any job opportunities in the sphere of IT in the future.

Keywords: Backend Development, Enterprise Application, Problem-Solving

Contents

1 Introduction **1**

1.1 Introduction 1

1.2 Objective of the internship report 2

2 Background **3**

2.1 Company Overview 3

2.2 Mission 4

2.3 Vision 5

2.4 Industry Environment and Marketplace 5

2.5 Purpose of the Internship 6

2.6 Role and Responsibilities During the internship 6

3 Scope of Work **8**

4 Internship Activities **10**

4.1 Introduction of the Organization/Institution 10

4.2 Work Accomplishment 11

4.3 Challenges Faced 14

5 Methodology **16**

5.1 Requirement Analysis 16

5.2 System Design and Architecture 16

5.3 Query Development and Database Design 17

5.4 Application of Repository Layer 17

5.5 Use of Projection Technique 17

5.6 Service Layer Implementation 18

5.7 API Development 18

5.8 Testing and Debugging 18
5.9 Performance Optimization 19
5.10 Handling Requirement Changes 19

6 Analysis and Findings **21**
6.1 Results/Findings 21
6.2 Learning and Development 22
6.3 Tools and Technology 23
6.3.1 Tools 23
6.3.2 Libraries and Frameworks 27
6.3.3 Important Functions and Annotation: 28
6.3.4 Projection: 30
6.4 User Guideline 32
6.5 Impact of the Internship 32

7 Conclusions and Recommendations **34**
7.1 Conclusion 34
7.2 Limitations 35
7.3 Recommendations 36

8 Acknowledgments **37**

References **43**

List of Figures

2.1 Logo of Synesis IT 3
2.2 Mission of Synesis 4

4.1 The Services Provided by The Organization 11

5.1 Development Workflow 20

6.1 JDK 17 24
6.2 Spring Boot 25
6.3 Intellij Idea 25
6.4 Oracle 26
6.5 Maven 26
6.6 Postman 27

8.1 Surrender Details Page 38
8.2 Surrender Application Page 38
8.3 Policy Information 39
8.4 Submit Application 39
8.5 Application List 40
8.6 Acknowledge and Reject Option 40
8.7 Surrender Schedule Report 41
8.8 Check and Download 41
8.9 Claim Module 42
8.10 Offer Letter 42

Chapter 1

Introduction

1.1 Introduction

The internship program is an important step in bridging the gap between the basic knowledge of software development and practical use in the real-world. It gave me a chance to gain practical experience in a professional environment, such as the software company Synesis IT PLC. Also, I learned more about the design, development, and maintenance of enterprise-level applications.

In this internship, I actively worked on backend development in Java and the Spring Boot framework. My tasks principally involved the construction and optimization of server-side applications, the processing of database applications, and the construction of efficient APIs to simplify application functionality. I have hands-on experience operating relational databases. Especially Oracle, where I created and optimized SQL queries, as well as complex native queries using Spring Data JPA.

One of the most important parts of my work was the introduction of projection methods to enhance the performance of the application. Only the required information could be obtained by the database. This experience allowed me to clearly see how effective the data management and optimization of the system is in large-scale applications. I also did some work in designing and implementing methods of repository and service layers that enhanced my knowledge of the layered architecture and the practice of clean code.

During the internship, I also encountered and solved real-world problems, including data mapping issues, query mismatches, and dynamic data representation using conditional logic. These difficulties have strengthened my problem-solving

skills and debugging abilities and made me more analytical and confident when facing technical problems. In addition, the internship environment promoted continuous learning and teamwork. I was able to communicate with professionals who had more experience, get constructive feedback, and enhance my technical and communication skills.

Altogether, I can state that the internship was a really enriching experience that helped me grow personally and professionally. It not only gave me the necessary experience and knowledge in the industry, but it also gave me the self-confidence to become a software developer.

1.2 Objective of the internship report

The purpose of this internship is to provide me with real-world exposure to software development and implementation, technical and professional skills. This program aimed to allow me to put my academic knowledge into practice in a professional environment, learn industry practices, and develop the skills to achieve a successful career in software engineering. The main objectives are :

- To develop practical experience in the development of applications at the enterprise level with modern technologies and frameworks.
- To become a good and efficient backend developer using Java and Spring Boot framework, with a focus on scalable and maintainable applications.
- To study and practice database management skills, such as writing effective SQL queries, manipulating relational data, and optimizing database performance.
- To learn and apply the principles of the Spring Data JPA, such as the design of repositories, native queries, and projections, to create an effective data retrieval approach.
- To have the experience of designing and implementing the layered architecture, such as controller, service, and repository, to organize the code and make it easier to maintain.
- To learn good communication and teamwork skills through working with team members and receiving feedback from the team leader.

Chapter 2

Background

2.1 Company Overview

Synesis IT PLC (2006) is one of the leading multinational information and communication technology (ICT) companies in Bangladesh. The company was formed in 2006 and has established a good reputation in providing large-scale and innovative digital solutions to both the public and the private sector. It is especially renowned in its contribution to government projects, particularly in the field of e-Governance. It has come up with systems that enhance the delivery of services to the people and efficiency in their operations.

The organization offers a variety of services, such as software development, system integration, e-health solutions, a fintech platform, and digital transformation services. Synesis IT PLC has a well-trained human resource and high technical capabilities, which have contributed to developing the digital infrastructure in the country.

Synesis IT PLC's certification is internationally recognized standards such as CMMI Level 3, ISO 27001, and ISO 9001, which guarantee quality, security, and process maturity in its operations. The organization has more than 550 talented employees, such as software engineers, business analysts, and IT consultants, who are cooperative in providing quality solutions.

Figure 2.1: Logo of Synesis IT

2.2 Mission

Synesis IT PLC's mission is to provide quality and innovative technology solutions that add value to clients and society. It has the following mission:

- To create and provide stable, scalable, and secure software solutions to both the public and private sectors.
- To assist digital transformation efforts, particularly with government services, with high-tech solutions.
- To achieve high levels of quality, efficiency, and performance in every project and service.
- To achieve constant innovation and the use of modern technologies to address changing client requirements.
- To offer customer-based solutions to improve customer experience and operational performance.
- To play a role in the national development through enhancing digital infrastructure and the delivery of services to the people.
- To ensure good levels of ethics, professionalism, and transparency in all business activities.

Figure 2.2: Mission of Synesis

2.3 Vision

The Synesis IT PLC's vision is to be a leader in the IT sector across the globe by providing innovative and effective solutions. The vision could be defined as follows:

- To become a global IT solutions provider with a strong presence in the world.
- To be innovative by embracing the latest technologies and contemporary development standards.
- To be instrumental in creating a sustainable digital ecosystem in Bangladesh and beyond.
- To deliver excellent services and customer satisfaction.
- To constantly increase organizational capabilities and technical skills.
- To deliver long-term value to stakeholders, such as clients, employees, and society.
- To be a reliable partner in large government and enterprise projects across the globe.

2.4 Industry Environment and Marketplace

The Information Technology (IT) business in Bangladesh is fast developing as the demand of digital services is increasing, the government is taking action, and there is a rise in technological developments. The government is more targeted at Digital Bangladesh and many opportunities are being provided to IT firms to develop and execute digital solutions on a large scale.

In this competitive environment, companies like Synesis IT PLC play a crucial role by delivering innovative and scalable systems, particularly in the public sector. The marketplace is constantly changing and the demand of e-Governance, fintech, and digital healthcare solutions is growing. The industry is characterized by:

- Rapid technological changes.
- Shortage of skilled professionals.
- Raising competition between IT service providers.
- Constant requirement of innovation and optimization of the system.

2.5 Purpose of the Internship

This internship was meant to expose me to the real world of software development, and to fill the gap between theory and practice in the industry. Its purpose was to offer practical experience in the use of current technologies, instruments, and structures applied in enterprise application development.

In this internship, I got to learn the process of developing, maintaining, and designing large-scale applications in a professional setting. It also enabled me to strengthen my technical expertise in the field of backend development with the help of Java and Spring Boot and to discover more about the management of databases and the architecture of a system.

The other significant objective of this internship was to gain problem-solving skills through real-time problems and learn how to debug and solve them effectively. The internship also provided me with the experience of working with a team, communicating, and adjusting to the dynamic needs of clients, which are crucial skills in the software business.

In general, the internship was supposed to prepare me for future professional positions. It will help me learn some practical knowledge, technical skills, and an idea of industry standards and workflows.

2.6 Role and Responsibilities During the internship

Throughout the internship, I received multiple tasks that involved backend development and database operations. My task was to be actively involved in the development and maintenance of application features according to the industry best practices.

My primary roles involved working on the backend development with Java and Spring Boot framework, during which I worked with the business logic and provided the proper functionality of the system. My role was to write and optimize SQLs to read and manipulate data in the database, including writing complex queries with numerous conditions and joins.

I also engaged in using native queries with Spring Data JPA and applied projection techniques to enhance the efficiency of data retrieval. This included having

appropriate mapping between query output and application-level structures. Moreover, I helped to develop repository and service layer methods, which guaranteed the process of communication between the layers of the application. I was involved with API-related work such as receiving requests and creating structured responses.

I was also engaged in testing and debugging. I deployed tools such as Postman to test API endpoints, check responses, and make sure that the system was operating properly. Data mapping, query execution, and application logic issues also came up, and I solved them. Also, I was able to adapt to changes in client requirements by altering the existing features and adding new functionalities without affecting system stability. I used clean code, had the right documentation, and I made sure that my work was in accordance with the project standards.

Altogether, my job demanded a mix of technical implementation, problem-solving, and teamwork that allowed me to acquire valuable experience in developing software in real life.

Chapter 3

Scope of Work

The scope of work during this internship at Synesis IT was mainly backend development and database management in an enterprise application setup. The given tasks were aimed at practical experience in the real-world software development process and at increasing the level of technical expertise in the latest technologies of the backend. I work on a JBC (Jibon-Bima Corporation) project, Bankassurance system. I successfully implemented two modules, such as surrender and claim, which have some submodules. The main points of the scope of work are presented below:

- **Backend Development:** Created and supported backend modules in Java and Spring Boot framework, making sure the business logic and application functionality are implemented correctly.

- **Database Operations:** Used a relational database in Oracle extensively to carry out data retrieval and manipulation with SQL statements, such as SELECT, JOIN, and conditional statements.

- **Native Query Implementation:** Created and implemented complex native SQL queries with the help of Spring Data JPA to address more complex data retrieval needs than the standard query techniques.

- **Native Query Implementation:** Created and implemented complex native SQL queries with the help of Spring Data JPA to address more complex data retrieval needs than the standard query techniques.

- **Repository Layer Development:** Designed and maintained repository interfaces to interact with the database effectively, such as parameterized and custom querying methods.

- **Service Layer Implementation:** The business logic, process data, and coordinate between the controller and repository layers are managed by a service layer.

- **API Development:** Helped construct and manage RESTful APIs to receive customer requests and provide structured responses.To test the response, use Postman.

- **Issue Resolution:** Identified and resolved problems with query execution, data mapping, and application logic. Ensuring that the data is displayed correctly and that the system is reliable.

- **Performance Optimization:** Enhanced efficiency of the system through optimization of queries, minimization of data loading that is not necessary, and improvement of response time.

- **Code Maintenance and Documentation:** Complied with the principles of clean coding, ensured the readability of the code, and documented the features that were implemented in the future.

Chapter 4

Internship Activities

4.1 Introduction of the Organization/Institution

Synesis IT PLC (2006) is a prominent multinational information and communication technology (ICT) company headquartered in Bangladesh. Since 2006, the organization has become one of the most renowned providers of software solutions in the country, offering innovative and massive digital transformation services to the public and private sectors.

The company has been known to have a very strong presence in government projects, especially in the e-Governance sector. It has implemented various mission-critical systems to enhance service delivery to people and e-governance. Synesis IT has already completed more than 200 plus government and enterprise-scale projects. It has made a significant contribution to the digitalization of different sectors, including health, education, finance, agriculture, and local governance.

Synesis IT PLC's certification is internationally recognized standards such as CMMI Level 3, ISO 27001, and ISO 9001, which guarantee quality, security, and process maturity in its operations. The organization has more than 550 talented employees, such as software engineers, business analysts, and IT consultants, who are cooperative in providing quality solutions.The services provided by the organization include:

- e-Governance solutions.
- e-Health systems and telemedicine.
- Digital payments and FinTech.

- Contact center and customer care solutions.
- System integration and software development.

The government projects it has undertaken, such as e-TIN registration, e-Return (tax filing), a national e-payment gateway (ekPay), biometric verification systems, and other tele-health services, have directly influenced millions of its citizens by enhancing the accessibility and efficiency of government services.

Synesis IT PLC works towards the vision of being a globally reputed IT firm and aims at providing value-based solutions by the use of innovation and technology. Its mission focuses on developing sustainable digital solutions that enable individuals and organizations without compromising its high ethical principles, including integrity, excellence, and professionalism.

Overall, Synesis IT PLC plays a vital role in the digital transformation of Bangladesh by developing scalable, secure, and efficient IT systems, especially in collaboration with government initiatives. It is a major participant in the ICT sector in the country, as it contributes to national-level projects.

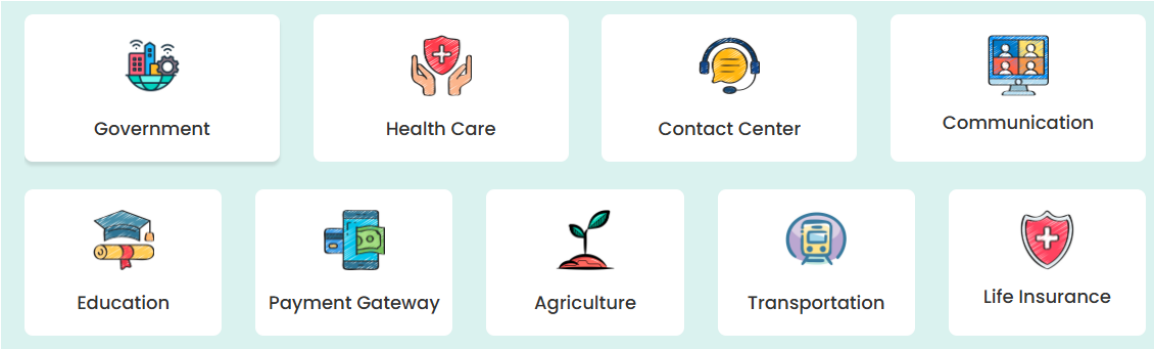

Figure 4.1: The Services Provided by The Organization

4.2 Work Accomplishment

The activities of the work were conducted weekly during the internship period, and I slowly developed my knowledge of the system and started to actively participate

in the development activities. I was actively engaged in backend development and completed various assignments that contributed to the overall system functionality and performance. I completed two modules in the system: the surrender and claim modules for insurance. Below is a summary of my activities, observations, experiences and learning outcomes during the internship.

- **Week 1: Orientation and System Understanding**
  During the first week, I was oriented on the organizational setting, development tools, and project set-up. I have seen the way in which enterprise-level applications are designed with layered architecture, such as controller, service and repository layers. I also examined the available codebase and database schema to get an idea of how data travels within the system.

  One of the challenges that I encountered during this period was the use of a large and complex code structure, unlike small academic projects. I got to know how applications in the real world are designed to be scalable and maintainable. I realized that, in contrast to academic projects, industry level systems need to be well documented, adhere to coding standards and have a modular design.

- **Week 2: Database Analysis and Query Writing**
  The second week was dedicated to the analysis of database tables and SQL query writing. I carried out data retrieval operations using select queries with conditions, and joins. I also got to see how the data is organized in real world systems as well as how various tables are linked with one another.

  I also got to know how to construct effective queries and deal with actual data situations such as null values and filtering condition. I realized that in practice, queries to the database should be optimized to be fast, which is not the focus of a theoretical study.

- **Week 3: Native Query and Projection**
  This week, I began to work with native SQL queries with Spring Data JPA. I used queries to access particular data and projection methods to maximize the performance by choosing necessary fields.

  In the process of doing these, I experienced issues with mapping queries to projection interfaces in terms of data. I discovered the significance of

relating query alias to method names in projection interfaces. This made me realize that minor discrepancies can lead to major problems in real world applications.

- **Week 4: Service Layer and Business Logic Implementation**
  During the fourth week, I was working on the application of service layer techniques. I carried out data processing, application of business logic and management of conditional situations like null values or dynamic field selection.

  I also had some practical problem-solving scenarios where I was required to be able to make sure that the data is represented correctly under various circumstances. I also got to know that business logic is very important in making sure that systems are accurate and reliable.

- **Week 5: API Development and Testing**
  I spent this week doing API-related work. I saw the architecture of APIs and tested them on Postman API Platform to measure the data of requests and responses.

  I tried various API endpoints, response formats, and made sure that information was properly returned by the backend. This made me realize the process of communication between frontend and backend systems in a real-world application.

- **Week 6: Debugging and Issue Resolution**
  During this stage, I was working on debugging and troubleshooting my problems like incorrect data representation and query mismatches. I did some troubleshooting by examining logs, query results and data flow.

  I also had the experience of the significance of systematic debugging and how to find the root cause of the problems. I have learnt that debugging is an important skill in software development, and that it can be quite tedious with a lot of patience and analytical thinking.

- **Week 7: Handling Requirement Changes and Feature Updates**
  The work I did in this week was on modifying the existing features according

to the changing requirements of the clients. I did some queries and business logic updates to support new functionalities.

This experience made me realize that real-world projects are dynamic and requirements can change very fast. I was taught to be flexible and make sure that the new changes do not impact on the functionality of the system.

- **Week 8: Performance Optimization and Final Review**
  The last stage was devoted to system performance optimization, which included query optimization and minimizing unnecessary data retrieval. I also checked my work and confirmed that the implemented features were working properly.

  I came to understand the significance of designing efficient code and ensuring the performance of the systems to enhance user experience.

4.3 Challenges Faced

I faced challenges during the internship that helped me develop problem-solving skills and gain experience in how real-world software development problems are handled. These problems not only tested my technical knowledge but also enhanced my flexibility and critical thinking skills. These are the challenges that I faced:

- **Mapping Issues in Projection:**
  A key problem was to have the appropriate mapping between database query output and the projection interfaces. The incorrect or missing display of data was due to mismatches between the SQL column alias and the method names. This was solved by carefully matching query aliases to projection method names.

- **Complex Native Query handling:**
  It was initially hard to write and maintain complicated native SQL queries that include multiple joins and conditions. It is necessary to have good knowledge of database structure and query optimization methods.

- **Information Not Showing Properly:**
  Sometimes, the occurrence of expected data, like activity logs, was not seen

properly because of errors in queries or mapping. The process of debugging these problems was to ensure that query conditions were checked, the values in the database were checked, and the flow of data between the application layers was also checked.

- **Dealing with Null Values and Conditional Logic:**
  It was necessary to deal with such situations when some of the fields (e.g., new bank name) might be null, so it was necessary to use some conditional logic to guarantee that data is represented properly without compromising system functionality.

- **Client Requirement Changes and Feature Enhancement:**
  One of the major issues was the resistance to adjusting to the evolving client needs in the process of development. Initial implementation involved new features and modifications, which needed to update existing logic, update database queries, and be backward compatible. This experience made me realize the value of flexibility, requirement analysis, and the scalability of the code that could be modified in the future.

- **Error Resolution and Debugging:**
  Systematic debugging and attention to detail were needed to identify the problems with the execution of the query, parameter binding, and logic of the service layer.

- **Understanding Existing Codebase:**
  First, it was hard to know how an already existing industry-level codebase was structured and flowed. It took time to get used to the layered architecture and the code standards employed in the project.

Chapter 5

Methodology

The methodology adopted in this internship was a systematic and structured approach to software development, where every task was done efficiently and in accordance with the project's needs. It was divided into several related steps, among which were requirement analysis, system design, database interaction, implementation, testing, and performance optimization. This methodology contributed to code quality, scalability, and the delivery of accurate results.

5.1 Requirement Analysis

The first stage of the work was the perfect knowledge of the system requirements and expectations of the client. This step was crucial in identifying what features needed to be developed or modified within the application. I analyzed the functional requirements, explored the current system workflow, and examined database structures to learn how information was stored and processed. The step also included learning business regulations, including the impact of various statuses on the data flow, and the way specific fields are to be presented in this or that situation. The process of requirement analysis was done properly to ensure that the implementation phase was correct and matched the desired system behavior.

5.2 System Design and Architecture

Once the requirements were comprehended, the second step was to adhere to a systematic system design, which was a layered architecture. The application was divided into three primary layers: controller, service, and repository. Incoming API

requests and responses were handled by the controller layer. The core business logic was in the service layer and served as an interface between the controller and the repository layers. The repository layer was used to interact directly with the database. This division of responsibility resulted in an application that is more maintainable, scalable, and debuggable.

5.3 Query Development and Database Design

Another important component of the work was the work with the database and the creation of effective SQL queries. I explored the database schema, tables, relationships, and constraints to get an idea of how the data was stored. It is based on this knowledge that I was able to come up with SQL queries to both access and manipulate data based on the requirements of the application. This involved the writing of more complicated queries involving multiple joins, filtering clauses, and sorting. The particular consideration was made to the work with null values and data consistency. In more complicated cases, native SQL queries were employed to have more control over retrieving data and enhance performance.

5.4 Application of Repository Layer

Spring Data JPA was used to implement the repository layer making it easier to work with the database. I developed the interfaces with repositories and methods of data access. Annotations were used to implement custom queries in which it was possible to run native SQL queries via the application. Queries also took parameters dynamically allowing to filter data flexibly. This layer provided effective communication between the database and the application.

5.5 Use of Projection Technique

To optimize data retrieval, projection techniques were used extensively. Only required fields were chosen and projected to interfaces of projections instead of retrieving whole database entities. This strategy minimized memory consumption and enhanced performance of applications. Consideration was given to the fact that the column names in the SQL query have to be the exact names of the method

names in the projection interfaces. Otherwise there may be a misalignment of data that could result in mis-map of the data. This method was especially effective in dealing with massive data volumes.

5.6 Service Layer Implementation

The service layer was important to process the data and to introduce business logic. It served as a communication between controller layer and repository layer. Data processing, business rules, and conditional logic were dealt with in this layer. Scenarios where some fields may be null, as an example, need further logic to identify the proper value that should be displayed. The service layer was also to guarantee the data that was sent back to the controller was well structured and uniform.

5.7 API Development

RESTful APIs were created to enable an interaction between the frontend and backend systems. The controller layer received incoming requests, sent them to the service layer to be processed and returned structured responses to the client. The correct handling of the request and formatting of the response were observed to facilitate the efficient data transfer. This process assisted in the creation of a trusted interface between the interface and the back end system.

5.8 Testing and Debugging

The development process involved testing and debugging. Once the features had been implemented, I checked the output accuracy by testing various situations. The process of debugging included finding and fixing bugs like wrong query output, data mapping error, and failure to display data. The development environment presented tools to trace errors and analyze system behavior. This cycle allowed the application to work properly and within the necessary specifications.

5.9 Performance Optimization

One of the crucial points of the methodology was performance optimization. Attempts were being made to enhance system efficiency by optimizing SQL queries, eliminating unnecessary data retrieval, and minimizing database calls. Projection was also very useful to improve performance since it reduced the volume of data being transferred out of the database. These optimizations led to reduced response times and improved the general performance of the system.

5.10 Handling Requirement Changes

Client requirements changes, and it would be necessary to modify existing features and add new features. This included revising queries, modifying of business logic, and making sure that new developments would not have an impact on the previous system behavior. These changes had to be managed with flexibility, critical analysis and adequate testing. This experience helped gain a better idea of what software development is like in the real-world, where flexibility is a necessity.

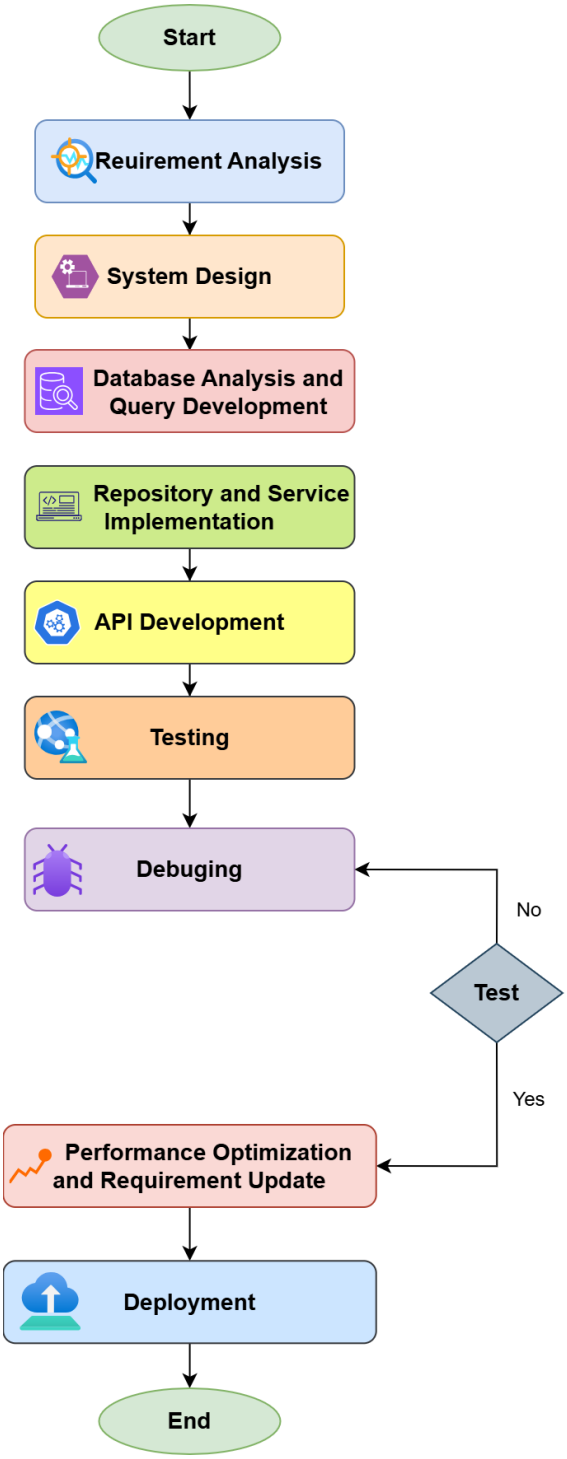

Figure 5.1: Development Workflow

Chapter 6

Analysis and Findings

6.1 Results/Findings

This internship experience provided a lot of technical and professional growth, as well as valuable contributions to the given project. Different tasks and duties completed provided some valuable results and findings. The main points of results and findings are as follows:

- **Improved System Performance**
  The system performance was improved by using projection techniques and optimization of the native SQL queries. It also made the data retrieval faster because only the necessary fields were retrieved and not the whole entities.

- **Efficient Data Handling**
  Spring Data JPA with projection and parameterized queries provided an efficient processing of large datasets without the need to process extra data and less time spent on responses.

- **Accurate Data Mapping**
  Problems with misaligned data display or data that was not displayed were fixed by making sure that query aliases were aligned correctly with projection techniques. This resulted in a better representation of data in the system.

- **Imnplemented Business Logic**
  The service layer methods had been successfully used to manage the business rules, conditional logic, such as the control of the null values, and presentation of the right data, like updated information must be shown.

- **Flexibility to Requirement Changes**
  Dealing with evolving client needs and the need to add new functionality taught the value of writing flexible and maintainable code. It also improved the ability to quickly adapt to new system needs.

- **Improved Knowledge of Layered Architecture**
  The real-world application of controller, service, and repository layers helped to have a clear insight into the structure and maintenance of enterprise applications.

- **Improved Debugging and Problem-Solving Skills**
  Addressing real-time problems like query errors, data mismatch, and log gaps greatly contributed to the development of debugging skills and analysis.

- **Practical Enterprise Development Exposure**
  Experienced in practical work with real-life projects, tools, and technologies in industry-level software development.

6.2 Learning and Development

The internship experience was also a good opportunity for both professional and technical development. It allowed me to build on what I already knew and learn new things by gaining real-world experience in a software development environment.The key areas of learning and development are outlined below:

- **Backend Development Skills**
  Obtained practical skills in backend development with Java and the Spring Boot framework, which includes the creation of application logic, request processing, and server-side operation.

- **SQL and Database Management**
  Enhanced my skills in handling relational databases, especially Oracle, through the writing of effective SQL queries, joins, filtering of data, and query optimization.

- **Knowledge of Spring Data JPA**
  Studied how to interact with a database with Spring Data JPA, such as how to

create a repository, how to implement a native query, how to bind parameters, and how to project.

- **Layered Architecture is applied**
  Gained a clear picture of the Controller-Service-Repository architecture, and this assisted in structuring the code in a modular and maintainable manner.

- **Problem-Solving and Debugging Skills**
  Improved my skills in detecting, diagnosing, and fixing real-time problems like data inconsistencies, query errors, and logic errors by performing systematic debugging.

- **Handling Real-World Challenges**
  Trained to adjust to evolving needs, add new functionality, and adjust existing code without compromising system stability and performance.

- **Quality and Best Practices of the Code**
  Learned the clean code standards, how to name items, and write maintainable and reusable code.

- **Communication and Teamwork Skills**
  Enhanced communication through collaboration with team members and supervisors, discussing issues, and providing feedback.

- **Professional Development**
  Learned to be responsible, manage time, and be flexible in the workplace.

6.3 Tools and Technology

6.3.1 Tools

- **JAVA:** The major programming language used to build the backend of the application is Java (JDK-17). It offers features like encapsulation, inheritance, and polymorphism that are object-oriented and assist in creating modular and maintainable systems. Java is used in this project to:
  - Implement business logic.
  - Define data models and entity classes.

- Create service and repository layers.
- Process data and format responses.

Figure 6.1: JDK 17

- **Spring Boot - Backend Framework:** Spring Boot - Backend Framework. Spring Boot is a robust framework that helps to simplify the development of Java-based enterprise applications. It lowers the configuration complexity and enables quick application development. Key features:
  - Auto-configuration: Automatically configures the application components according to their dependencies.
  - Embedded Server: It uses embedded servers such as Tomcat, and this type of server does not need external deployment.
  - Spring Data JPA Integration: Uncomplicates the database functions.
  - Dependency Injection: Takes care of object creation and lifecycle.

  Spring Boot is used in this project:
  - Build RESTful APIs.
  - Operate application layers (Controller, Service, Repository).
  - Integrate with the database using JPA.
  - Efficiently process requests and responses.

Figure 6.2: Spring Boot

- **IntelliJ IDEA -Integrated Development Environment (IDE):** Code is written, tested, and debugged through an IDE. IntelliJ IDEA is a complete Java development tool. Key features:
  - Code Completion and Suggestions: Improves productivity and reduces errors.
  - Debugging Tools: Assists in finding and correcting problems.
  - Project Management: arranges files and dependencies.
  - Database Integration: Allows direct interaction with databases.
  - Version Control Integration: Git support for source code management.

  The IDE is applied in this project in:
  - Java code management and writing.
  - Operating and testing the Spring Boot app.
  - Troubleshooting problems like a mismatch of queries and data mapping.

Figure 6.3: Intellij Idea

- **Oracle -Database management system:** Oracle -Database management system: Application data is stored, managed, and accessed via a Database Management System (DBMS). In this project, Oracle Database is used. JDBC

connects the database to the application with the help of Spring Data JPA. Key roles:

  - Store organized information like policy details, application status, and activity logs.
  - Run SQL statements to access and manipulate data.
  - Maintain data integrity and safety.

Figure 6.4: Oracle

- **Maven -Dependency Management and Build Tool:** Maven is a build automation tool to manage dependencies of the project and build processes.
  - Has an XML configuration file (pom.xml).
  - Handles dependencies through downloading the necessary libraries in Spring Boot Initializer.
  - Offers a uniform project format.
  - Handles build lifecycle (compile, test, package, install).

Figure 6.5: Maven

- **Postman:** Postman was used as an API testing tool to test and validate backend services. It was utilized to send HTTP requests such as GET, POST, PUT, and DELETE to the server and analyze the responses returned by the application. This helped in verifying the correctness of API endpoints, checking response status codes, and ensuring proper data exchange between the client and server. It also assisted in debugging and identifying issues in the backend logic.

Figure 6.6: Postman

6.3.2 Libraries and Frameworks

- **Spring Data JPA:** Spring Data JPA is a part of the Spring framework that enhances easy interaction with the database by minimizing the amount of boilerplate code. That is needed to perform data access layers. It offers a repository-based model to carry out CRUD (Create, Read, Update, Delete) operations without writing long SQL queries. It greatly enhances the efficiency of development and the clean separation of business logic and data access layers. In this project, Spring Data JPA is used to:
  - Specify database interface functions.
  - Run JPQL and native SQL queries with annotations.
  - Generate automatic queries by the name of methods.
  - Generate automatic queries by the name of methods.
  - Bring in harmony with Hibernate as an ORM.
- **Hibernate (JPA Implementation):** Hibernate is an Object-Relational Mapping (ORM) framework, which realizes the JPA (Java Persistence API) specification. It provides a mapping between Java objects and tables in the relational database. Mapping Java classes to database tables, Java data types to SQL data types and vice versa, entity lifecycle (creation, updating, deletion), caching, and slow loading are some of the important functions of Hibernate, used to provide performance optimization. Hibernate is used in this project to:

  - Map database tables with entity classes.
  - Carry out database actions using JPA repositories.
  - Maximize data access and storage.

- **Lombok:** Hibernate is an Object-Lombok is a Java library to minimize boilerplate code, through the automatic generation of commonly used methods at compile time such as getters, setters, constructors, and toString. Annotations that are commonly used include: @Data, @Getter, and @Setter. In this project Lombok is used to:
  - Minimize repetitive code in entity and DTO classes.
  - Enhance readability and maintainability of code.
  - Reduce development time.

6.3.3 Important Functions and Annotation:

This project implemented several significant Spring Framework and Spring Data JPA annotations and features to perform database operations, manage application layers, and ensure efficient data handling. Such annotations are essential in minimising boilerplate code and enhancing the application structure.

- **@Query:** Custom database queries may be defined using the @Query annotation specified in the repository interface. It supports JPQL (Java Persistence Query Language) as well as native SQL queries. In this project, querying with (nativeQuery = true) is mostly employed to run raw SQL queries to retrieve complex data. Key purposes:
  - Perform complicated queries, which are not manageable using method naming conventions.
  - Access particular columns rather than entire entities.
  - Enhance the performance by retrieving only the necessary data.
  - Example: @Query(value = ”SELECT S.POLICY-NO AS policyNo, S.STATUS AS status FROM BA-SUR-APPLICATION S WHERE S.STATUS = :status”, nativeQuery = true) List¡SurrenderAppProjection¿ getSurrenderAppListByStatus(@Param(”status”) String status);

- **@Param:** Parameters in a query are named and bound to method parameters by the annotation, which is the @Param annotation. It keeps dynamic values securely sent to SQL queries.Key purposes:
  - Enhance the readability of queries with named parameters.
  - Avoid SQL injection vulnerabilities.
  - Allow data to be dynamically filtered.
  - Example: @Param(”status”) String status
- **@Repository:** The @Repository annotation is used to define the data access layer (DAO). It denotes the class or interface that interacts with the database. Key purposes:
  - Allows Spring Data JPA repository functionality.
  - Offers automatic exception translation.
  - Simplifies database operations.
- **@Service:** The service layer is defined using the annotation of the service, which is the business logic.Key purposes:
  - Divide business logic and data access logic.
  - Enhances the code structure and maintainability.
  - Acts as an intermediary between the controller and the repository.
- **@Entity:** The Entity Java class is marked with the help of the @Entity annotation. A row in a database table is associated with each instance of the class.Key purposes:
  - Maps Java class to database table.
  - Allows ORM with Hibernate.
- **@Table:** To identify the table name in the database, the annotation of the table is denoted by the symbol @Table.
  - Example: @Table(name = ”BA-SUR-APPLICATION”).

- **@Id:** The primary key of an entity is defined with the help of the annotation of the type of an entity called the primary key, which is denoted by the term of the annotation as @ID. Key purposes:
  - A unique identifier of each record in the table.
  - Required to map entities.
- **@Modifying:** The @Modifying annotation is applicable with the @Query in performing update or delete operations.Key purposes:
  - Allows non-select queries to be executed.
  - DELETE, UPDATE or INSERT operation required.
- **@Transactional:** The database transactions are managed using the annotation @Transactional. Key purposes:
  - Ensures data consistency.
  - Auto-commits or auto-rolls back transactions.
  - Needed to alter queries.

6.3.4 Projection:

Projection is a method in spring data JPA which is utilized to obtain particular columns in the database rather than the whole entity. This method enhances performance, minimizes the use of memory, and improves the performance of the applications since it only transfers the necessary data. Projection is an important feature in this project and is implemented with native SQL queries to streamline data retrieval to the surrender application module.The primary reasons why projection will be used in this project are:

- Selecting minimum fields in the database.
- Enhance query response by decreasing the data load.
- Not unnecessarily mapping full entity objects.
- Improving API call response time.

Types of Projection that will be used in this project:

- **Interface-Based Projection:** The results of the query are projected into an interface using interface-based projection. Spring Data JPA will automatically provide the interface and map the result of the query to the existing getters.Key Points:
  - No longer necessary to create object instances manually.
  - The name of methods should be the same as the column names in the query.
  - Easy and compact to use in read-only applications.
  - Example:
    public interface SurrenderAppLProjection (
    String getPolicyNo();
    String getStatus();
    String getApplicantName();)
- **DTO (Data Transfer Object) Projection:** When greater control of data transformation is needed, DTO-based projection is employed. It entails the development of a class that has a constructor that corresponds to the chosen fields in the query. Key Points:
  - Apposite to the multifaceted transformation of data.
  - Gives the ability to be flexible in processing business logic.
  - Needs explicit mapping of constructors.
  - Example: public class SurrenderDTO (
    private String policyNo;
    private String status;
    public SurrenderDTO(String policyNo, String status) (
    this.policyNo = policyNo;
    this.status = status;)
)

6.4 User Guideline

This section will give guidelines on how to utilize the developed software system effectively. The guideline will assist the user to learn the system functionalities, access the application and interact with the features of the application. The application is primarily a backend solution that exposes services via APIs which can be tested with API testing tools or incorporated front end system.

- Use a web interface or API testing software such as Postman to access the system.
- Always verify before getting access to secured features by verifying with valid credentials.
- Appropriate API endpoints, and appropriate HTTP methods (GET, POST, PUT, DELETE).
- Enter correct input information and adhere to the necessary formats to prevent mistakes.
- Make requests and responses in JSON format.
- Status messages and codes of checks to verify successful operations.
- Manage error by going through messages and fixing input or parameters.
- Follow best practices, like testing in a safe setting, and data security.

6.5 Impact of the Internship

The development of my technical competencies was one of the greatest effects of this internship. My knowledge of backend development with Java and the Spring Boot framework was strong. Also, I had practical experience with database management, optimization of SQL queries, and the application of Spring Data JPA. This real-world experience enhanced my skills to design and deploy a scalable and effective application setup.

My problem-solving and analytical skills were also enhanced by the internship. The real-world problems that had to be solved, like data mapping errors,

query mismatch, and system optimization, required critical thinking and systematic debugging. Consequently, I gained greater confidence in finding out issues and applying solutions to those issues.

The other significant effect was the formation of a professional attitude. I was taught how to operate in a well-organized development process, follow the coding standards, meet the deadlines, and adjust to shifting needs. My experience in exposure to actual client-based projects helped me realize the relevance of flexibility, communication, and teamwork in the workplace.

Also, the experience helped me gain knowledge about the enterprise-level application architecture, such as how to use the controller, service, and repository layers correctly. This experience will be of great use in future developmental work and in one's career.

All in all, the internship was essential in equipping me with the software industry, enhancing my technical skills and confidence, as well as giving me a glimpse into the practice of development in the real world. It has provided me with a solid base for my future career in Information Technology.

Chapter 7

Conclusions and Recommendations

7.1 Conclusion

The internship program has been a transformative and rewarding experience that has managed to help bridge the gap between the academic knowledge and the practical. It also gave me the experience of working within a real-world professional environment, during which I could implement theoretical concepts to find solutions to practical issues and work on an enterprise-level application.

During the internship, I already learned practical experience in development in the backend with Java and the Spring Boot framework. A good idea of what database management and optimization of SQL queries is and how to use Spring Data JPA. Application of native queries, projection methods, and layered architecture further contributed to improvement of my technical knowledge and expanded my knowledge on efficient system design.

Besides technical skills, the internship was a great experience in my professional growth. I got to know how to work in a well organized team setting, how to handle changing client needs, time management and follow industry best practices. The difficulties that I had to face during the internship allowed me to work on my problem-solving skills, critical thinking, and debugging skills.

Moreover, this experience provided valuable insights into real-world software development processes, including collaboration, communication, and maintaining code quality. It has boosted my confidence and equipped me to handle future challenges in the software development sector.

In conclusion, the internship has been a great opportunity to develop my career direction as it has enabled me to learn necessary technical skills and professional

skills. It has provided a strong foundation to my future development and encouraged me to keep learning and advancing the discipline of Information Technology.

7.2 Limitations

Even though the learning experience was very valuable during the internship, it had some drawbacks that limited the ability to explore more and explore the full process of the system. The following are some of the limitations:

- **Restricted Access to Complete System:** Some core modules and sensitive data were limited to access because of security and confidentiality concerns. This restricted a full comprehension of the system architecture.
- **Time Constraints:** The internship was not that long, which was a disadvantage because it was not possible to explore more complex concepts and deepen into the knowledge of all technologies used in the project.
- **Little experience with Frontend Development:** This internship was more of a backend development and I was exposed to little frontend technology and user interface design.
- **Dependency on Existing Codebase:** There was a time constraint, in that, I needed to work on an already existing large-scale project and therefore needed time to learn about the codebase and system business logic.
- **Changing Requirements:** The need to often rework due to changing client requirements impacted time management and reduced the opportunity to investigate further functionality.
- **Little Involvement in Deployment Process:** Little was exposed to deployment, DevOps practices, and managing systems at the production level.

7.3 Recommendations

- Enhance technical documentation to have a better understanding of the system.
- Give organized training or orientation to interns.
- Adopt a routine code inspection to guarantee code quality.
- Deal with requirement changes more formally.
- Enhance full development lifecycle including deployment.
- Monitor performance using performance monitoring tools.
- Foster sharing of knowledge and teamwork.

Chapter 8

Acknowledgments

I would sincerely like to say that both of my academic and idustry supervisors and the team members at the organization has been supportive. They guiding me, and encouraging to be my constant support and guide during my internship. Their insights and constructive feedback were very valuable to my learning and development as a professional. I also owe my institution the privilege of being able to have a hands-on experience in a real-world setting.

I would particularly like to acknowledge my parents who were always supporting, motivating, and encouraging me, which allowed me to successfully complete this internship. Their faith in me has been my strength. Without all this help and collaboration, this would not have been accomplished.

Appendix

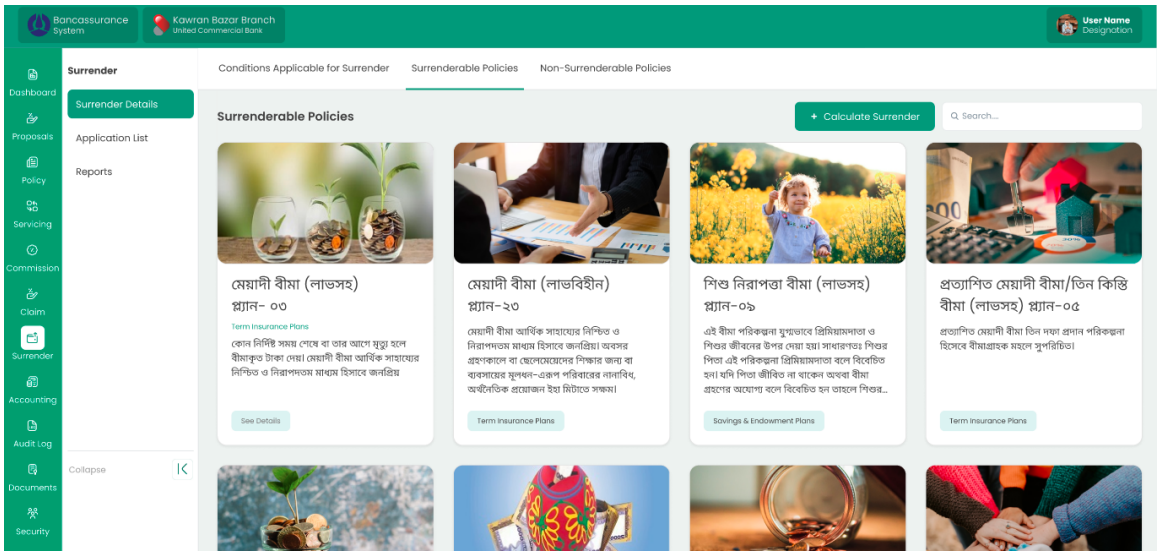

Figure 8.1: Surrender Details Page

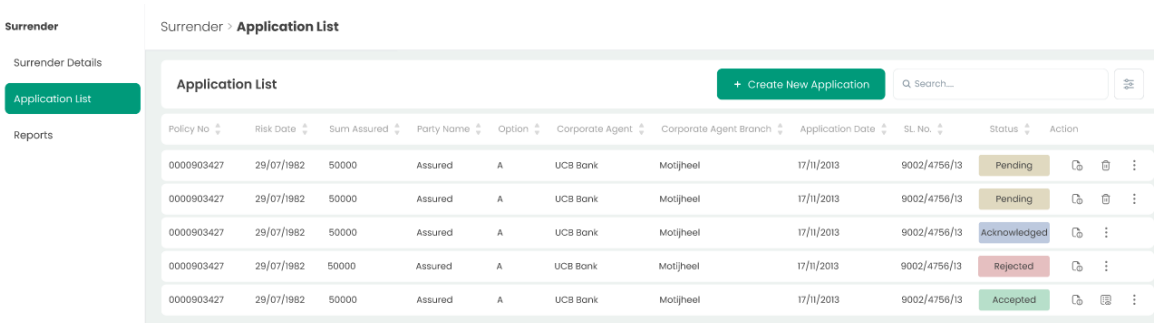

Figure 8.2: Surrender Application Page

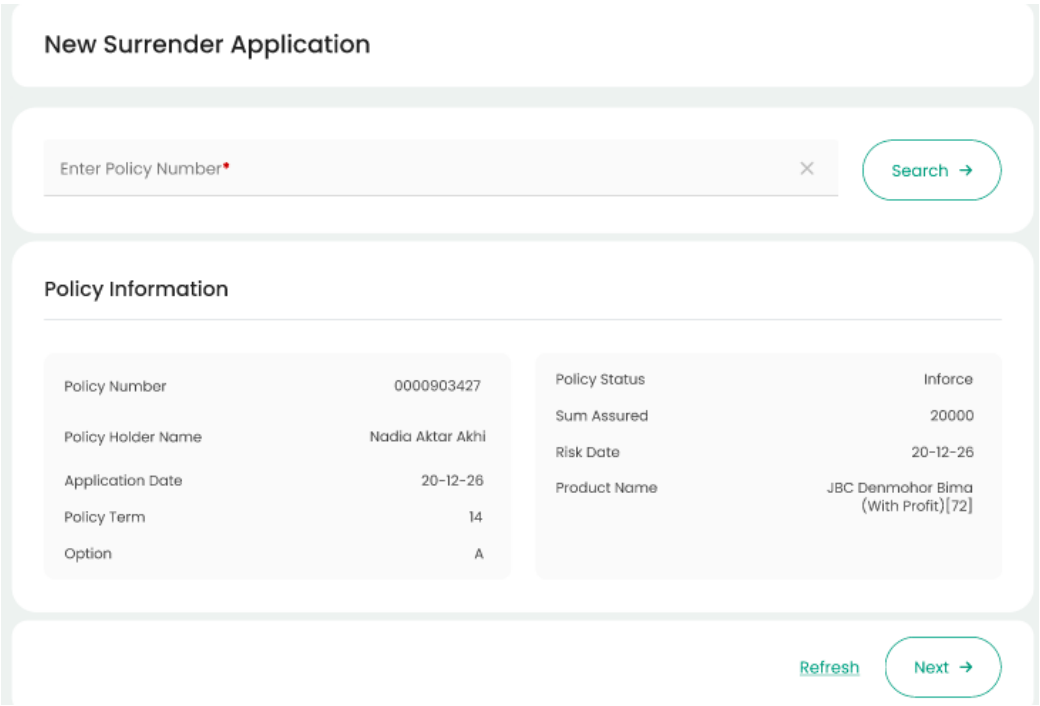

Figure 8.3: Policy Information

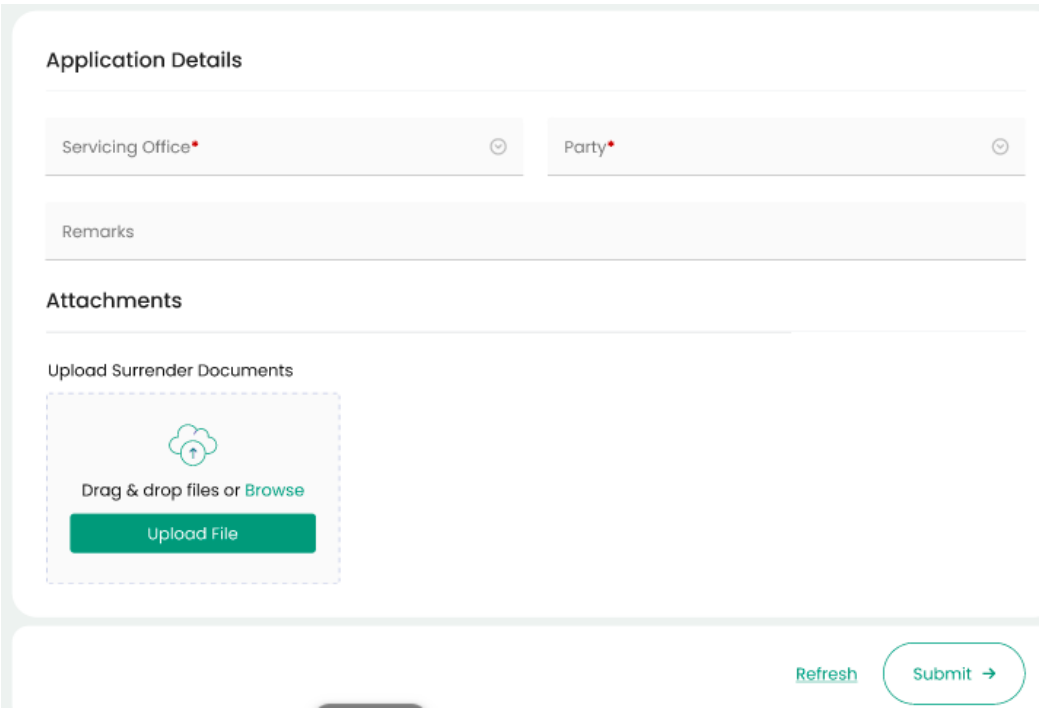

Figure 8.4: Submit Application

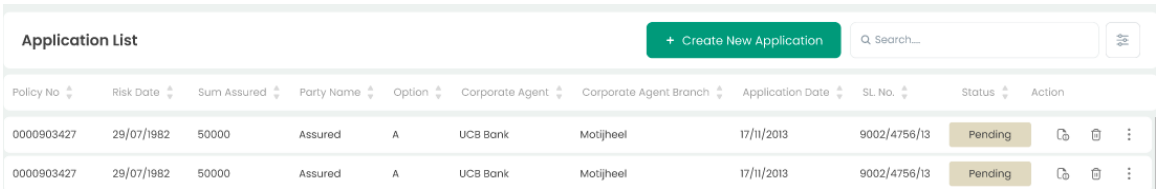

Figure 8.5: Application List

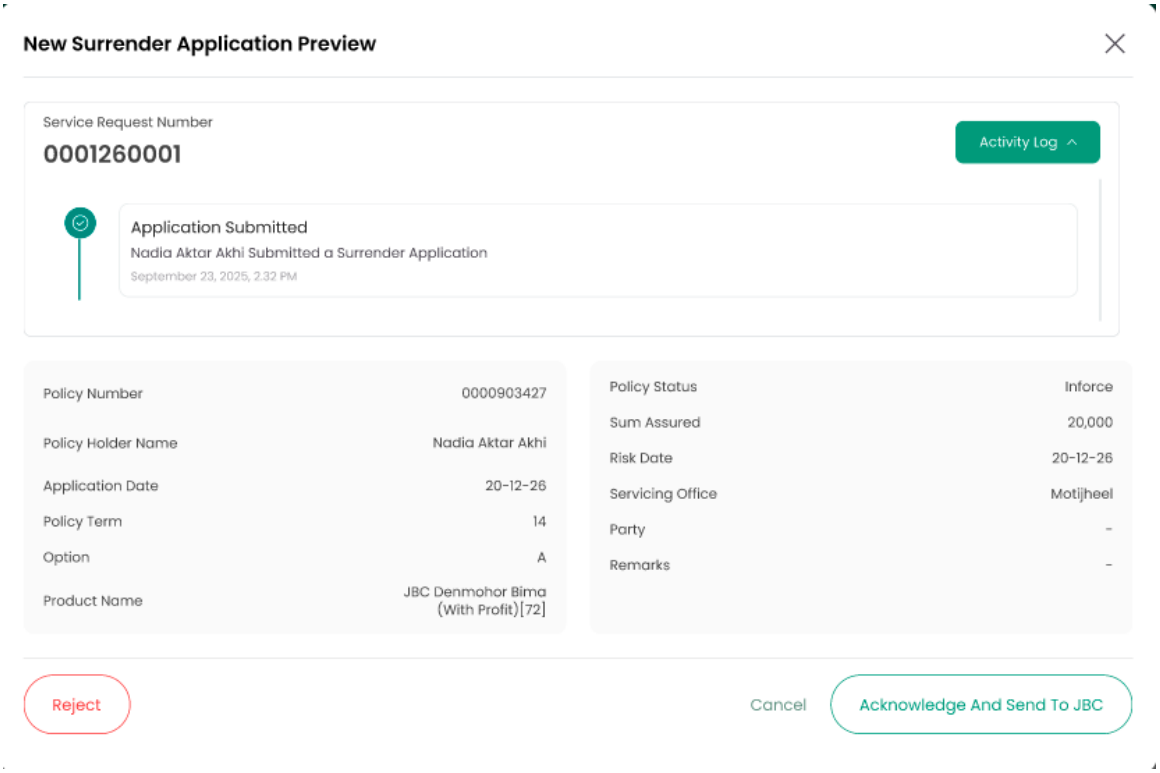

Figure 8.6: Acknowledge and Reject Option

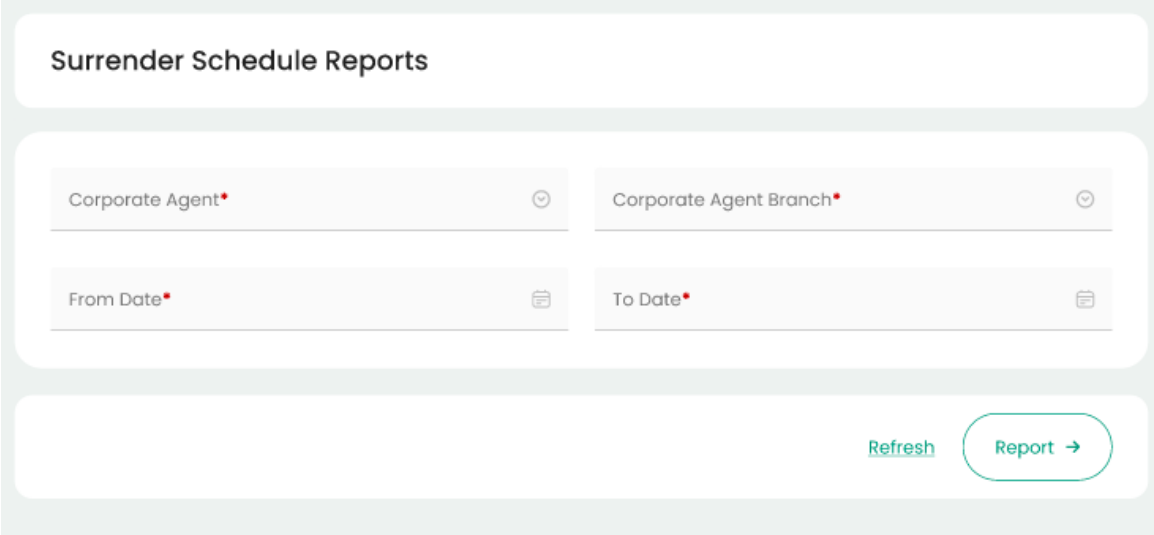

Figure 8.7: Surrender Schedule Report

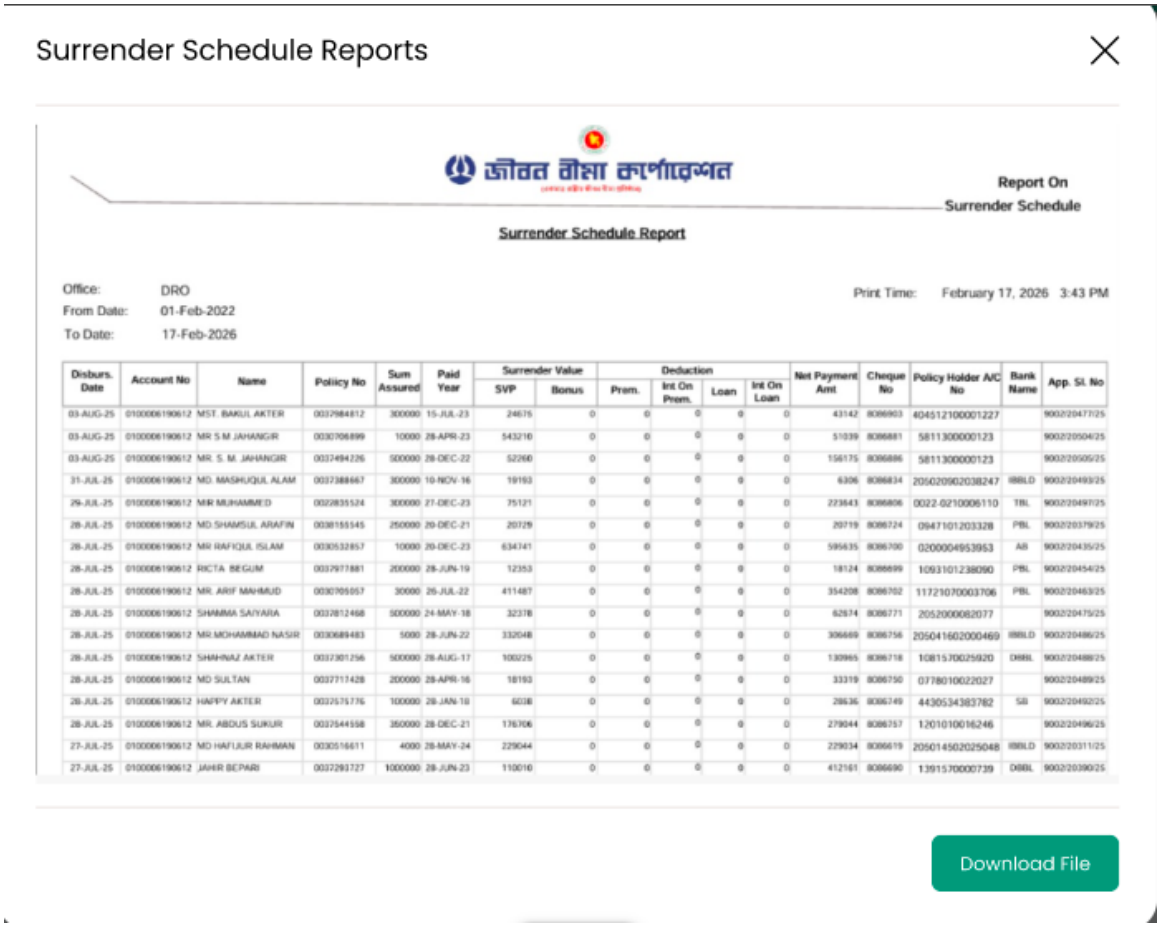

Figure 8.8: Check and Download

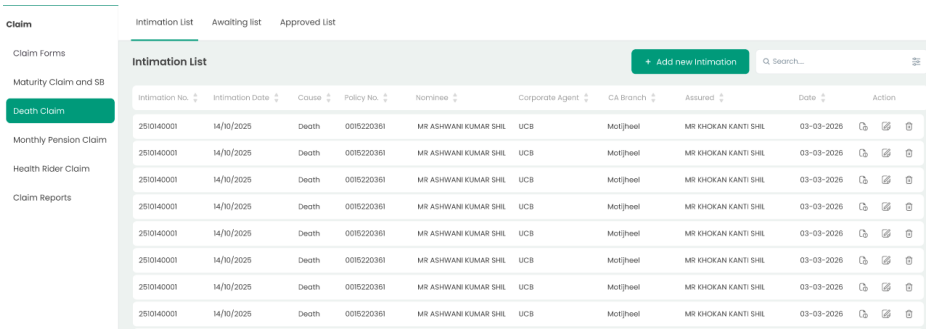

Figure 8.9: Claim Module

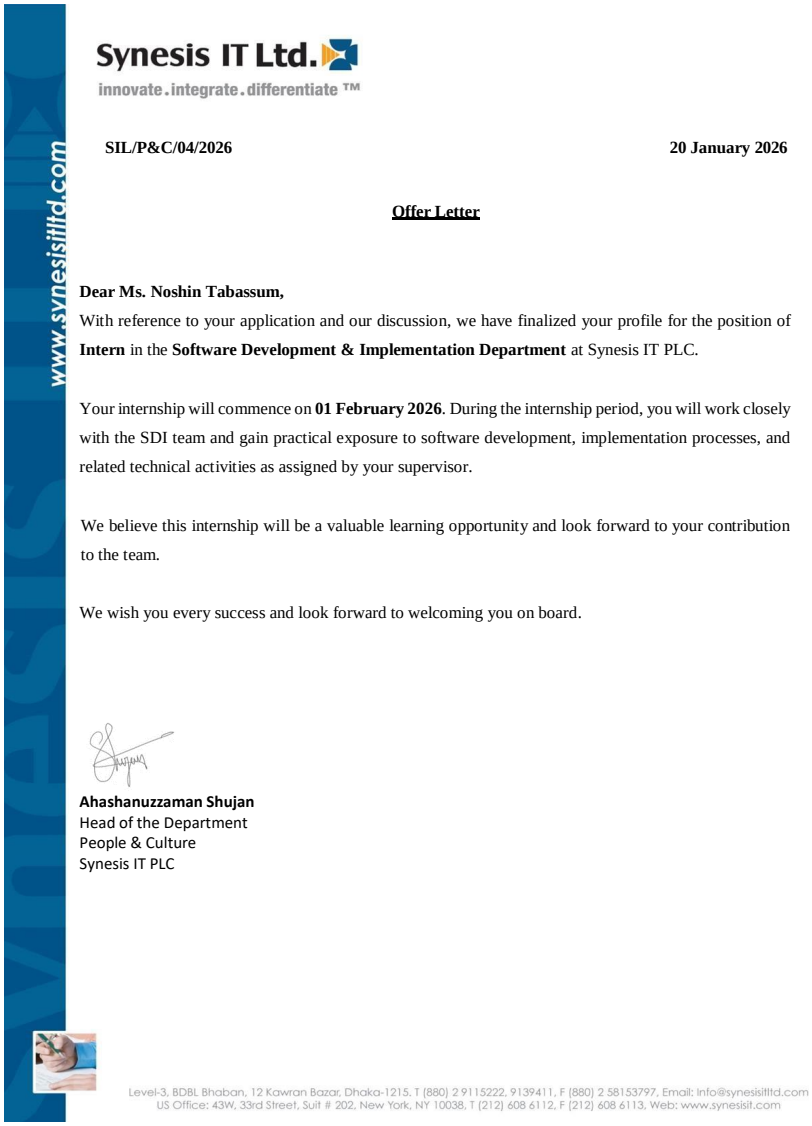

Figure 8.10: Offer Letter

References

Synesis IT PLC. (2006). *Synesis it plc*. https://synesisitltd.com/.