**Acknowledgement**

I wish to express my deep sense of gratitude to **Dr.V.Abhai Kumar,** Principal of Thiagarajar College of Engineering for his support and encouragement throughout this project work.

I wish to express my sincere thanks to **Dr.S.Mercy Shalinie**, Head of the Department of Computer Science and Engineering for her support and ardent guidance.

I owe my special thanks and gratitude to **Mr.K.Narasima Mallikarjunan M.E.,** Department of Computer Science and Engineering for his guidance and support throughout our project.

I am also indebted to all the teaching and non-teaching staff members of our college for helping us directly or indirectly by all means throughout the course of our study and project work.

I extremely thank my parents, family members and friends for their moral support and encouragement for my project.

**Abstract**

The purpose of this project is to build a system that will allow clients to initiate their pledging process using SPA. Currently Clients are handling their own collateral funding and pledging process using various tools and platform. They use the one of the system to view securities that are available for pledging and releasing and use another tool to complete their pledging process by filling their shells (method used to classify the quality of securities).

The Client is now looking to be a fully disclosed client of Clearing house and will need to be able to perform their pledging process using the Clearing house systems. Client will be using the Front end screens to only see securities for pledging and releasing. Entitlements will be created and be available to client. No clients will be having direct control over pledge and release process. There will be an intermediate for every client namely broker dealer, who takes care of all the process.

The clearing house system mainly focuses on complete settlement policies between the two parties in a transaction. Some transaction may get cleared in two days, some may take more than two days which depends on their pledging policies, usually it is termed as t+2, t+3 days transaction. It means within the specified days after the transaction has done, the complete clearance of the securities and the proof of transaction will be provided to both the parties and no pending works will be there.

The data for pledging and releasing will be maintained for 10 years with the current open items. The Pledging Screen will be available 24/7.  Action can only be taken on US business days 6 AM – 6 PM. During bank holidays, the screen will display the account holdings but no action can be taken from the screens.

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Chapter No** | **Title** | **Page No** |
|  |  |  |
|  | Abstract | **ii** |
|  | **List of Tables** | **viii** |
|  | **List of Figures** | **ix** |
|  | **List of Abbreviations** | **xi** |
|  |  |  |
|  |  |  |
| **1.** | **Introduction** | **1** |
|  |  |  |
| 1.1 | Introduction to Technologies | 2 |
|  | 1.1.1 Angular 2 | 2 |
|  | 1.1.2 Type Script  1.1.3 Node.js  1.1.4 JavaScript  1.1.5 Web Services    1.1.6 XML  1.1.7 JSON  1.1.8 HTTP  1.1.9 REST | 3  4  5  6  7  8  8  9 |
| 1.2 | Introduction to Financial Services | 10 |
|  | 1.2.1 Clearing House  1.2.2 Pledge  1.2.3. Release | 10  12  12 |
|  |  |  |
| **2.** | **Problem Definition and Background** | **14** |
|  |  |  |
| 2.1 | Existing Approach | 14 |
| 2.2 | Problem Statement | 16 |
|  |  |  |
| **3.** | **Requirements analysis** | **18** |
|  |  |  |
| 3.1 | Hardware Support | 18 |
|  | 3.1.1 Client Side  3.1.2 Server Side | 18  19 |
| 3.2 | Software Support | 21 |
|  | 3.2.1 Windows Operating System  3.2.2 Java8  3.2.3 Visual Code  3.2.4 Eclipse IDE  3.2.5 Bootstrap Framework | 21  21  22  23  23 |
|  |  |  |
| **4.** | **Proposed Approach** | **26** |
|  |  |  |
| 4.1 | Overview of proposed Approach | 26 |
| 4.2 | Block Diagram | 27 |
|  | 4.2.1 Control Flow  4.2.2 Pledge and Release | 27  29 |
|  |  |  |
| **5.** | **Interaction Scenario** | **31** |
|  |  |  |
| 5.1 | Class Diagram | 31 |
| 5.2 | Use Case Diagram | 33 |
| 5.3 | Activity Diagram | 35 |
| 5.4 | Sequence Diagram | 37 |
| 5.5 | Collaboration Diagram | 39 |
| **6.** | **Implementation** | **42** |
| 6.1 | npm and node.js installation | 42 |
|  | 6.1.1 Code Snippet | 44 |
| 6.2 | Interfacing Client Front End and RESTful Services Using Spring MVC framework | 46 |
|  | 6.2.1 Spring MVC  6.2.2 Interfacing RESTful Services and  Backend Mainframe Server Using  Message Queuing | 46  47 |
| 6.3 | Setting up the Initial Software and configuring all the configuration files | 52 |
| 6.4 | Performing Pledging Process | 52 |
| 6.5 | Performing Release Process | 54 |
| **7.** | **Results** | **56** |
| 7.1 | Pledge | 56 |
| 7.2 | Release | 57 |
| 7.3 | Error | 58 |
|  |  |  |
| **8.** | **Conclusion** | **60** |

**List of Tables**

|  |  |  |
| --- | --- | --- |
| 4.1 | Hardware Specification | 32 |
| 4.2 | Mainframe Specification | 34 |
|  |  |  |
|  |  |  |

**List of Figures**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | |  |  | |
| 2.1 | Block diagram of the existing approach | | | 15 |
| 4.1 | Block diagram showing flow of control from front end to back end mainframe | | | 27 |
| 4.2 | Flowchart of Pledging and Release Screen | | | 29 |
| 5.1 | Class Diagram for Pledging and Releasing Process | | | 31 |
| 5.2 | Use case Diagram for Pledging and Releasing | | | 33 |
| 5.3 | Activity Diagram for overall pledging and releasing process | | | 35 |
| 5.4 | Sequence Diagram for Pledging and Release Process | | | 37 |
| 5.5 | Collaboration Diagram for Pledging and Releasing Process | | | 39 |
| 6.1 | Spring Framework | | | 46 |
| 6.2 | Dependency injection | | | 50 |
| 6.3 | Data format | | | 53 |
|  |  | | |  |
| 7.1 | Output Screen of Pledging Process | | | 56 |
| 7.2 | Output Screen of Releasing Process | | | 57 |
| 7.3 | Output Screen of Error Message | | | 58 |
|  |  | | |  |
|  |  | | |  |
|  |  | | |  |
|  |  | | |  |
|  |  | | |  |
|  |  | | |  |
|  |  | | |  |
|  |  | | |  |
|  |  | | |  |

**List of Abbreviations**

|  |  |
| --- | --- |
| CPU | Central Processing Unit |
| IO | Input Output |
| IDE | Integrated Development Environment |
| RAM | Random Access Memory |
| HTML | Hypertext Markup Language |
| CSS | Cascading Style Sheet |
| API | Application Programming Interface |
| HTTP | Hypertext Transfer Protocol |
| XML | Extensible Markup Language |
| JSON | JavaScript Object Notation |
| REST | Representational state transfer |
| SOAP | Simple Object Access Protocol |
| FTP | File Transfer Protocol |
| AJAX | Asynchronous JavaScript and XML |
| FDIC | Federal Deposit Insurance Corporation |
| IBD | Intermediate Broker Dealer |
| HCL | Hardware Compatibility List |
| CUSIP | Committee on Uniform Securities Identification Procedures |