Portfolio: What is SDLC?

Examples of UML Diagrams and Documentation

CNIT38000 Fall 2024: West Lafayette

Jack Foreman, Noah Zhou, Zhitian Wang

Submitted To: Professor Casner

Date Submitted: 11/26/2024

Date Due: 11/26/2024

Table of Contents

CNIT 38000 Diagram Portfolio

SYSTEMS DEVELOPMENT LIFE CYCLE

During the planning phase, our team defined the scope of application and service objectives of the project, clarified the main functions and user roles that the system should provide and complete, such as potential customers, customers, check payees, and bank managers, and distinguished the roles that each function should correspond to. At the same time, the project planning included and identified the stakeholders of the project, and conducted a feasibility analysis to ensure that the project can achieve its goals within the specified time.

During the analysis phase, our team collected detailed user needs and system requirements. For example, we analyzed how potential customers submit account applications, how bank managers review applications and request credit reports, and how customers check accounts and conduct transactions. This step helps clarify the functional requirements and business rules that the system needs to meet by using use case descriptions to list and analyze various situations.

In the design phase, we transform the requirements into specific system designs. For the banking service management system, we need to design the system's data structure, user interface, account management process, and the way to obtain data from the credit bureau. In this phase, we created the context model diagram (CMD) and use case diagram (UCD) to ensure the visualization and clarity of the system design, and also prepared the subsequent design direction.

In the development stage, our team wrote the project object code and implemented the system program according to the specifications formulated in the design stage. We have successfully completed the core functional modules for reviewing and responding to credit report requests from account submission to ensure that the system meets the defined specifications of the analysis and design phase. To create a sound and reliable banking service management system.

During the test phase, our team performed unit tests, integration tests, and system tests to verify that all system functions worked correctly and met the expected requirements standards. The tests included checking the accuracy of account application processing and the correct response to credit report requests and ensuring that the system functioned as expected. This part of the work is critical to identifying and resolving any issues or errors to improve system reliability and stability before deployment.

In the deployment phase, our team provides system services and teaching to relevant staff to ensure adaptability when the system is about to be launched online. This part of the deployment will involve installing the system on a physical server and ensuring that all components are connected and running efficiently. We created deployment diagrams that illustrate the distribution of the system on different hardware to ensure that the system can be deployed efficiently and maintain optimal performance and scalability.

During the maintenance phase, our team continuously updates and fixes the system based on user feedback and operational requirements. This move is designed to ensure that the system can continue to meet customer needs in the long term. Maintenance also includes applying security updates and feature enhancements to adapt to changing business needs and regulatory requirements to ensure stable system operation.

What is the UML Context Model Diagram?

A context model diagram provides a high-level overview of a system, helping people quickly understand its scope and boundaries without delving into detailed internal processes. It illustrates how the system interacts with external entities and the environment, which can include users, other systems, and processes. This diagram is crucial for identifying potential system requirements, risks from external interactions, and defining the system's scope and boundaries. The diagram consists of four main components. The first is the system, typically represented by a circle at the center. Surrounding the system are external entities, depicted as rectangles. The final component is data flows, shown as arrows that can be bidirectional or unidirectional, indicating the direction of data exchange between the system and external entities.

What is the Requirements Documentation?

The requirements document is a crucial component of the system development life cycle (SDLC), used to detail both functional and non-functional requirements of the system. Its primary purpose is to clarify the project's objectives and user needs, ensuring all stakeholders share the same expectations. Functional requirements specify the system's functions, such as user login and account management, while non-functional requirements cover aspects like performance, security, and scalability. This document serves as the foundation for the design and development phases and acts as a reference for project validation.

What is the Event Analysis Matrix?

The event analysis matrix is a tool used to identify and analyze key events within a system and their corresponding responses. By linking these events to functional requirements, the matrix helps the team gain a clearer understanding of the system's behavioral logic. It typically includes the trigger conditions for each event, the participating system components, and the desired output. This approach effectively supports requirements gathering and functional design.

What is the UML Use Case Diagram?

UML use case diagrams illustrate the functional requirements of a system and how users interact with it. They consist of several core components: actors, which are external entities or other systems interacting with the system; use cases, which describe the functions the system performs; and relationships, which depict the connections between use cases and between use cases and actors. The primary purpose of use case diagrams is to define the functional boundaries of a system and its interactions.

What is a Use Case Narrative?

Use case narratives provide a structured description of the interactions between actors and the system. They typically include elements such as the use case name, actors and stakeholders, trigger conditions and preconditions, normal and alternative processes, and postconditions and business rules. The primary objective of a use case narrative is to ensure that the team has a consistent understanding of the functional requirements, offering a clear blueprint for system development.

What is the UML Activity Diagram?

Activity diagrams are utilized to model business processes or workflows within a system. They include several key elements: activities, which denote specific actions in the process; transitions, which illustrate the connections between activities; and decision points, which indicate conditional judgments within the process. These diagrams visually represent the logical sequence of the process, aiding in the analysis and optimization of system operations.

What is the UML Class Diagram?

Class diagrams are essential for modeling the static structure of a system. They consist of several core elements: classes, which represent the objects within the system; attributes, detailing the data or features of a class; methods, defining the operations or behaviors of a class; and relationships, such as associations, inheritance, and dependencies. These diagrams play a crucial role in outlining the data model and logical structure, making them a vital component of system design.

What is the UML State Machine Diagram?

State machine diagrams are used to depict the various states of a system or object and their transitions. The main components include states, which represent the different conditions of the system; events, which are the conditions that trigger state transitions; and transitions, which are the paths from one state to another. These diagrams help in understanding and analyzing the dynamic behavior of the system, providing support for the design of complex systems.

What is the UML Component Diagram?

Component diagrams illustrate the physical structure of a system, detailing its components and their relationships. The core elements include components, which are the modules or units within the system; interfaces, which are the points of interaction between components; and connectors, which represent dependencies between components. These diagrams offer crucial guidance for system deployment and maintenance, ensuring scalability and reliability.

What is the UML Deployment Diagram?

Deployment diagrams illustrate the hardware and software configuration of a system. The main elements include nodes, which represent hardware or devices; components, which are the software modules deployed on these nodes; and connections, which show how nodes communicate with each other. These diagrams are crucial during the system implementation phase to ensure efficient operation and that performance requirements are met.

What is the UML Sequence Diagram?

UML sequence diagrams are used to model the flow of interactions between objects in a system over time. The main elements include objects, which are the entities participating in the interactions; lifelines, which represent the existence of an object over a period; messages, which are the communications between objects; and activation bars, which indicate the period an object is active during an interaction. Sequence diagrams use a vertical timeline to show the order of interactions, making it easier to understand the sequence of events and the dynamic behavior of the system.

SDLC METHODOLOGY COMPARISIONS

SDLC methodology is compared through traditional and modern methods. Traditional methods, such as the waterfall model, focus on the phased nature of the process, that is, it is more like building a building, and each step must be completed strictly according to plan. The phases, such as requirements analysis, design, development, testing, and maintenance, must be completed one by one, without skipping or backtracking. The biggest advantage of this approach is that the process is clear and each step is supported by detailed documentation, making it ideal for large projects where the requirements are particularly clear and there are few changes. However, its limitations are also obvious, for example, if the requirements change in the middle of the process, the whole project may need to be rebuilt, and the cost and time may increase significantly.

Modern methods, like agile development, are the "opposite" of the waterfall model. Its thinking is people-oriented and emphasizes flexibility and rapid response to change. Agile is more like building blocks, achieving a small goal in stages, and then iterating to optimize. This approach is particularly suitable for projects where the requirements are less clear or change frequently, because each iteration can be adjusted to meet new requirements. Agile development also has the advantage of a high degree of user involvement, such as through frequent demonstrations and feedback, to ensure that the direction of development is always in line with user expectations. However, agile is not a cureall, and it may not be suitable for particularly large projects, or where team members do not communicate and collaborate well.

DIAGRAM COMPARISIONS

The first two diagrams that we will be comparing are the use case narratives and use case diagrams. These two diagrams very closely align with each other when it comes to similarities, since the information taken from the use case diagrams is what used for use case narratives. What this means necessarily is that the basic interactions between the system and actors shown in the diagram are then taken to a more detailed level and explained on a level that accounts for all necessary information and steps necessary to develop the use case for later examination and development. One such example of this from the appendix is in the provided use case diagram where the potential customer actor submits an application. All that is shown in the use case diagram for this is that the customer initiates, and the system receives and performs other actions based on this initiation. While the use case narrative made for this action, shown in the appendix as the first use case narrative, takes this simple interaction in the diagram, breaks it down into steps that are occurring, takes into account all actors, and any alternate outcomes. While the use case narratives do not necessarily need a diagram, a diagram is a good way at displaying as many possible narratives as possible. The main difference between these two is simply that one is a diagram that displays all possible use cases while the other is a specific instance of a use case within that diagram.

The next two diagrams that we compared where the component diagram and the deployment diagram. This is another set of diagrams that are closely related to each other, as the component diagram is something important to consider before creating a deployment diagram. Component diagrams are similar to deployment diagrams in that they both consider how the interface of the system will be developed and what components are necessary for the system to run according to the given requirements and limitations. The difference between the two comes with that component they are look at specifically. Component diagrams tend to look at the physical structure of the system, and well, all of the components that will be interacting with each other within the system. This can be seen in our appendix where our component diagram displays how all different components of the system interact from the UI all the way down to the security component securing the data base structure. The deployment diagram on the other hand, takes these components of the system, and then also takes into account the hardware of the system as well and how these components of the system will be implemented onto a hardware structure. This is also shown in the appendix where the deployment diagram takes the security aspect of the component diagram and applies it to a security server hardware component. The same is also done for the databases, which are shown to be applied to an infrastructure node.

The final two diagrams that we will be comparing are the sequence diagram and activity diagram. This is another set of diagrams that are very similar to each other in many aspects, but also different in others. In the case of what actors and objects are being considered for both diagrams, they both will consider the same diagrams and actions. One such example is in the case of potential customer application again. For both diagram types provided in the appendix, they both list all of the same actors provided, they also provide the same actions occurring between actors. The main difference that these diagrams then have between each other is what events and information is being shown in each diagram. While they both may show the same actors and actions, a activity diagram does a much better job at displaying how control may flow between objects and actors as well as how they interact between each other during these actions, while a sequence diagram does a much better job at showing how messages are sent between these object and actors as these actions are occurring in real time. While both are pretty similar in components, they vary in their importance for specific needs and in how their actions play out with consideration for time.

DIAGRAM X LIMITATIONS

The first diagram that we chose is the requirements analysis as seen in Figures 2A to 2D in the appendix. Requirements are basically the foundation of the project and provided a basis for the rest of the project and associated Unified Modeling Language (UML) diagrams such as the use case diagram, activity diagram, and sequence diagram. Casner (2024) mentions in the lecture about Requirements Analysis and Context Diagrams that “requirements errors are the costliest and most expensive to fix” and that “incorrect or inadequate requirements are the number one cause of project failures.” Without requirements, it would virtually be impossible to know what other diagrams such as use case diagrams, need. For example, as seen in Figure 4 in the appendix, the use case diagram has actors and actions performed. Without knowing the requirements of the system, there is no way of populating the use case diagram because we do not know who the actors are and what the actions they take are.

The second diagram we chose is the activity diagram as seen in Figure 6 in the appendix. Activity diagrams model what must occur in a system, basically illustrating a business process or logic flow between users and the system. Activity diagrams also make it much easier for stakeholders to understand the flow between users and the system. Activity diagrams also clarify use case narratives where steps are often confusing and complicated. For example, Figure 6 depicts the events that occur when a potential customer applies for an account. We can see that the flow starts with the potential customer and ends with either the bank manager or the customer. The logic flow behind where the end occurs can be easily seen within the activity diagram with several events looping around based on decisions or forks, basically saying that if one event occurs, this is what happens and vice versa.

The last diagram that we chose is the class diagram, of which an example can be seen in Figures 7A and 7B. As mentioned previously, class diagrams are used to model the static structure of system and include classes, objects, attributes, and relationships. It provides a clear view of the necessary items to both the project team and the stakeholders. Without this sort of clarity, systems often become poorly designed with classes, objects, and attributes missing. Many systems become very complex and face problems with maintenance and system scaling. For example, in Figure 7A and 7B, we can see that there are various classes such as Bank Teller and Potential Customer and all of the classes have relationships with one another. This translates easily to stakeholders in which they can see that one customer can have multiple accounts but accounts can only have one customer associated with it.

BIBLOGRAPHY

Adobe Communications Team. (2022, March 18). *Waterfall methodology: Project management | Adobe Workfront*. Adobe Workfront. https://business.adobe.com/blog/basics/waterfall

Army Intelligence. (n.d.). *Event Analysis Matrix*. Intelligence Reference and Training Manuals. https://armyintelligence.tpub.com/IT0462/IT04620078.htm

Atlassian. (n.d.-a). *Waterfall methodology for project management*. https://www.atlassian.com/agile/project-management/waterfall-methodology

Atlassian. (n.d.-b). *What is agile?* https://www.atlassian.com/agile#:~:text=The%20Agile%20methodology%20is%20a,READ%20ON%20BELOW

Casner, M. (2024). Personal Communication.

Decaprio, E. (2006, September 24). Event Analysis. https://www.projectmanagement.com/wikis/233049/Event-Analysis

Dennis, A., Wixom, B. H., & Tegarden, D. P. (2021). *Systems Analysis & Design: An object-oriented approach with UML*. John Wiley & Sons, Inc.

GeeksforGeeks. (2024a, February 21). *Types of Requirements in System Design*. https://www.geeksforgeeks.org/types-of-requirements-in-system-design/

GeeksforGeeks. (2024b, March 21). *Context diagrams*. https://www.geeksforgeeks.org/context-diagrams/

GeeksforGeeks. (2024c, September 23). *What is agile methodology?* https://www.geeksforgeeks.org/what-is-agile-methodology/

Kirvan, P., Lutkevich, B., & Lewis, S. (2024, November 15). *What is the waterfall model? definition and guide: Definition from TechTarget*. TechTarget.

CNIT 38000 Diagram Portfolio

https://www.techtarget.com/searchsoftwarequality/definition/waterfall-model

Laoyan, S. (2024, February 2). *What is agile methodology? (a beginner's guide) [2024] • asana*. Asana. https://asana.com/resources/agile-methodology

Larson, E. (2021, October 21). *A PM's guide to use cases part 3: Use case narratives: Business analysis, PM, and agile training and consulting*. Watermark Learning Blog. https://www.watermarklearning.com/blog/a-pms-guide-to-use-cases-part-3/

Lucidchart. (n.d.-a). *UML Component Diagram Tutorial*. Lucidchart. https://www.lucidchart.com/pages/uml-component-diagram

Lucidchart. (n.d.-b). *UML Deployment Diagram Tutorial*. Lucidchart. https://www.lucidchart.com/pages/uml-deployment-diagram

Lucidchart. (n.d.-c). *UML Sequence Diagram Tutorial*. Lucidchart. https://www.lucidchart.com/pages/uml-sequence-diagram

Lucidchart. (n.d.-d). *UML State Machine Diagram Tutorial*. Lucidchart. https://www.lucidchart.com/pages/uml-state-machine-diagram

Lucidchart. (n.d.-e). *UML Use Case Diagram Tutorial*. Lucidchart. https://www.lucidchart.com/pages/uml-use-case-diagram

Microsoft (2024). *Microsoft Excel* (Version 2410 Build 18129.20158 Click-to-Run). Microsoft. https://www.microsoft.com/en-us/microsoft-365/excel

Microsoft (2024). *Microsoft Visio Professional 2021* (Version 2410 Build 18129.20158 Click-to-Run). Microsoft. https://www.microsoft.com/en-us/microsoft-365/visio/flowchart-software

Microsoft (2024). *Microsoft Word* (Version 2410 Build 18129.20158 Click-to-Run). Microsoft. https://www.microsoft.com/en-us/microsoft-365/word

ProjectManagement. (n.d.). Event Analysis Technique.

CNIT 38000 Diagram Portfolio https://www.projectmanagement.com/process/popup.cfm?ID=23545

Sebok Wiki. (n.d.). *System Requirements Definition*. Guide to the System Enginnering Body of Knowledge. https://sebokwiki.org/wiki/System_Requirements_Definition

UML Activity Diagrams. (n.d.). https://www.ibm.com/docs/en/rational-soft-arch/9.6.1?topic=diagrams-activity

Visual Paradigm (2024). Visual Paradigm Community Edition (Version: 17.2 Build Number: 20241101). Visual Paradigm. https://www.visual-paradigm.com/download/community.jsp

Visual Paradigm. (n.d.). UML Class Diagram Tutorial. https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/

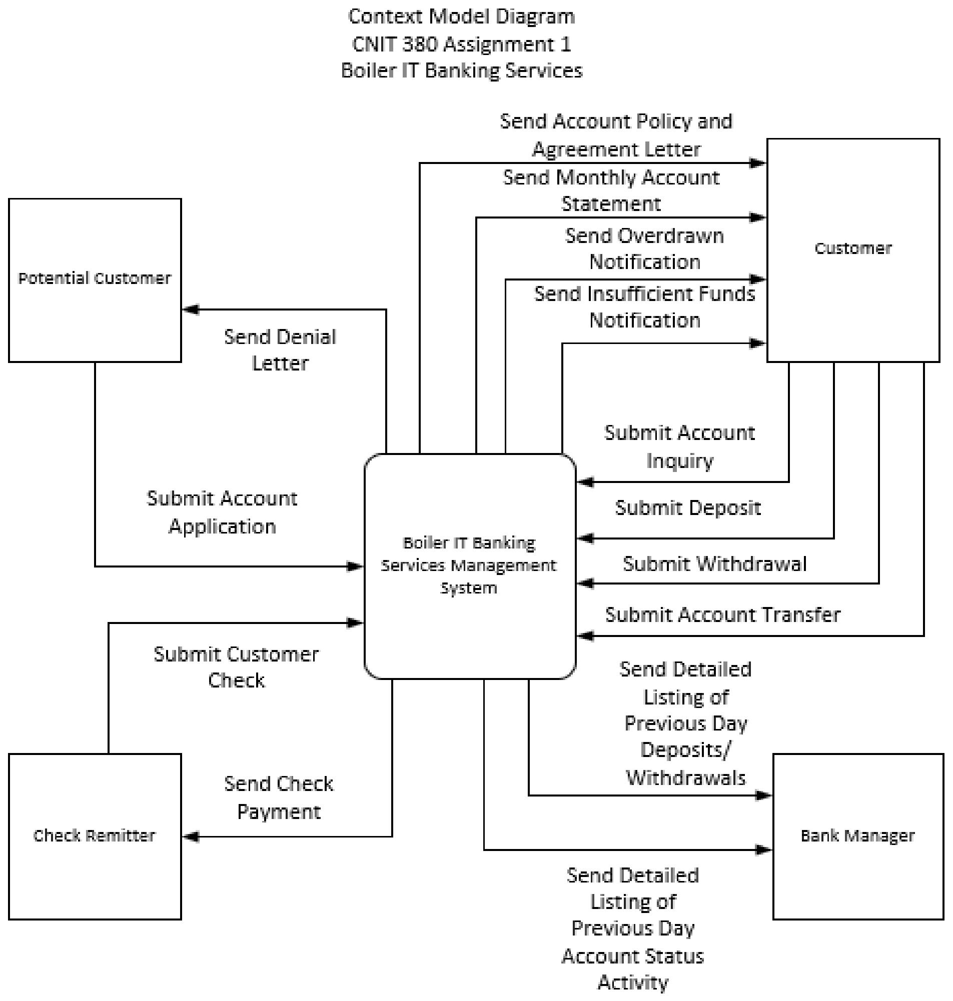
Visual Paradigm. (n.d.). *What is System Context Diagram*. What is System Context Diagram? https://online.visual-paradigm.com/knowledge/system-context-diagram/what-is-system-context-diagram/

Wikimedia Foundation. (2024, November 21). *Agile Software Development*. Wikipedia. https://en.wikipedia.org/wiki/Agile_software_development

APPENDIX: EXAMPLES OF UML DIAGRAMS AND DOCUMENTATION

The appendix contains comprehensive documentation of all of the UML diagrams created and referenced throughout this portfolio. Each diagram is included to provide a visual representation of the system's architecture, processes, and components, offering further clarity on the design and structure described in the main sections of the portfolio.

Context Model Diagram


Context Model Diagram
CNIT 380 Assignment 1
Boiler IT Banking Services
Send Account Policy and Agreement Letter
Send Monthly Account Statement
Send Overdrawn Notification
Send Insufficient Funds Notification
Customer
Potential Customer
Send Denial Letter
Submit Account Inquiry
Submit Account Application
Submit Deposit
Boiler IT Banking Services Management System
Submit Withdrawal
Submit Account Transfer
Submit Customer Check
Send Detailed Listing of Previous Day Deposits/ Withdrawals
Send Check Payment
Check Remitter
Bank Manager
Send Detailed Listing of Previous Day Account Status Activity

Assumptions:
- Application process for savings and checking accounts are the same

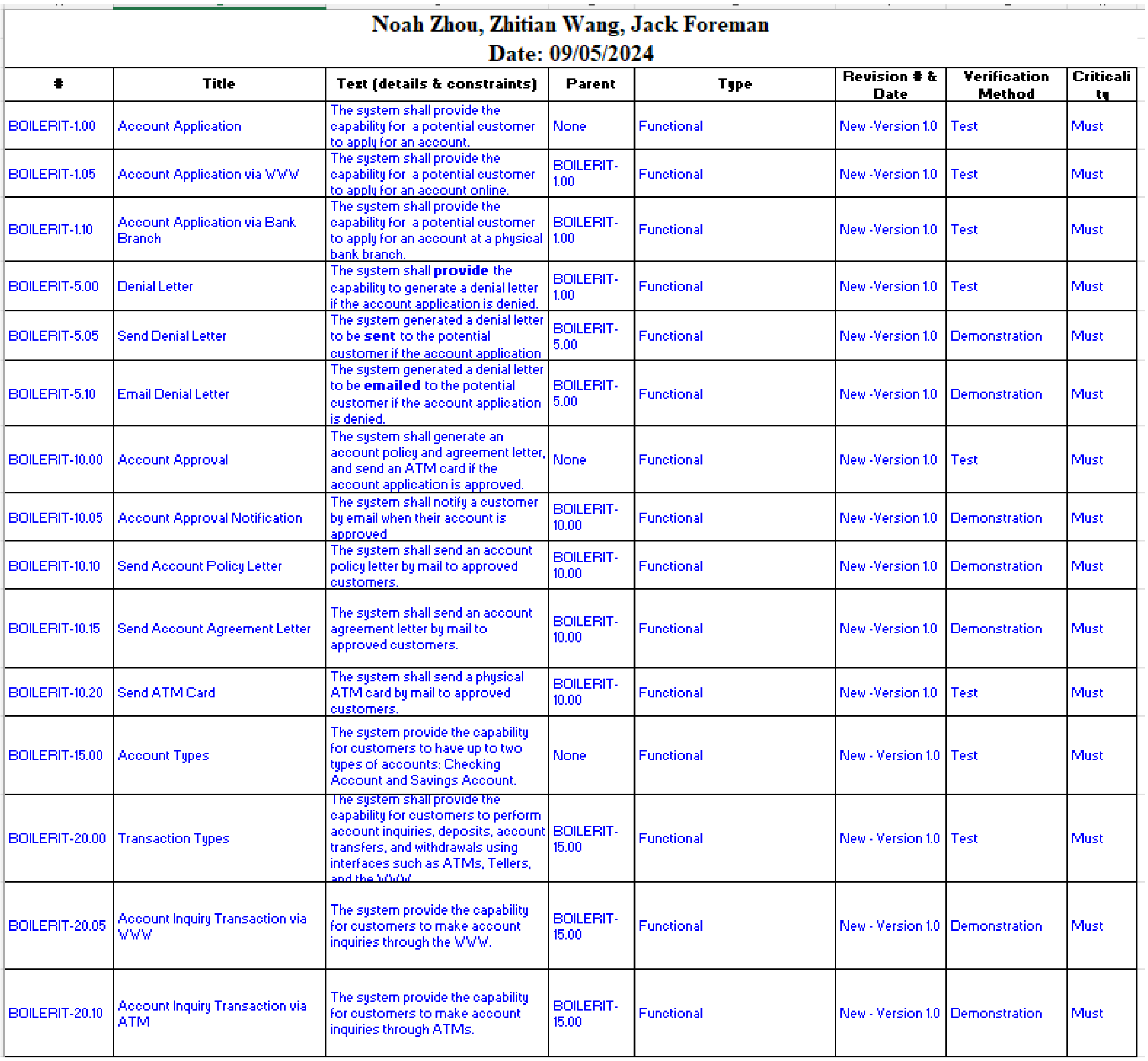
- Denial and Account Policy and Agreement Letters are the same for savings and checking accounts

- Bank account events are the same for savings and checking accounts

Names: Noah Zhou, Jack Foreman, Zhitian Wang
Date: 08/28/2024

Figure 1: This is the Context Model Diagram for the Boiler IT Banking Services Management System.

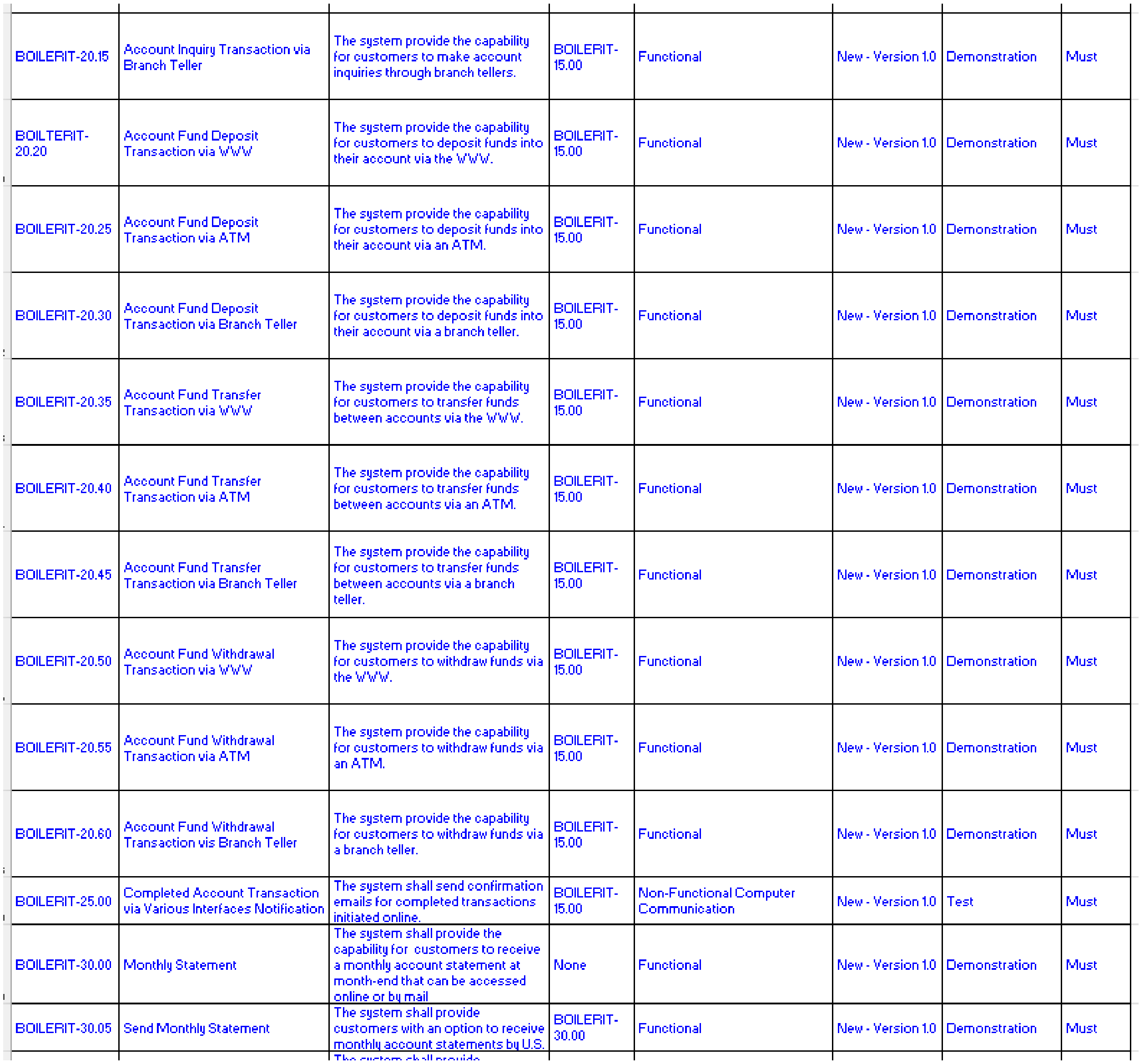
Requirements



Noah Zhou, Zhitian Wang, Jack Foreman
Date: 09/05/2024

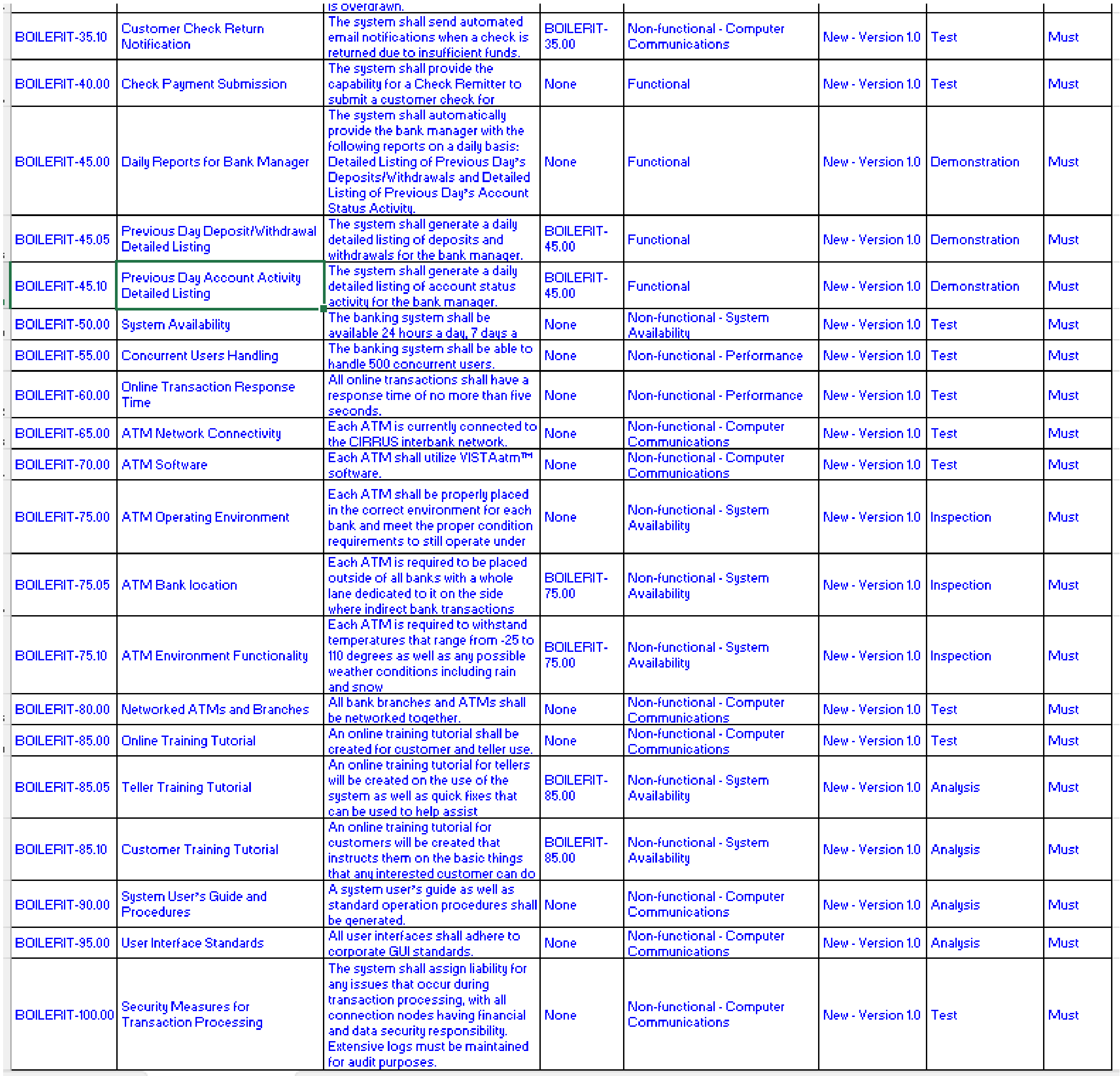
#	Title	Text (details & constraints)	Parent	Type	Revision # & Date	Verification Method	Criticality
BOILERIT-1.00	Account Application	The system shall provide the capability for a potential customer to apply for an account.	None	Functional	New -Version 1.0	Test	Must
BOILERIT-1.05	Account Application via WWW	The system shall provide the capability for a potential customer to apply for an account online.	BOILERIT-1.00	Functional	New -Version 1.0	Test	Must
BOILERIT-1.10	Account Application via Bank Branch	The system shall provide the capability for a potential customer to apply for an account at a physical bank branch.	BOILERIT-1.00	Functional	New -Version 1.0	Test	Must
BOILERIT-5.00	Denial Letter	The system shall **provide** the capability to generate a denial letter if the account application is denied.	BOILERIT-1.00	Functional	New -Version 1.0	Test	Must
BOILERIT-5.05	Send Denial Letter	The system generated a denial letter to be **sent** to the potential customer if the account application	BOILERIT-5.00	Functional	New -Version 1.0	Demonstration	Must
BOILERIT-5.10	Email Denial Letter	The system generated a denial letter to be **emailed** to the potential customer if the account application is denied.	BOILERIT-5.00	Functional	New -Version 1.0	Demonstration	Must
BOILERIT-10.00	Account Approval	The system shall generate an account policy and agreement letter, and send an ATM card if the account application is approved.	None	Functional	New -Version 1.0	Test	Must
BOILERIT-10.05	Account Approval Notification	The system shall notify a customer by email when their account is approved	BOILERIT-10.00	Functional	New -Version 1.0	Demonstration	Must
BOILERIT-10.10	Send Account Policy Letter	The system shall send an account policy letter by mail to approved customers.	BOILERIT-10.00	Functional	New -Version 1.0	Demonstration	Must
BOILERIT-10.15	Send Account Agreement Letter	The system shall send an account agreement letter by mail to approved customers.	BOILERIT-10.00	Functional	New -Version 1.0	Demonstration	Must
BOILERIT-10.20	Send ATM Card	The system shall send a physical ATM card by mail to approved customers.	BOILERIT-10.00	Functional	New -Version 1.0	Test	Must
BOILERIT-15.00	Account Types	The system provide the capability for customers to have up to two types of accounts: Checking Account and Savings Account.	None	Functional	New - Version 1.0	Test	Must
BOILERIT-20.00	Transaction Types	The system shall provide the capability for customers to perform account inquiries, deposits, account transfers, and withdrawals using interfaces such as ATMs, Tellers, and the WWW	BOILERIT-15.00	Functional	New - Version 1.0	Test	Must
BOILERIT-20.05	Account Inquiry Transaction via WWW	The system provide the capability for customers to make account inquiries through the WWW.	BOILERIT-15.00	Functional	New - Version 1.0	Demonstration	Must
BOILERIT-20.10	Account Inquiry Transaction via ATM	The system provide the capability for customers to make account inquiries through ATMs.	BOILERIT-15.00	Functional	New - Version 1.0	Demonstration	Must

Figure 2A: This is Part A of the System Requirements Analysis Spreadsheet.



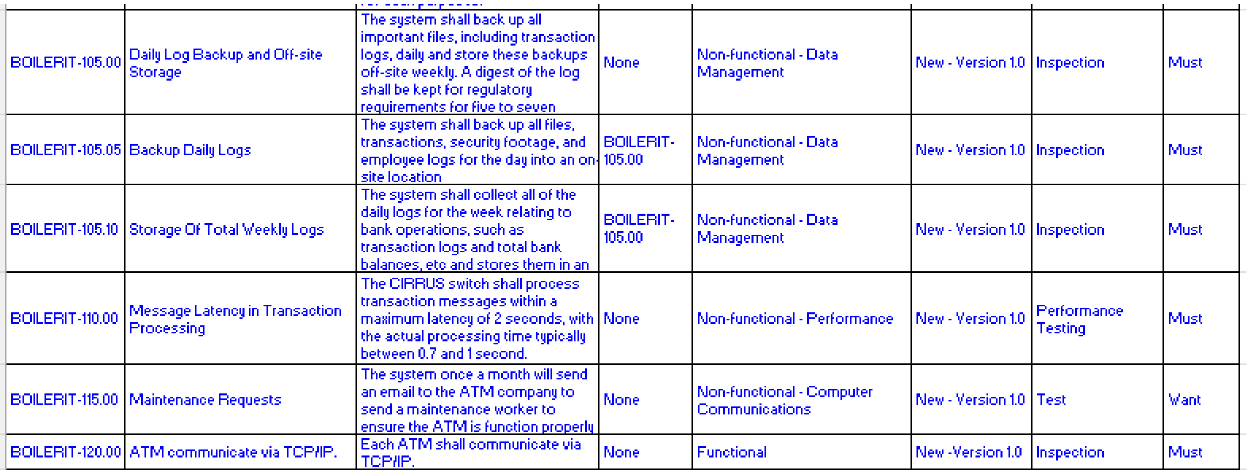
BOILERIT-20.15	Account Inquiry Transaction via Branch Teller	The system provide the capability for customers to make account inquiries through branch tellers.	BOILERIT-15.00	Functional	New - Version 1.0	Demonstration	Must
BOILTERIT-20.20	Account Fund Deposit Transaction via WWW	The system provide the capability for customers to deposit funds into their account via the WWW.	BOILERIT-15.00	Functional	New - Version 1.0	Demonstration	Must
BOILERIT-20.25	Account Fund Deposit Transaction via ATM	The system provide the capability for customers to deposit funds into their account via an ATM.	BOILERIT-15.00	Functional	New - Version 1.0	Demonstration	Must
BOILERIT-20.30	Account Fund Deposit Transaction via Branch Teller	The system provide the capability for customers to deposit funds into their account via a branch teller.	BOILERIT-15.00	Functional	New - Version 1.0	Demonstration	Must
BOILERIT-20.35	Account Fund Transfer Transaction via WWW	The system provide the capability for customers to transfer funds between accounts via the WWW.	BOILERIT-15.00	Functional	New - Version 1.0	Demonstration	Must
BOILERIT-20.40	Account Fund Transfer Transaction via ATM	The system provide the capability for customers to transfer funds between accounts via an ATM.	BOILERIT-15.00	Functional	New - Version 1.0	Demonstration	Must
BOILERIT-20.45	Account Fund Transfer Transaction via Branch Teller	The system provide the capability for customers to transfer funds between accounts via a branch teller.	BOILERIT-15.00	Functional	New - Version 1.0	Demonstration	Must
BOILERIT-20.50	Account Fund Withdrawal Transaction via WWW	The system provide the capability for customers to withdraw funds via the WWW.	BOILERIT-15.00	Functional	New - Version 1.0	Demonstration	Must
BOILERIT-20.55	Account Fund Withdrawal Transaction via ATM	The system provide the capability for customers to withdraw funds via an ATM.	BOILERIT-15.00	Functional	New - Version 1.0	Demonstration	Must
BOILERIT-20.60	Account Fund Withdrawal Transaction vis Branch Teller	The system provide the capability for customers to withdraw funds via a branch teller.	BOILERIT-15.00	Functional	New - Version 1.0	Demonstration	Must
BOILERIT-25.00	Completed Account Transaction via Various Interfaces Notification	The system shall send confirmation emails for completed transactions initiated online.	BOILERIT-15.00	Non-Functional Computer Communication	New - Version 1.0	Test	Must
BOILERIT-30.00	Monthly Statement	The system shall provide the capability for customers to receive a monthly account statement at month-end that can be accessed online or by mail	None	Functional	New - Version 1.0	Demonstration	Must
BOILERIT-30.05	Send Monthly Statement	The system shall provide customers with an option to receive monthly account statements by U.S.	BOILERIT-30.00	Functional	New - Version 1.0	Demonstration	Must
		The system shall provide					

Figure 2B: This is Part B of the System Requirements Analysis Spreadsheet.



		is overdrawn.					
BOILERIT-35.10	Customer Check Return Notification	The system shall send automated email notifications when a check is returned due to insufficient funds.	BOILERIT-35.00	Non-functional - Computer Communications	New - Version 1.0	Test	Must
BOILERIT-40.00	Check Payment Submission	The system shall provide the capability for a Check Remitter to submit a customer check for	None	Functional	New - Version 1.0	Test	Must
BOILERIT-45.00	Daily Reports for Bank Manager	The system shall automatically provide the bank manager with the following reports on a daily basis: Detailed Listing of Previous Day's Deposits/Withdrawals and Detailed Listing of Previous Day's Account Status Activity.	None	Functional	New - Version 1.0	Demonstration	Must
BOILERIT-45.05	Previous Day Deposit/Withdrawal Detailed Listing	The system shall generate a daily detailed listing of deposits and withdrawals for the bank manager.	BOILERIT-45.00	Functional	New - Version 1.0	Demonstration	Must
BOILERIT-45.10	Previous Day Account Activity Detailed Listing	The system shall generate a daily detailed listing of account status activity for the bank manager.	BOILERIT-45.00	Functional	New - Version 1.0	Demonstration	Must
BOILERIT-50.00	System Availability	The banking system shall be available 24 hours a day, 7 days a	None	Non-functional - System Availability	New - Version 1.0	Test	Must
BOILERIT-55.00	Concurrent Users Handling	The banking system shall be able to handle 500 concurrent users.	None	Non-functional - Performance	New - Version 1.0	Test	Must
BOILERIT-60.00	Online Transaction Response Time	All online transactions shall have a response time of no more than five seconds.	None	Non-functional - Performance	New - Version 1.0	Test	Must
BOILERIT-65.00	ATM Network Connectivity	Each ATM is currently connected to the CIRRUS interbank network.	None	Non-functional - Computer Communications	New - Version 1.0	Test	Must
BOILERIT-70.00	ATM Software	Each ATM shall utilize VISTAatm™ software.	None	Non-functional - Computer Communications	New - Version 1.0	Test	Must
BOILERIT-75.00	ATM Operating Environment	Each ATM shall be properly placed in the correct environment for each bank and meet the proper condition requirements to still operate under	None	Non-functional - System Availability	New - Version 1.0	Inspection	Must
BOILERIT-75.05	ATM Bank location	Each ATM is required to be placed outside of all banks with a whole lane dedicated to it on the side where indirect bank transactions	BOILERIT-75.00	Non-functional - System Availability	New - Version 1.0	Inspection	Must
BOILERIT-75.10	ATM Environment Functionality	Each ATM is required to withstand temperatures that range from -25 to 110 degrees as well as any possible weather conditions including rain and snow	BOILERIT-75.00	Non-functional - System Availability	New - Version 1.0	Inspection	Must
BOILERIT-80.00	Networked ATMs and Branches	All bank branches and ATMs shall be networked together.	None	Non-functional - Computer Communications	New - Version 1.0	Test	Must
BOILERIT-85.00	Online Training Tutorial	An online training tutorial shall be created for customer and teller use.	None	Non-functional - Computer Communications	New - Version 1.0	Test	Must
BOILERIT-85.05	Teller Training Tutorial	An online training tutorial for tellers will be created on the use of the system as well as quick fixes that can be used to help assist	BOILERIT-85.00	Non-functional - System Availability	New - Version 1.0	Analysis	Must
BOILERIT-85.10	Customer Training Tutorial	An online training tutorial for customers will be created that instructs them on the basic things that any interested customer can do	BOILERIT-85.00	Non-functional - System Availability	New - Version 1.0	Analysis	Must
BOILERIT-90.00	System User's Guide and Procedures	A system user's guide as well as standard operation procedures shall be generated.	None	Non-functional - Computer Communications	New - Version 1.0	Analysis	Must
BOILERIT-95.00	User Interface Standards	All user interfaces shall adhere to corporate GUI standards.	None	Non-functional - Computer Communications	New - Version 1.0	Analysis	Must
BOILERIT-100.00	Security Measures for Transaction Processing	The system shall assign liability for any issues that occur during transaction processing, with all connection nodes having financial and data security responsibility. Extensive logs must be maintained for audit purposes.	None	Non-functional - Computer Communications	New - Version 1.0	Test	Must

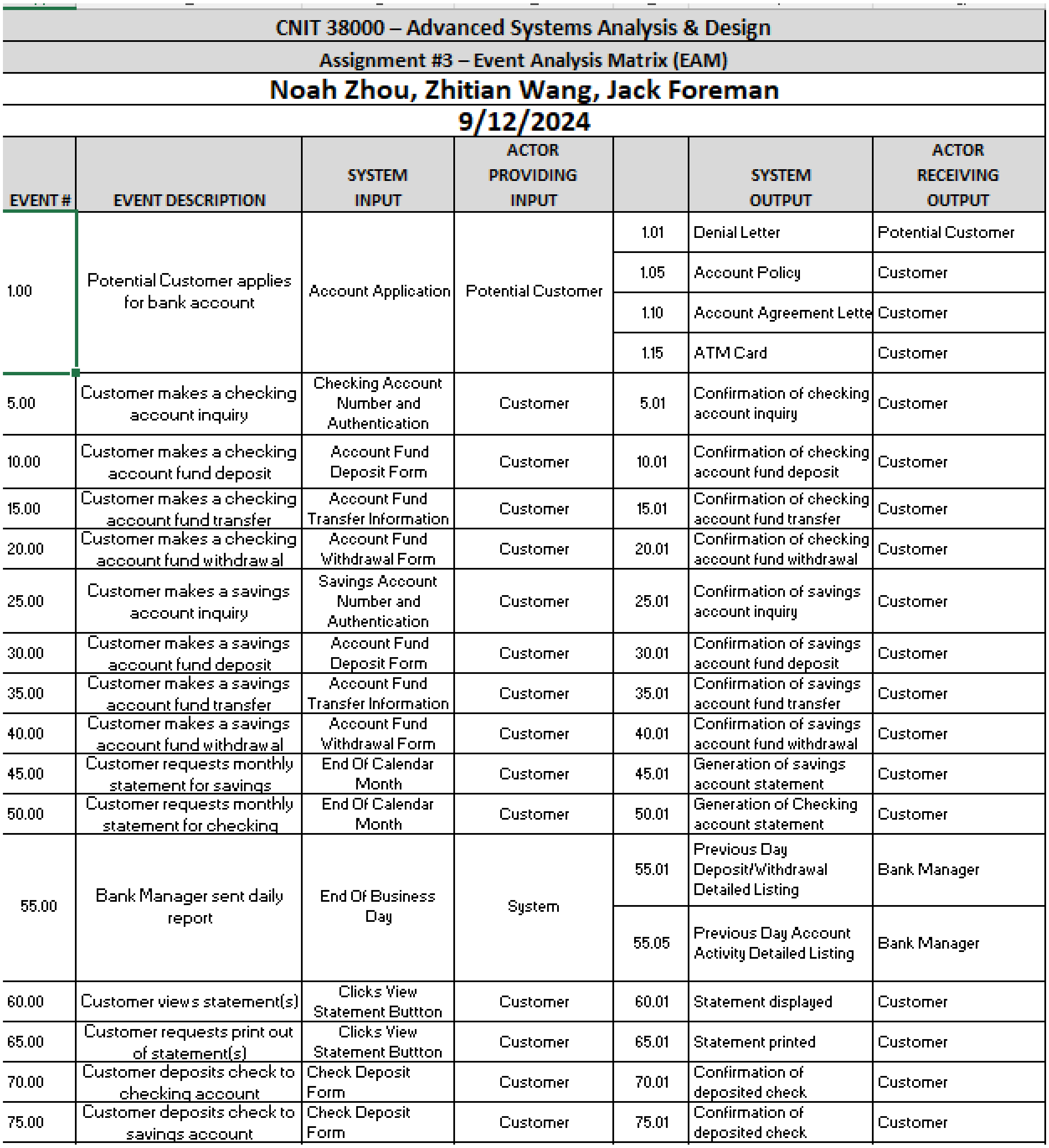
Figure 2C: This is Part C of the System Requirements Analysis Spreadsheet.



BOILERIT-105.00	Daily Log Backup and Off-site Storage	The system shall back up all important files, including transaction logs, daily and store these backups off-site weekly. A digest of the log shall be kept for regulatory requirements for five to seven	None	Non-functional - Data Management	New - Version 1.0	Inspection	Must
BOILERIT-105.05	Backup Daily Logs	The system shall back up all files, transactions, security footage, and employee logs for the day into an on-site location	BOILERIT-105.00	Non-functional - Data Management	New - Version 1.0	Inspection	Must
BOILERIT-105.10	Storage Of Total Weekly Logs	The system shall collect all of the daily logs for the week relating to bank operations, such as transaction logs and total bank balances, etc and stores them in an	BOILERIT-105.00	Non-functional - Data Management	New - Version 1.0	Inspection	Must
BOILERIT-110.00	Message Latency in Transaction Processing	The CIRRUS switch shall process transaction messages within a maximum latency of 2 seconds, with the actual processing time typically between 0.7 and 1 second.	None	Non-functional - Performance	New - Version 1.0	Performance Testing	Must
BOILERIT-115.00	Maintenance Requests	The system once a month will send an email to the ATM company to send a maintenance worker to ensure the ATM is function properly	None	Non-functional - Computer Communications	New - Version 1.0	Test	Want
BOILERIT-120.00	ATM communicate via TCP/IP.	Each ATM shall communicate via TCP/IP.	None	Functional	New -Version 1.0	Inspection	Must

Figure 2D: This is Part D of the System Requirements Analysis Spreadsheet.

Event Analysis Matrix



CNIT 38000 – Advanced Systems Analysis & Design

Assignment #3 – Event Analysis Matrix (EAM)

Noah Zhou, Zhitian Wang, Jack Foreman

9/12/2024

EVENT #	EVENT DESCRIPTION	SYSTEM INPUT	ACTOR PROVIDING INPUT		SYSTEM OUTPUT	ACTOR RECEIVING OUTPUT
1.00	Potential Customer applies for bank account	Account Application	Potential Customer	1.01	Denial Letter	Potential Customer
				1.05	Account Policy	Customer
				1.10	Account Agreement Lette	Customer
				1.15	ATM Card	Customer
5.00	Customer makes a checking account inquiry	Checking Account Number and Authentication	Customer	5.01	Confirmation of checking account inquiry	Customer
10.00	Customer makes a checking account fund deposit	Account Fund Deposit Form	Customer	10.01	Confirmation of checking account fund deposit	Customer
15.00	Customer makes a checking account fund transfer	Account Fund Transfer Information	Customer	15.01	Confirmation of checking account fund transfer	Customer
20.00	Customer makes a checking account fund withdrawal	Account Fund Withdrawal Form	Customer	20.01	Confirmation of checking account fund withdrawal	Customer
25.00	Customer makes a savings account inquiry	Savings Account Number and Authentication	Customer	25.01	Confirmation of savings account inquiry	Customer
30.00	Customer makes a savings account fund deposit	Account Fund Deposit Form	Customer	30.01	Confirmation of savings account fund deposit	Customer
35.00	Customer makes a savings account fund transfer	Account Fund Transfer Information	Customer	35.01	Confirmation of savings account fund transfer	Customer
40.00	Customer makes a savings account fund withdrawal	Account Fund Withdrawal Form	Customer	40.01	Confirmation of savings account fund withdrawal	Customer
45.00	Customer requests monthly statement for savings	End Of Calendar Month	Customer	45.01	Generation of savings account statement	Customer
50.00	Customer requests monthly statement for checking	End Of Calendar Month	Customer	50.01	Generation of Checking account statement	Customer
55.00	Bank Manager sent daily report	End Of Business Day	System	55.01	Previous Day Deposit/Withdrawal Detailed Listing	Bank Manager
				55.05	Previous Day Account Activity Detailed Listing	Bank Manager
60.00	Customer views statement(s)	Clicks View Statement Buttton	Customer	60.01	Statement displayed	Customer
65.00	Customer requests print out of statement(s)	Clicks View Statement Buttton	Customer	65.01	Statement printed	Customer
70.00	Customer deposits check to checking account	Check Deposit Form	Customer	70.01	Confirmation of deposited check	Customer
75.00	Customer deposits check to savings account	Check Deposit Form	Customer	75.01	Confirmation of deposited check	Customer

Figure 3: This is the Event Analysis Matrix that details the primary events that occur in the system.

Use Case Diagram

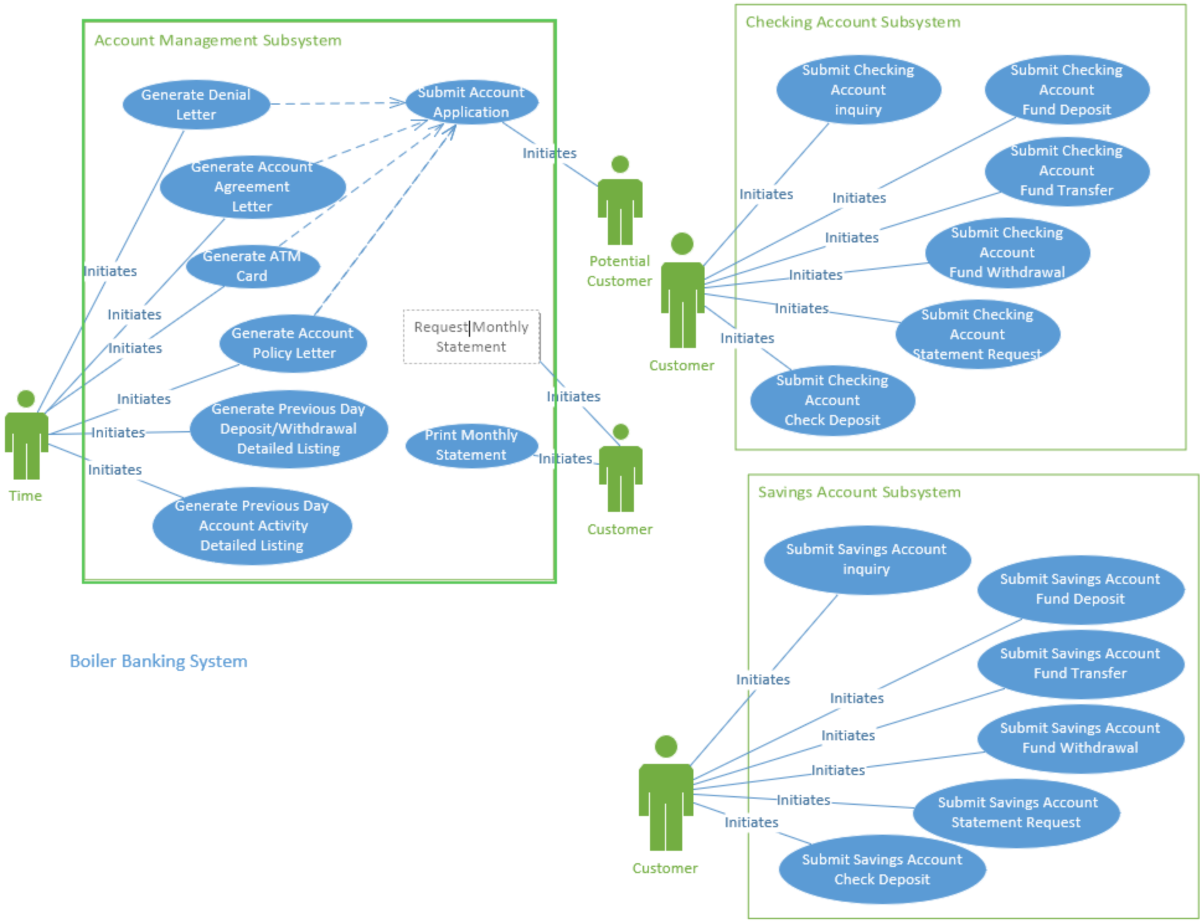

Account Management Subsystem
Generate Denial Letter
Submit Account Application
Generate Account Agreement Letter
Generate ATM Card
Generate Account Policy Letter
Generate Previous Day Deposit/Withdrawal Detailed Listing
Generate Previous Day Account Activity Detailed Listing
Request Monthly Statement
Print Monthly Statement
Initiates
Time
Potential Customer
Customer
Boiler Banking System
Checking Account Subsystem
Submit Checking Account inquiry
Submit Checking Account Fund Deposit
Submit Checking Account Fund Transfer
Submit Checking Account Fund Withdrawal
Submit Checking Account Statement Request
Submit Checking Account Check Deposit
Savings Account Subsystem
Submit Savings Account inquiry
Submit Savings Account Fund Deposit
Submit Savings Account Fund Transfer
Submit Savings Account Fund Withdrawal
Submit Savings Account Statement Request
Submit Savings Account Check Deposit

Figure 4: This is the use case diagram that depicts the main actions in the bank system.

Use Case Narratives

Account Management Subsystem

Author (s): Group 2 **Date:2024-09-26**

Version: 1.00

USE CASE NAME:	Submit Account Application	**USE CASE TYPE & LEVEL**	
USE CASE ID:	AMS-UC001	**Business:**	□
PRIORITY:	High	***System:*** ***Requirements***	□
SOURCE:	Requirements document	**Analysis** **Design**	□ ☑
PRIMARY BUSINESS ACTOR	Potential customer		
PRIMARY SYSTEM ACTOR	Bank Manager		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	Banking Institution, Regulatory Authority		
DESCRIPTION:	This use case describes the process of a potential customer submitting an account application to the bank. The bank manager reviews the application for errors or omissions. If the information is correct, the system sends a request to the credit bureau for a credit report. Based on the credit report, the manager either approves or denies the account. If approved, the system generates an account number, stores the account information, and creates an account identification card. If denied, the system generates a rejection letter and sends it to the potential customer.		
PRE-CONDITION:	The potential customer has completed the application form correctly		

TRIGGER:	The potential customer submits the application to the bank manager.	
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE
	Step 1: The potential customer submits an account application form.	
	Step 2: The bank manager checks the application for any errors or omissions.	
	Step 3: The bank manager confirms that the application is correct and has no errors or omissions	
	Step 4: The bank manager logs into the system	**Step 6:** The system prompts the manager for new account information (name, address, DOB, phone number, SSN, etc.).
	Step 7: The bank manager enters in information for the new account (Name, Address, DOB, Phone, SSN, etc.)	**Step 8:** The system verifies the information in terms of data and format checking
		Step 9: The information is correct and the system sends a request to the credit bureau for a report and credit score
		Step 10: The system displays the report and score and prompts the bank manager to approve or deny the application
	Step 11: The bank manager reviews the report	
	Step 12: The bank manager approves the account application	**Step 13:** The system generates the account number
		Step 14: The system stores the account information and generates an account identification card
ALTERNATE COURSES:	**Alt-Step 3:** The bank manager finds errors or omissions and requests the potential customer to resolve and resubmit the application **(Go back to Step 1)**	

	Alt Step 4: The bank manager is already logged into the system
	Alt-Step 9: The information is incorrect and the system prompts the manager to correct information **(Go back to Step 8)**
	Alt-Step 12: The bank manager denies the account application and notifies the potential customer that their application has been denied
	Alt-Step 13: The system stores the application as rejected
	Alt-Step 14: The system generates a letter stating the reasons to why the application was rejected
	Alt-Step 15: The letter is then sent to the potential customer
CONCLUSION:	The potential customer either receives an account or is informed of their application denial along with reasons.
POST-CONDITION:	A new bank account is created if approved; otherwise, the application is stored as rejected.
BUSINESS RULES	• Applications must be completed accurately to proceed. • Creditworthiness must be verified before account approval.
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• The system must comply with banking regulations for data handling.
ASSUMPTIONS:	• The bank manager has the necessary permissions to open accounts. • The potential customer has provided accurate information.
OPEN ISSUES:	• What specific criteria are used for creditworthiness evaluation?

Figure 5A: This is the first out of six use case narratives that describe the events that occur when a potential customer submits an account application.

Account Management Subsystem

Author (s): **Group 2** **Date:** **2024-09-26**

Version: **1.00**

USE CASE NAME:	Credit Report Request	**USE CASE TYPE & LEVEL**
USE CASE ID:	AMS-UC001.05	**Business:** □
PRIORITY:	High	***System:*** ***Requirements*** □

SOURCE:	Requirements document	**Analysis** □ **Design** ☑
PRIMARY BUSINESS ACTOR	Bank System	
PRIMARY SYSTEM ACTOR	Credit Bureau	
OTHER PARTICIPATING ACTORS:	None	
OTHER INTERESTED STAKEHOLDERS:	Banking Institutions, Potential Customers	
DESCRIPTION:	This use case describes the process through which the credit bureau receives a request for a credit report from a bank and returns the necessary credit information for the bank manager to determine the creditworthiness of the potential customer before accepting or denying the account application.	
PRE-CONDITION:	A request for a credit report has been made by the bank system.	
TRIGGER:	The bank manager submits a request for a credit report based on the potential customer's application.	
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE
	Step 1: The bank system sends a request for a credit report to the credit bureau	**Step 2:** The credit bureau processes the request using the given information **(Name, Address, DOB, Phone, SSN, etc.)**
		Step 3: The credit bureau retrieves the credit report and credit score
		Step 4: The credit bureau sends the credit report and credit score back to the bank system

	Step 5: The bank system displays the credit report information to the bank manager	
ALTERNATE COURSES:	**Alt-Step 3:** If the provided information is insufficient to retrieve a report, the credit bureau request additional information from the bank. **(Go back to Step 2)**	
CONCLUSION:	The bank manager receives the credit report necessary for the application process.	
POST-CONDITION:	The credit report is successfully delivered to the bank's system for evaluation.	
BUSINESS RULES	• Credit reports must be delivered securely and within a defined timeframe. • Only authorized requests from financial institutions are processed.	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• Compliance with privacy laws regarding the handling of sensitive information.	
ASSUMPTIONS:	• The credit bureau maintains accurate and up-to-date credit information.	
OPEN ISSUES:	• How are credit freezes handled?	

Figure 5B: This is the second out of six use case narratives that detail the events that occur when the bank system submits a credit report request.

Bank System

Author (s): **Group 2** **Date:** **2024-09-26**

Version: **1.00**____

USE CASE NAME:	Generate Previous Day Deposit and Withdrawal Report	**USE CASE TYPE & LEVEL**
USE CASE ID:	BS-UC005.00	**Business:** □
PRIORITY:	High	***System:*** ***Requirements*** □
SOURCE:	Requirements document	**Analysis** □ **Design** ☑

PRIMARY BUSINESS ACTOR	Bank Manager	
PRIMARY SYSTEM ACTOR	Bank System	
OTHER PARTICIPATING ACTORS:	None	
OTHER INTERESTED STAKEHOLDERS:	Banking Institution	
DESCRIPTION:	This use case describes the process by which the system generates a report detailing deposits and withdrawals from customer accounts for a specified previous day.	
PRE-CONDITION:	The system is configured to track all deposits and withdrawals and the bank manager is logged into the system with proper permissions.	
TRIGGER:	The bank manager requests a daily report of deposits and withdrawals for the previous business day.	
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE
	Step 1: The bank manager logs into the bank system.	
	Step 2: The bank manager selects the option to generate the previous day's deposit and withdrawal report.	**Step 3:** The system processes the request and retrieves deposit and withdrawal data for the specified day from the database.
		Step 4: The system generates the report, listing detailed transactions (account number, type of transaction, amount, time, etc.).
		Step 5: The system displays the report to the bank manager for review
	Step 6: Bank manager reviews report	

ALTERNATE COURSES:	**Alt-Step 5:** If there is an issue generating the report, the system displays an error message
CONCLUSION:	The bank manager successfully generates and reviews a daily report of deposits and withdrawals.
POST-CONDITION:	The report is available for printing or downloading, and the system logs the generation event.
BUSINESS RULES	• Reports must be accurate and generated for completed business days only. • Only authorized users can generate transaction reports.
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• The system must handle large amounts of transaction data efficiently.
ASSUMPTIONS:	• The system has access to up-to-date transaction data. • The bank manager has the necessary permissions to request reports.
OPEN ISSUES:	• How long should reports remain accessible in the system?

Figure 5C: This is the third out of six use case narratives that detail the events that occur when the bank manager wants to view the previous day's deposit and withdrawal report.

Bank System

Author (s): Group 2 **Date: 2024-09-26**

Version: 1.00

USE CASE NAME:	Generate Previous Day Account Activity Report	**USE CASE TYPE & LEVEL**
USE CASE ID:	BS-UC005.05	**Business:** □
PRIORITY:	High	***System:*** ***Requirements*** □
SOURCE:	Requirements document	**Analysis** □ **Design** ☑
PRIMARY BUSINESS ACTOR	Bank Manager	
PRIMARY SYSTEM ACTOR	Bank System	

OTHER PARTICIPATING ACTORS:	None	
OTHER INTERESTED STAKEHOLDERS:	Banking Institution	
DESCRIPTION:	This use case describes the process by which the system generates a report summarizing account activity for all customer accounts for the previous business day.	
PRE-CONDITION:	The system has recorded all transactions (deposits, withdrawals, account transfers, etc.) for the previous day, and the bank manager is logged into the system.	
TRIGGER:	The bank manager requests a daily account activity report for the previous business day.	
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE
	Step 1: The bank manager logs into the bank system.	
	Step 2: The bank manager selects the option to generate the previous day's account activity report.	**Step 3:** The system processes the request, retrieving account activity data (deposits, withdrawals, transfers, fees, etc.) for all customer accounts.
		Step 4: The system compiles the account activity data into a report, organized by account number and type of transaction.
		Step 5: The system displays the report for review
	Step 6: Bank manager reviews report	
ALTERNATE COURSES:	**Alt-Step 5:** If there is an issue generating the report, the system displays an error message.	
CONCLUSION:	The bank manager successfully generates and reviews a report summarizing account activity for the previous day.	
POST-CONDITION:	The report is available for printing or downloading, and the system logs the generation event.	

BUSINESS RULES	• Reports must accurately reflect all account activity for the specified business day. • Only authorized users can generate account activity reports.
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• The system must handle large datasets efficiently and produce reports within a reasonable timeframe.
ASSUMPTIONS:	• The system has access to accurate and up-to-date transaction data. • The bank manager has the necessary permissions to generate reports.
OPEN ISSUES:	• How long should reports remain accessible in the system?

Figure 5D: This is the fourth out of six use case narratives that detail the events that occur when the bank manager wants to view the previous day's account activity report.

Checking Account Subsystem

Author (s): **Group 2** **Date:** **2024-09-26**

Version: 1.00

USE CASE NAME:	Checking Account Actions	**USE CASE TYPE & LEVEL** **Business:** □ ***System:*** ***Requirements*** □ **Analysis** □ **Design** ☑
USE CASE ID:	CAS-UC001.00	
PRIORITY:	High	
SOURCE:	Requirements document	
PRIMARY BUSINESS ACTOR	Customer with Checking Account	
PRIMARY SYSTEM ACTOR	Bank System	
OTHER PARTICIPATING ACTORS:	Teller (optional for in-branch transactions)	

OTHER INTERESTED STAKEHOLDERS:	Banking Institution	
DESCRIPTION:	This use case describes the various actions a customer with a checking account can perform, including checking account inquiries, fund deposits, withdrawals, fund transfers, statement requests, and check deposits.	
PRE-CONDITION:	The customer must have an active checking account, and the bank system must be operational and connected to the database.	
TRIGGER:	The customer initiates an inquiry or transaction related to their checking account.	
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE
	Step 1: The customer logs into their online banking portal	**Step 2:** The system displays actions that can be performed in the checking account
	Step 3A: The customer selects "Account Inquiry" from the available options.	**Step 4A**: The system displays the account information to the customer.
	Step 3B: The customer selects “Fund Deposit” from the available options specifies the amount and source of funds.	**Step 4B:** The system confirms the deposit and updates transaction history
	Step 3C: The customer selects "Fund Withdrawal" and specifies the amount to withdraw.	**Step 4C**: The system verifies the withdrawal request and ensures sufficient funds are available.
	Step 3D: The customer selects "Fund Transfer" and specifies the recipient account and amount to transfer.	**Step 4D**: The system verifies the recipient account, checks available funds, and processes the transfer.
	Step 3E: The customer selects "Statement Request" for a specific date range.	**Step 4E**: The system retrieves the transaction history for the requested period and generates a statement.
	Step 3F: The customer selects "Check Deposit" and submits a photo of the check	**Step 4F**: The system verifies the check information and processes the deposit and then confirms the deposit and updates the account balance

ALTERNATE COURSES:	**Alt-Step 4**: If any issues arise (e.g., insufficient funds, incorrect account information), the system notifies the customer of the error and prompts for correction.
CONCLUSION:	The customer successfully completes the selected checking account action.
POST-CONDITION:	The checking account balance and transaction history are updated based on the performed action.
BUSINESS RULES	• Deposits and withdrawals are subject to bank policies and limits. • Fund transfers must verify the recipient account before processing.
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• The system must ensure data accuracy and handle multiple transactions simultaneously.
ASSUMPTIONS:	• The customer's account is active and the system has access to up-to-date information.
OPEN ISSUES:	• How will the system handle transactions during system downtimes?

Figure 5E: This is the fifth out of six use case narratives that detail the events that occur when a customer accesses their checking account.

Savings Account Subsystem

Author (s): **Group 2** **Date:** **2024-09-26**

Version: **1.00**

USE CASE NAME:	Savings Account Actions	**USE CASE TYPE & LEVEL**
USE CASE ID:	SAS-UC001.00	**Business:** □
PRIORITY:	High	***System:*** ***Requirements*** □
SOURCE:	Requirements document	**Analysis** □ **Design** ☑
PRIMARY BUSINESS ACTOR	Customer with Savings Account	
PRIMARY SYSTEM ACTOR	Bank System	

OTHER PARTICIPATING ACTORS:	Teller (optional for in-branch transactions)	
OTHER INTERESTED STAKEHOLDERS:	Banking Institution	
DESCRIPTION:	This use case describes the various actions a customer with a savings account can perform, including savings account inquiries, fund deposits, withdrawals, fund transfers, statement requests, and check deposits.	
PRE-CONDITION:	The customer must have an active savings account, and the bank system must be operational and connected to the database.	
TRIGGER:	The customer initiates an inquiry or transaction related to their savings account.	
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE
	Step 1: The customer logs into their online banking portal	**Step 2:** The system displays actions that can be performed in the savings account
	Step 3A: The customer selects "Account Inquiry" from the available options.	**Step 4A**: The system displays the account information to the customer.
	Step 3B: The customer selects “Fund Deposit” from the available options specifies the amount and source of funds.	**Step 4B:** The system confirms the deposit and updates transaction history
	Step 3C: The customer selects "Fund Withdrawal" and specifies the amount to withdraw.	**Step 4C**: The system verifies the withdrawal request and ensures sufficient funds are available.
	Step 3D: The customer selects "Fund Transfer" and specifies the recipient account and amount to transfer.	**Step 4D**: The system verifies the recipient account, checks available funds, and processes the transfer.
	Step 3E: The customer selects "Statement Request" for a specific date range.	**Step 4E**: The system retrieves the transaction history for the requested period and generates a statement.

	Step 3F: The customer selects "Check Deposit" and submits a photo of the check	**Step 4F**: The system verifies the check information and processes the deposit and then confirms the deposit and updates the account balance
ALTERNATE COURSES:	**Alt-Step 4**: If any issues arise (e.g., insufficient funds, incorrect account information), the system notifies the customer of the error and prompts for correction.	
CONCLUSION:	The customer successfully completes the selected savings account action.	
POST-CONDITION:	The savings account balance and transaction history are updated based on the performed action.	
BUSINESS RULES	• Deposits and withdrawals are subject to bank policies and limits. • Fund transfers must verify the recipient account before processing.	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• The system must ensure data accuracy and handle multiple transactions simultaneously.	
ASSUMPTIONS:	• The customer's account is active and the system has access to up-to-date information.	
OPEN ISSUES:	• How will the system handle transactions during system downtimes?	

Figure 5F: This is the sixth and final use case narrative that details the events that occur when the customer accesses their savings account.

Activity Diagram

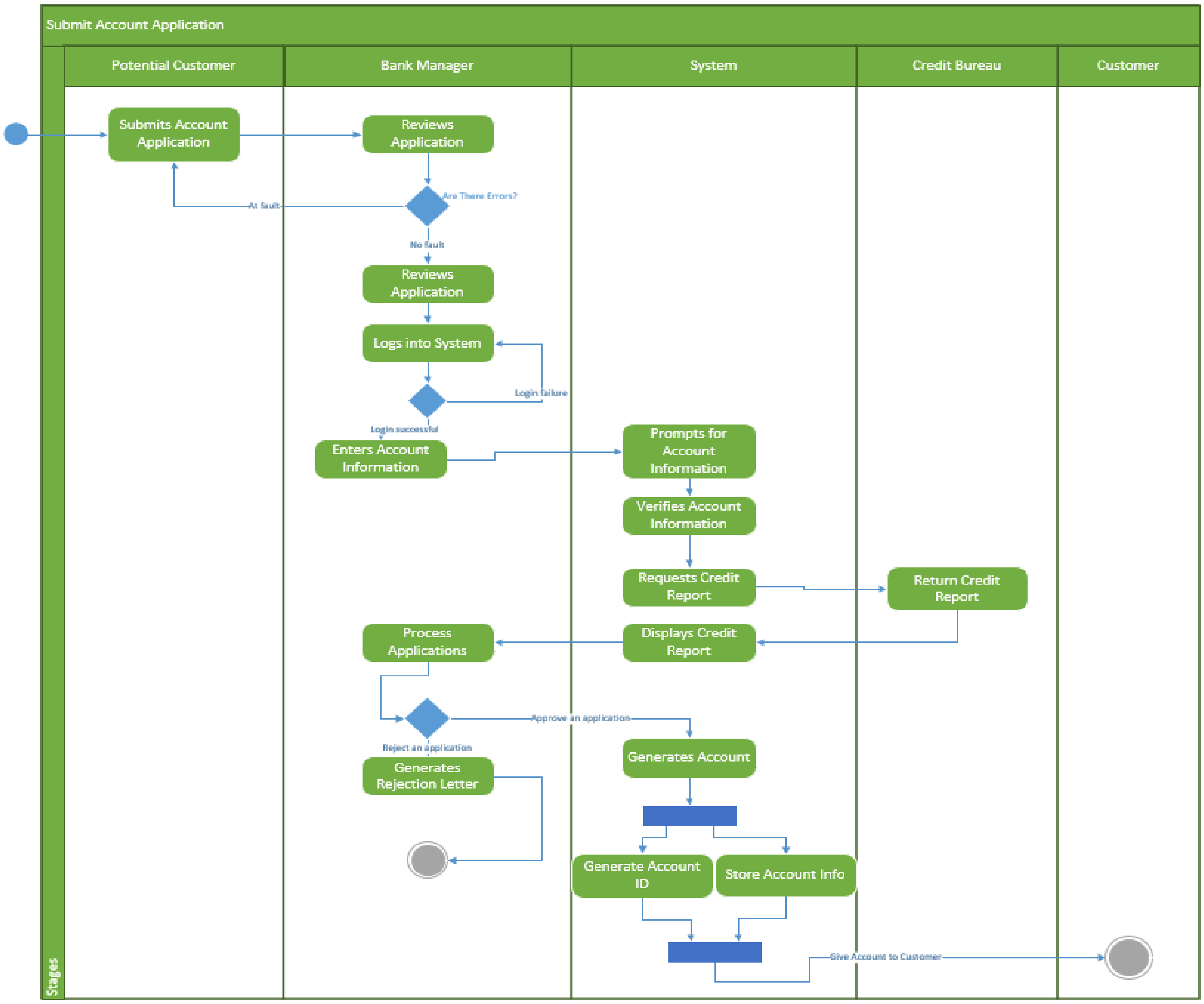


Figure 6: This is the Activity Diagram for the Boiler IT Banking Services System that depicts a sample event flow for when a potential customer applies for an account.

Class Diagram

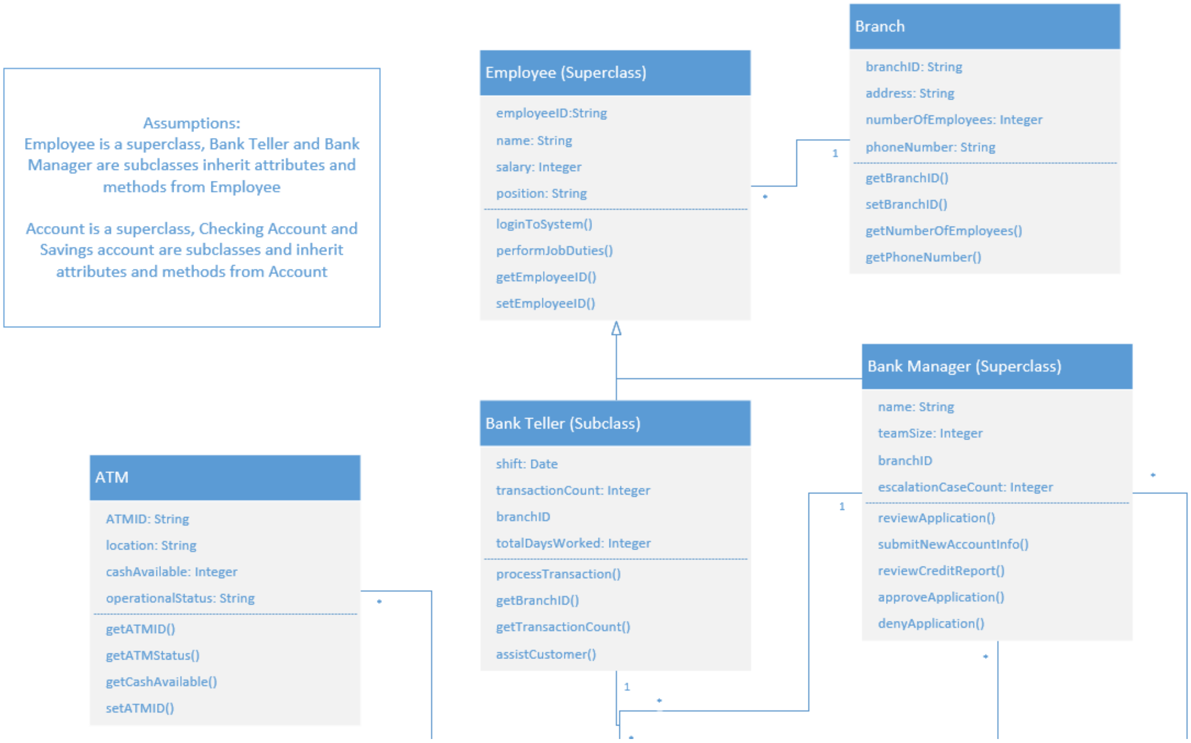


Figure 7A: This is Part A of the Class Diagram which depicts the objects and attributes in the Boiler IT Banking Services System.

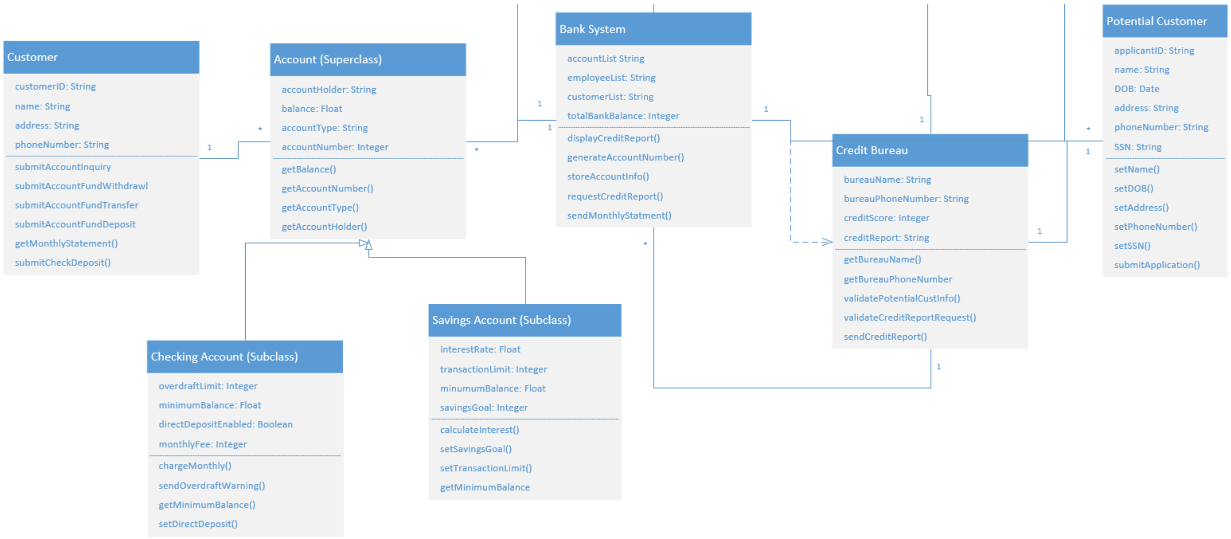


Figure 7B: This is Part B of the Class Diagram which depicts the objects and attributes in the Boiler IT Banking Services System.

State Machine Diagram

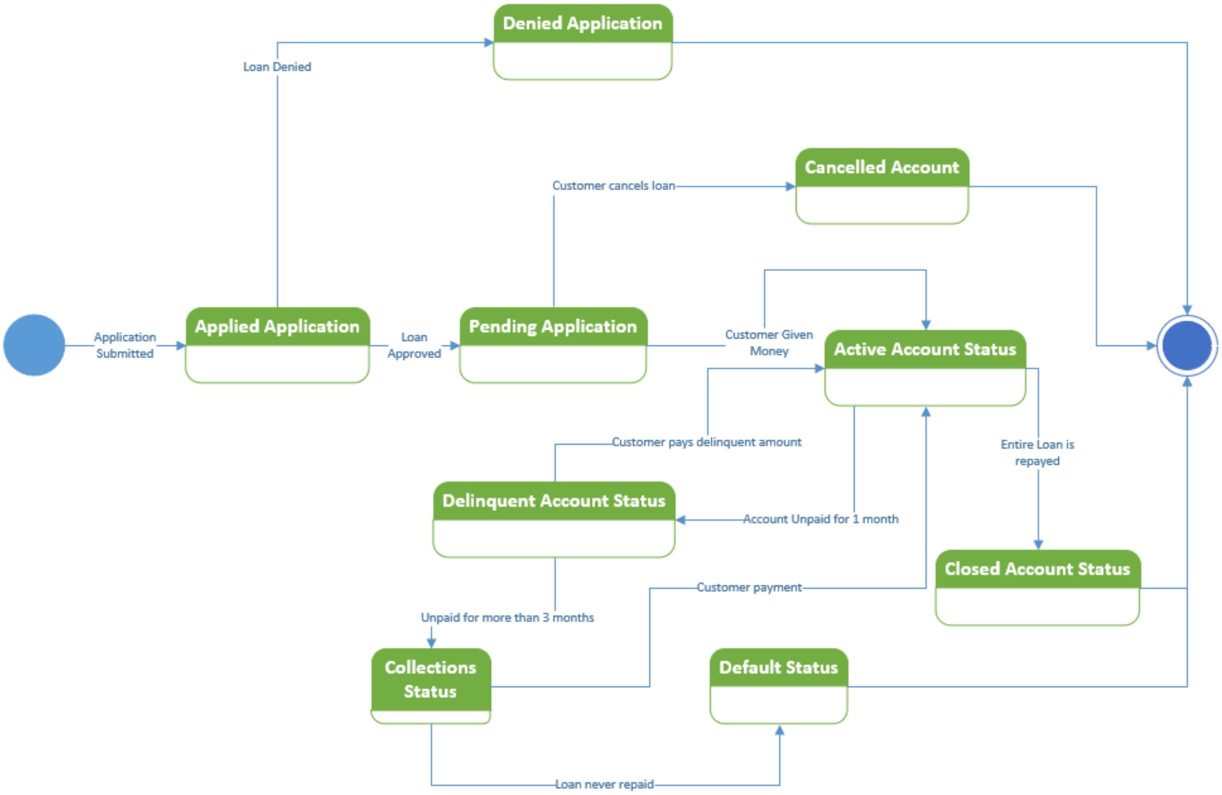
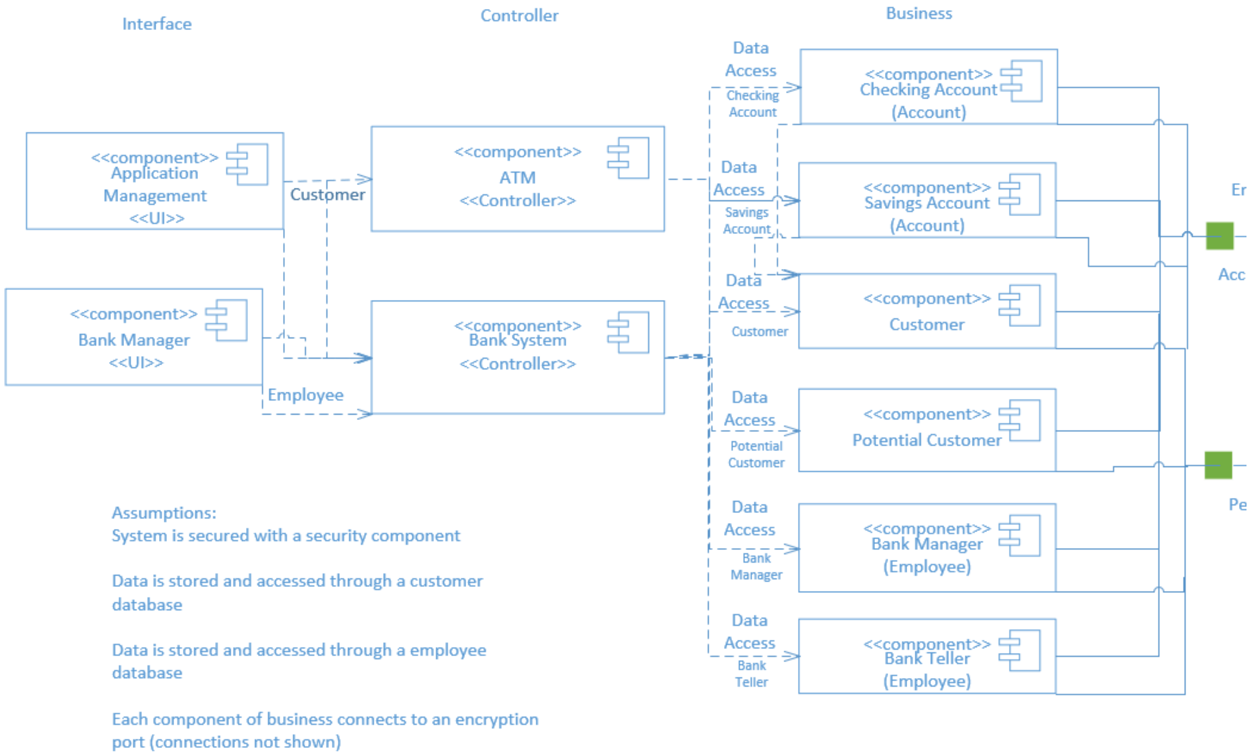


Figure 8: This is the State Machine Diagram that depicts the different states of a typical bank account.

Component Diagram


Interface
Controller
Business
<<component>> Application Management <<UI>>
<<component>> Bank Manager <<UI>>
Customer
Employee
<<component>> ATM <<Controller>>
<<component>> Bank System <<Controller>>
Data Access
Checking Account
Savings Account
Customer
Potential Customer
Bank Manager
Bank Teller
<<component>> Checking Account (Account)
<<component>> Savings Account (Account)
<<component>> Customer
<<component>> Potential Customer
<<component>> Bank Manager (Employee)
<<component>> Bank Teller (Employee)
Er
Acc
Pe

Assumptions:
System is secured with a security component

Data is stored and accessed through a customer database

Data is stored and accessed through a employee database

Each component of business connects to an encryption port (connections not shown)

Figure 9A: This is Part A of the Boiler IT Banking Services System Component Diagram.

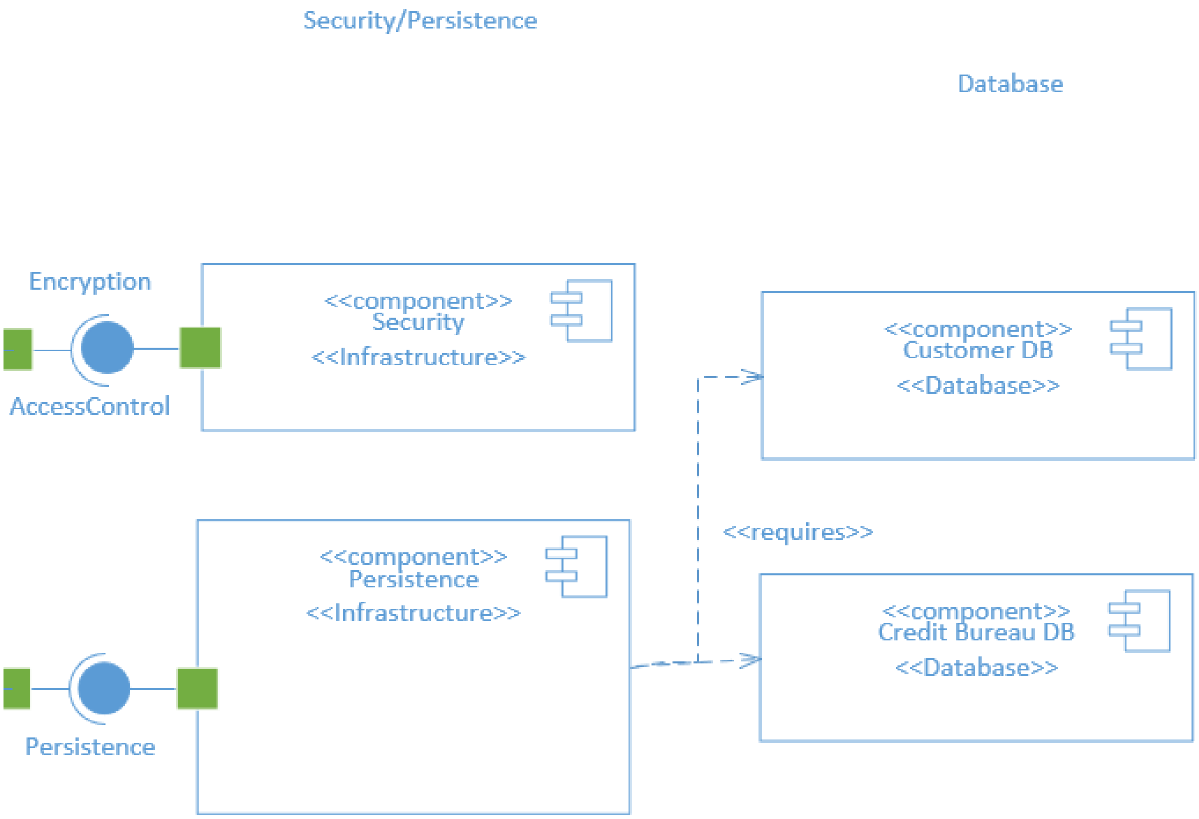


Figure 9B: This is Part B of the Boiler IT Banking Services System Component Diagram.

Deployment Diagram

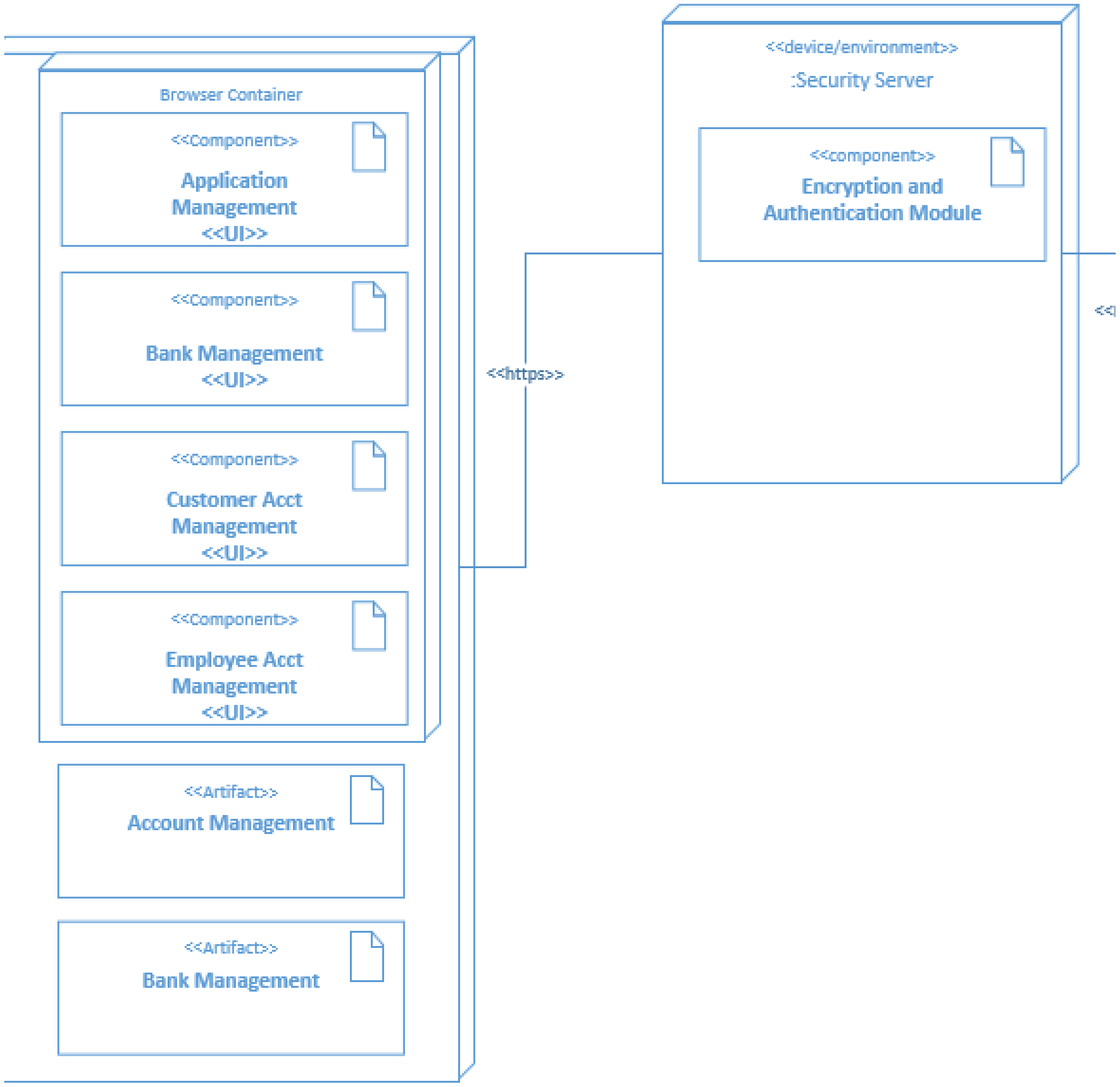


Figure 10A: This is Part A of the Boiler IT Banking Services Deployment Diagram.

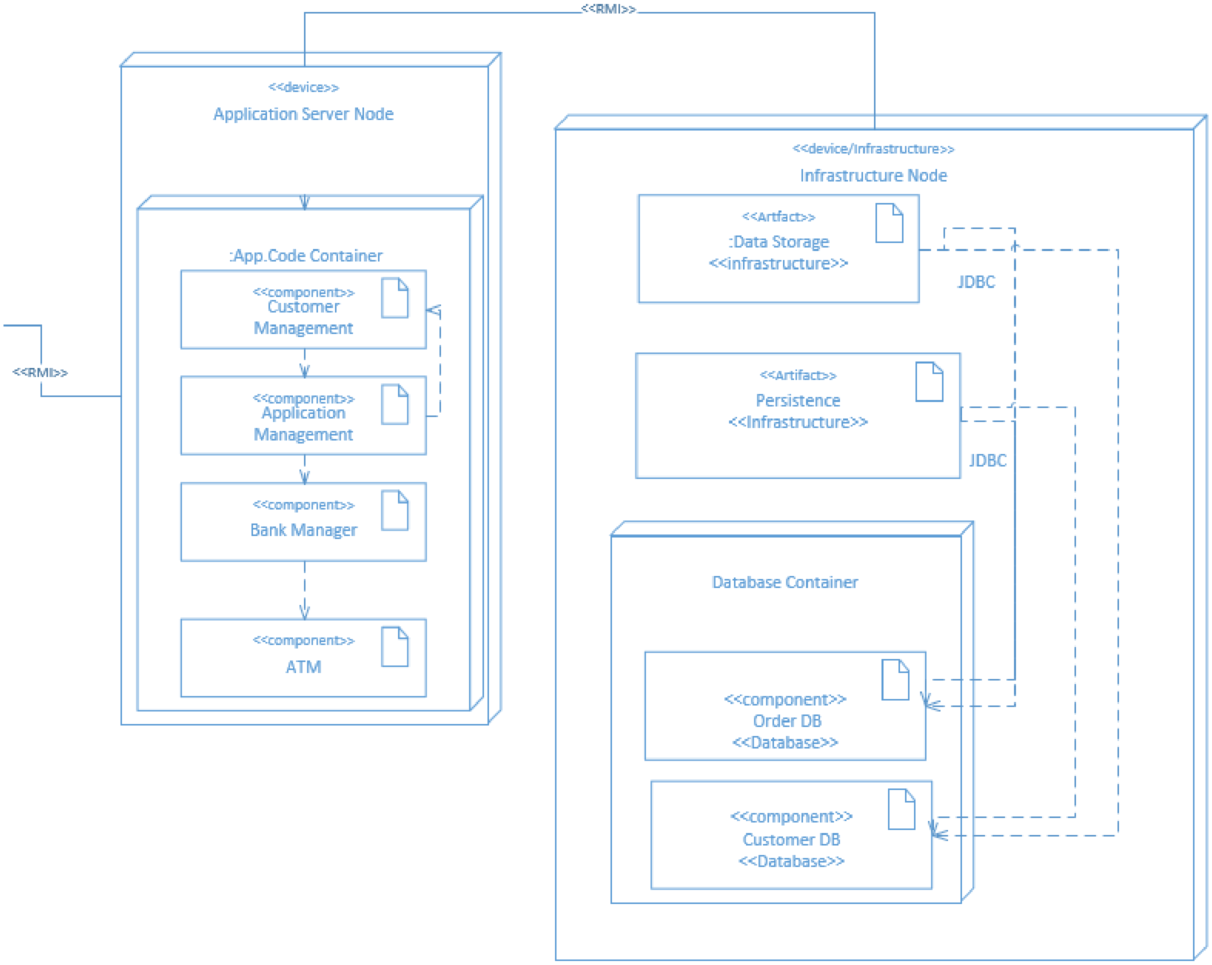


Figure 10B: This is Part B of the Boiler IT Banking Services Deployment Diagram.

Sequence Diagram

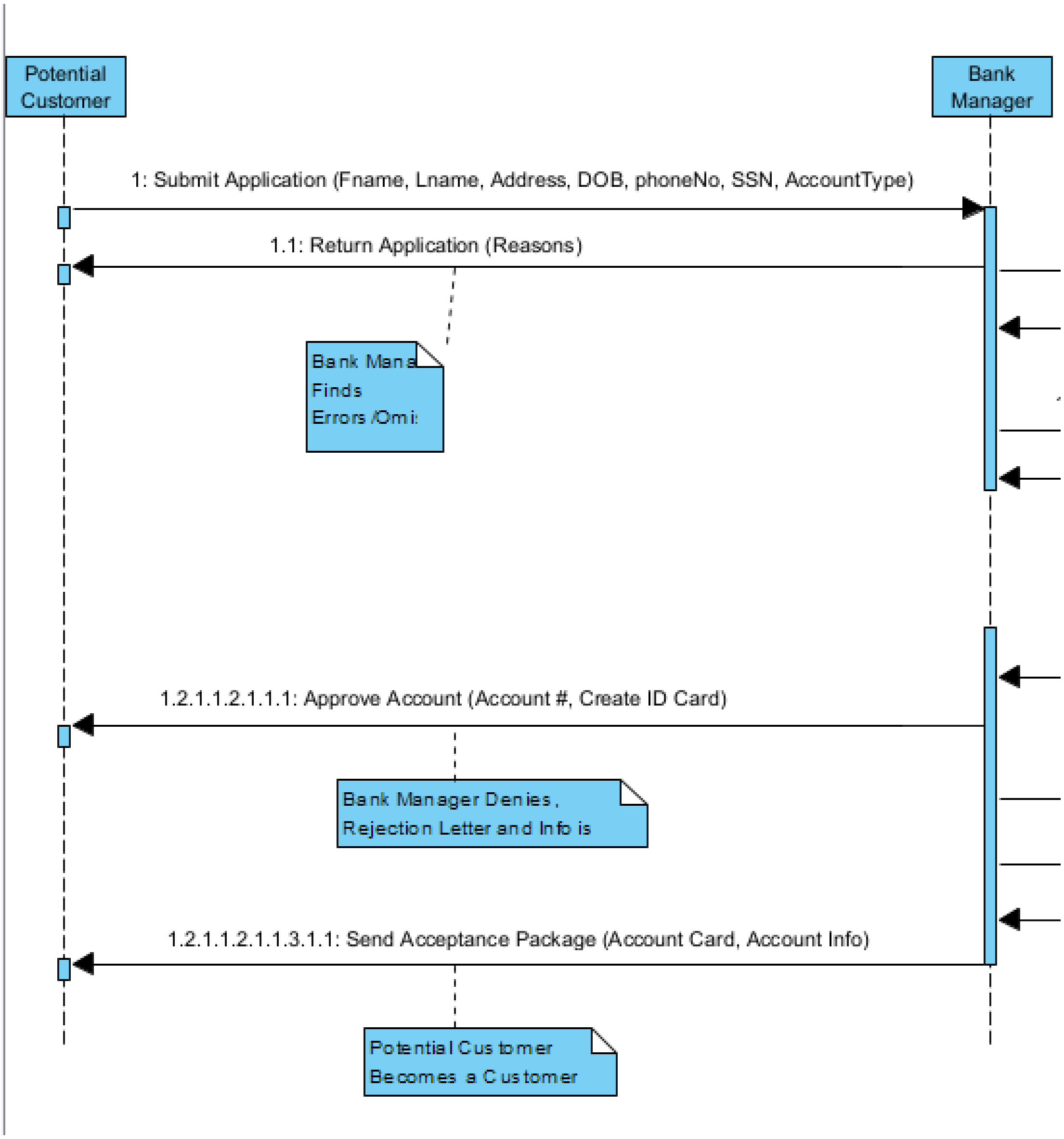

Potential Customer
Bank Manager
1: Submit Application (Fname, Lname, Address, DOB, phoneNo, SSN, AccountType)
1.1: Return Application (Reasons)
Bank Mana
Finds
Errors /Omi:
1.2.1.1.2.1.1.1: Approve Account (Account #, Create ID Card)
Bank Manager Denies,
Rejection Letter and Info is
1.2.1.1.2.1.1.3.1.1: Send Acceptance Package (Account Card, Account Info)
Potential Customer
Becomes a Customer

Figure 11A: This is Part A of the Sequence Diagram that depicts the events related to the Potential Customer.

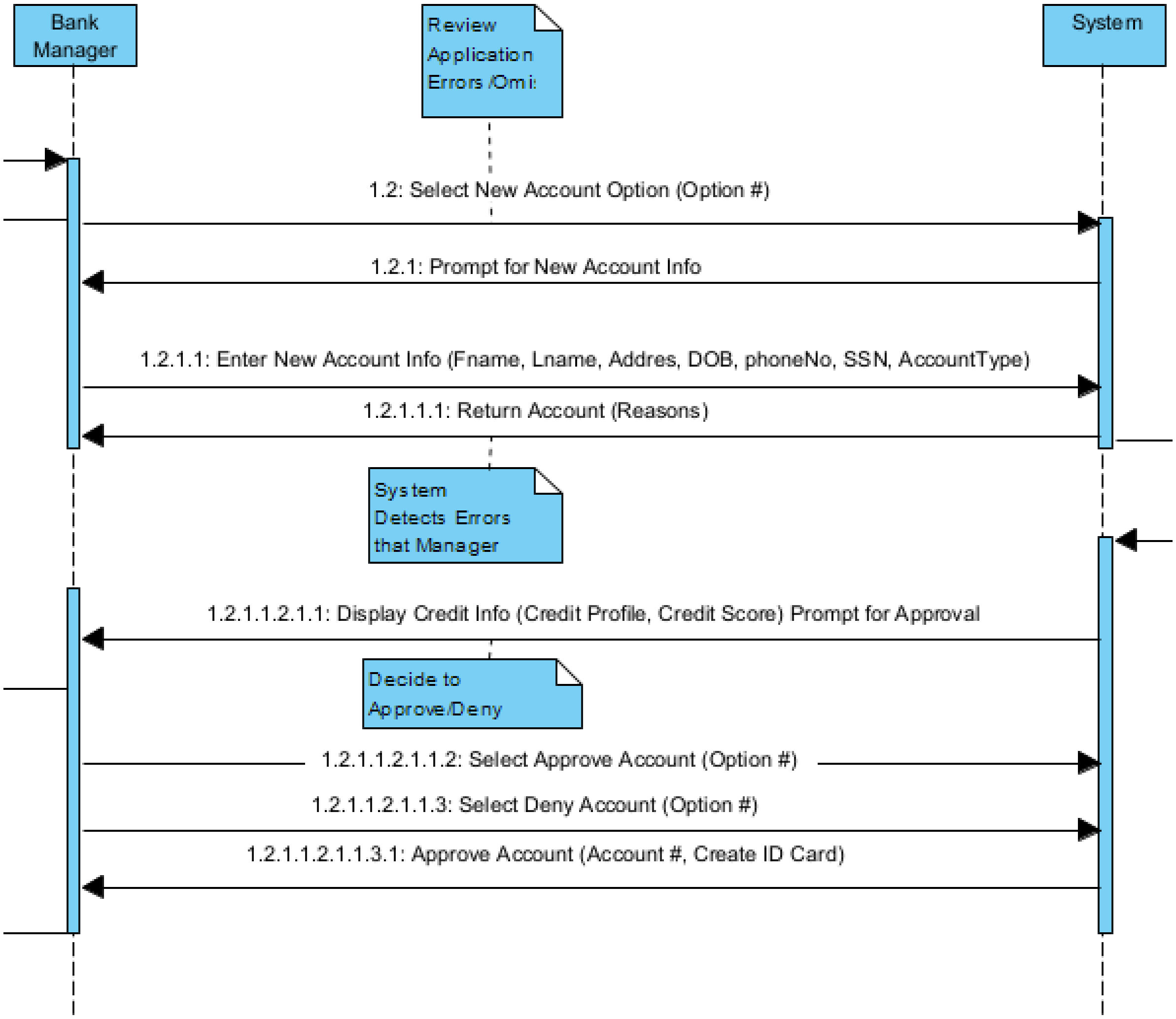

Bank Manager
Review Application Errors /Omi:
System
1.2: Select New Account Option (Option #)
1.2.1: Prompt for New Account Info
1.2.1.1: Enter New Account Info (Fname, Lname, Addres, DOB, phoneNo, SSN, AccountType)
1.2.1.1.1: Return Account (Reasons)
System Detects Errors that Manager
1.2.1.1.2.1.1: Display Credit Info (Credit Profile, Credit Score) Prompt for Approval
Decide to Approve/Deny
1.2.1.1.2.1.1.2: Select Approve Account (Option #)
1.2.1.1.2.1.1.3: Select Deny Account (Option #)
1.2.1.1.2.1.1.3.1: Approve Account (Account #, Create ID Card)

Figure 11B: This is Part B of the Sequence Diagram that depicts the events that are related to the Bank Manager.

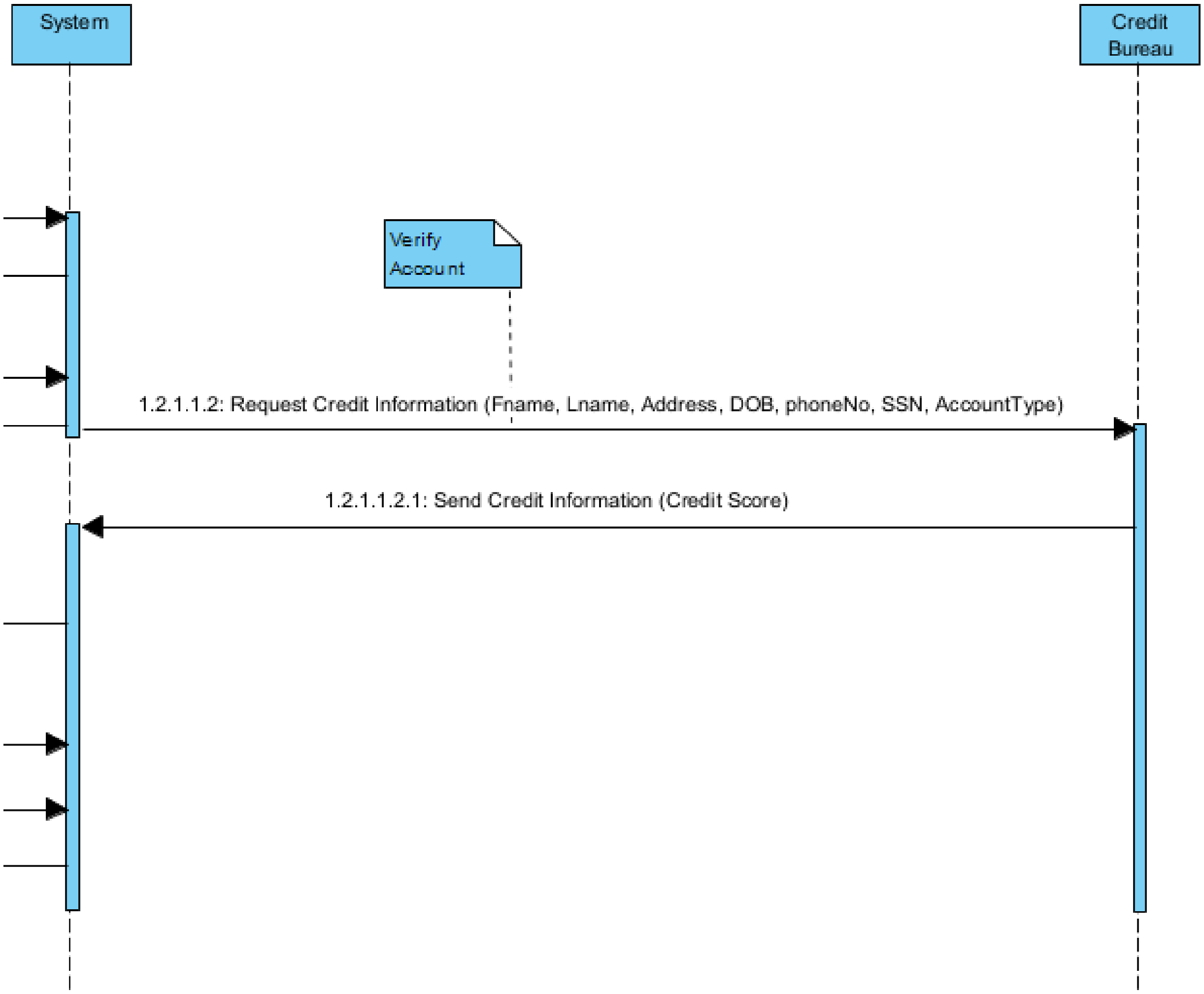

System
Credit Bureau
Verify Account
1.2.1.1.2: Request Credit Information (Fname, Lname, Address, DOB, phoneNo, SSN, AccountType)
1.2.1.1.2.1: Send Credit Information (Credit Score)

Figure 11C: This is Part C of the Sequence Diagram that depicts the events that are related to the Credit Bureau