Czech University of Life Sciences Prague
Faculty of Economics and Management
Department of Information Engineering

Diploma Thesis

Design of E-Banking Information system in UML

Urmina Hossain Depy

CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

Faculty of Economics and Management

DIPLOMA THESIS ASSIGNMENT

Urmina Hossain Depy, BSc

Systems Engineering and Informatics
Informatics

Thesis title

Design of E-Banking information system in UML

Objectives of thesis

The main aim of this thesis is to design a particular information system in UML to develop web applications for E-Banking system.
Now -a -days E-Banking system has became so popular and after this pandemic has started this is safe and handy. So, at the very beginning We will discuss the theoretical part about information system to get basic details and different types on UML modeling uses along with various system design concepts. In next phase is building UML model for e-banking system that will be use a typical area for information system and how a company work on that. Lastly, a general analysis will be carried out according to UML.

Methodology

In this thesis our main focus to find out how e-banking system works and the uses of UML which includes the current development modeling and understandable UML model for e-banking system by formulation the necessary requirements. We will discuss about the problems of e-banking how admin user can search and view details as well as add or update services and also users will be able to search of debit, credit so on. Examine the relationship between the dimensions of E-banking service quality and customer satisfaction.

To reach out our main goal we will build Class diagram, Use case diagram, Sequence diagram, Collaboration diagram, State chart diagram, Activity diagram, Component diagram, Deployment diagram for our E-banking information system. Through this client can easily reach their desire services.

E-banking has become one of the essential banking services that can, if properly implemented, increase customer satisfaction and give banks a competitive advantage. Overall, we will see how UML can work for e-banking system and some tools, procedure and materials. We can survey anonymously through some participants to see the respond about UML design for e-banking system. To evaluate and justify our methodological we see the limitations or weakness in the approach.

The proposed extent of the thesis

60-80

Keywords

100-200

Recommended information sources

Arlow, J. and Neustadt, I., 2005. UML 2 and the Unified Process: Practical Object• Oriented Analysis and Design, (2nd ed.). Boston, MA: Addison Wesley

Hamilton, K. and Miles, R., 2006. Learning UML 2.0, Boston, MA: O'Reilly

OMG, 2017. OMG Unified Modeling Language (OMG UML): Infrastructure specification, Version 2.5.1, formal/2017-12-05, s.l.: Object Management Group.

Rumpe, B., 2016. Modeling with UML: Language, Concepts and Methods, Cham, Switzerland: Springer.

Expected date of thesis defence

2021/22 SS – FEM

The Diploma Thesis Supervisor

Ing. Jakub Konopásek, Ph.D.

Supervising department

Department of Information Engineering

Electronic approval: 8. 3. 2022

Ing. Martin Pelikán, Ph.D.

Head of department

Electronic approval: 24. 11. 2022

doc. Ing. Tomáš Šubrt, Ph.D.

Dean

Prague on 30. 03. 2023

Declaration

I declare that I have worked on my diploma thesis titled “Information System Design in UML for E-Banking system” by myself and I have used only the sources mentioned at the end of the thesis.

As the author of the diploma thesis,

I declare that the thesis does not break the copyrights of any third person.

In Prague in March of 2023

Signature_____

Urmina Hossain Depy

Design of E-Banking Information system in UML

Návrh informačního systému elektronického bankovnictví v UML

Abstraktní:

Informační systém pro-online bankovnictví Podrobný plán, který popisuje architekturu a vlastnosti systému digitálního bankovnictví, se nazývá návrh UML. Komponenty systému, jejich vztahy a tok informací uvnitř systému jsou vizuálně reprezentovány návrhem pomocí Unified Modeling Language (UML).

Cílem informačního systému elektronického bankovnictví je poskytnout uživatelům bezproblémové online bankovnictví. Součástí systému jsou moduly pro autentizaci klienta, správu účtu, zpracování transakcí a reporting. Pro zabezpečení citlivých klientských informací zahrnuje návrh také bezpečnostní opatření včetně ověřování uživatelů, šifrování dat a řízení přístupu.

Diagramy případů užití, sekvenční diagramy, diagramy tříd a diagramy aktivit jsou jen některé z diagramů, které návrh UML používá k zobrazení architektury systému. Zúčastněné strany mohou rychle pochopit a posoudit návrh systému díky jasnému a stručnému zobrazení jeho fungování v těchto diagramech.

Celkově návrh UML pro informační systém elektronického bankovnictví nabízí kompletní a spolehlivý základ pro vytvoření bezpečného a uživatelsky přívětivého systému digitálního bankovnictví.

Abstract:

The information system for online banking A detailed blueprint that describes the architecture and features of a digital banking system is called a UML design. The components of the system, their relationships, and the information flow inside the system are all represented visually by the design using the Unified Modeling Language (UML).

The goal of the electronic banking information system is to give users a smooth online banking experience. Modules for client authentication, account administration, transaction processing, and reporting are all included in the system. To secure sensitive client information, the design also includes security measures including user authentication, data encryption, and access controls.

Use case diagrams, sequence diagrams, class diagrams, and activity diagrams are just a few of the diagrams that the UML design uses to show the architecture of the system. Stakeholders may quickly comprehend and assess the system's design thanks to these diagrams' clear and succinct portrayal of its operation.

Overall, the UML design for the e-banking information system offers a complete and reliable foundation for creating a safe and user-friendly digital banking system.

Keywords:

E-Banking, UML, OODBMS, Use-Case diagram, Class Diagram, Sequence diagram and State diagram.

Acknowledgment:

I would like to express my sincere gratitude to my thesis advisor, Ing. Jakub Konopásek, Ph.D, for his invaluable guidance, support, and encouragement throughout this research project. His expertise and constructive feedback have been instrumental in shaping my ideas and improving the quality of this thesis.

I am deeply grateful to my husband who continuously supported me throughout my full master journey. And also, to the most precious gift from God “My son URAIF SHAFI” who is an amazing kid I must say. I should mention how clam and gentle person he is, such a supportive kid. This joinery wouldn’t be possible to complete without his tremendous support. I would like to mention from my heart about my parents and siblings for their unwavering love, support, and understanding. Their sacrifices and encouragement have been the foundation of my academic success, and I dedicate this thesis to them.

I also want to thank my colleagues and friends at the Department of Information engineering. Their contributions and knowledge have enriched my research and made this journey much more enjoyable.

Finally, I want to express my appreciation to the participants who generously volunteered their time and effort to participate in this study. Their willingness to share their experiences and insights has been the driving force behind this research.

Thank you all for your support, encouragement, and inspiration.

Contents

List of Figures

➢ Introductory Part

1. INTRODUCTION

Technology is the first thing for any works nowadays and with the time being it's getting more higher in the priority list. In a simple word we can say "Technology is a part of our daily life/ Technology has become our life". If we can summarize our day-to-day life from Bed to Bed (morning to night), it's clearly visible that technology is in every steps. Developments of technology touched also the bank department, giving birth to a new product called online banking or e-banking, and day to day it's improving. Even we look 15 years back e-banking wasn't that much popular how it is nowadays. People no need to go to bank like before how they did. Everything is possible through online. What an online banking offers is an opportunity to perform different bank operations, where a customer can access his or her bank account via internet. And such operations we can perform from our personal laptop or even from our mobile phone that makes our life handier and saving our time. I must say that's a great blessing for us. Clients can check their current account, saving account, transfer money both manually and by QR code as well at any place. E-banking usage is becoming very common due to increase of daily usage of mobile which avail transfers. These days all the banks have e-banking system and they are offering variety of services while client make or open account with their desire bank. (Rumbaugh, and Jacobson, 1998)

UML (Unified modeling language) is used to describe a system, including its major actors, roles, actions, artifacts or classes, in order to better understand/edit/maintain/ document system information. Some UML tools help us to create code on programming language.

I choose to use the Unified Software Development Process approach for my thesis (hereinafter referred to as Unified Process). Because it uses the UML modelling language, I selected the methodology. This language captures all significant behavioral patterns. A broad software development technique tool is the Unified Process. The technique is built on a few fundamental principles: Use cases and risk drive development. In the development process, architecture is given the most priority. The development process is iterative. The

growth of a bigger unit can be broken down into several smaller, more manageable issues through interaction. The use cases that are textually defined in the system requirements and then examined are crucial to this thesis fragment. The analysis and design systems used in software development are the main procedures of this thesis.

1.1 An introduction to online banking

Banking organizations make modifications in response to the quickening growth of information technology.

By implementing electronic banking, they are modernizing their existing client interactions. It modifies the job instructions. They fix technological apparatus. They merge database systems. They teach employees who work with clients. They utilize tools for customer relationship management.

These modifications force them to modify their information strategies.

Previously, the bank anticipated that its investments in the information technology would improve corporate performance.

This, together with the bank's initiatives to modify the quality and manner of delivering financial services, necessitates a conceptual shift in the information systems industry, with a focus on the anticipated return on investment and the establishment of a long-term competitive advantage.

The flexibility on using services 365 days a year, seven days a week, and twenty-four hours a day is the primary benefit of electronic banking services. The consumer may conveniently do routine banking activities nowadays through electronic banking without being compelled to visit the bank. The client can directly arrange for the bank to supply its services during the purchase of the items. While providing goods directly at the point of purchase may seem straightforward, it is actually a very complex procedure, supported by sophisticated information systems with high levels of performance, reliability, and stability that are

specialized to meet the needs of banks. The system's time-dependent actions are described by the dynamic view. (Ambler, 2004)

In this thesis, we will explore the use of UML design in E- Banking system. We will learn about the different aspects of UML design and how it can be used to improve the efficiency of E- Banking system. It is a modeling language that is used to create visual representations of software systems. UML allows developers to create diagrams that show the structure of a system and its components. UML diagrams can be used to represent many different types of information, including: • Use case diagrams: Show the interactions between users and a system • Class diagrams: Show the relationships between classes and objects • Activity diagrams: Show the flow of activities in a system • State machine diagrams: Show the state transitions in a system.

➢ Objectives of Thesis:

The aim of this thesis is to design a particular information system in UML to develop web applications. We will discuss the theoretical part about information system to get basic details and different types on UML modeling uses along with various system design concepts. In next phase is building UML model for e-banking system that will be use a typical area for information system and how a company work on that. Lastly, a general analysis will be carried out according to UML.

➢ Methodology:

In this thesis our main focus to find out how e-banking system works and the uses of UML which includes the current development modeling and understandable UML model for e-banking system by formulation the necessary requirements. We will discuss about the problems of e-banking how admin user can search and view details as well as add or update services and also users will be able to search of debit, credit so on. Examine the relationship between the dimensions of E-banking service quality and customer satisfaction. E-banking has become one of the essential banking services that can, if properly implemented, increase customer satisfaction and give banks a competitive advantage. Overall, we will see how UML can work for e-banking system and some tools, procedure and materials. We can survey anonymously through some participants to see the respond about UML design for e-banking system. To evaluate and justify our methodological we see the limitations or weakness in the approach.

❖ Literature review

2. The basics of UML

UML is a standard way of visualizing the design of a system. It is often used in software engineering to help developers create and understand complex systems. In this article, we will discuss the basics of UML and how it can be used in designing an E-Banking system.

UML stands for Unified Modeling Language. It is a graphical language that was created to help developers visualize the design of complex systems. The UML standard is managed by the Object Management Group (OMG). (OMG, 2017)

The UML notation consists of a set of symbols that can be used to create diagrams that represent the different parts of a system and their relationships. The most common types of diagrams are class diagrams, component diagrams, and deployment diagrams. (Conallen, 2002)

Class diagrams are used to represent the static structure of a system. They show the relationships between classes and interfaces. Component diagrams are used to represent the Dynamic structure Of a System and Deployment Diagrams Are Used to Represent the Physical Structure of a System.
UML can be used to design any kind of system, but it is particularly well suited for designing E- Banking Systems. Because E- Banking Systems Are Complex Systems That Involve Many Different Parts.
For example, an E- Banking system would need to take into account customer accounts, transactions, security, and compliance with regulations. (Hamilton and Miles,2006)

Using UML helps developers create a mental model of how the E-Banking system will work.

The ability to use services 365 days a year, seven days a week, and twenty-four hours a day is the primary benefit of electronic banking services. Today, a client may conveniently do routine financial activities using electronic banking without having to physically visit a bank. The client can directly arrange for the bank to supply its services during the purchase of the

items. While providing goods directly at the point of purchase may seem straightforward, it is actually a very complex procedure, supported by sophisticated information systems with high levels of performance, reliability, and stability that are specialized to meet the needs of banks. The system's time-dependent actions are characterized by the dynamic view.

3. History of UML

A strong new combination of concepts for structuring computer programmers, including instantiation of abstract data types, inheritance, and polymorphism, were presented with the introduction of Simula, the first object-oriented programming language, in the late 1950s (Cook, 2012, p. 471). Alongside this new concept of object-oriented languages, techniques for creating software in an object-oriented manner also began to appear. Over time, these techniques came to be known as modelling languages. More than fifty different modelling languages, each with their own syntax, structure, and notation, were available by the late 1980s.

For instance, failure rates for system development and implementation remained persistently high. Cost and time overruns continued to be the rule rather than the exception (Erickson & Siau, 2013, p. 296). Alarmingly high project failure rates resulted from a lack of communication and technical understanding, as well as the fact that the majority of the languages available were unable to meet the demands required of them. This was due to the fact that it was humanly impossible in this type of environment for all system analysts and other relevant personnel to be trained in all methods.

UML 1.0 initially appeared to satisfy all requirements and to be a useful modelling language when it was recognized as a standard. But since then, a number of changes have been made to the notation in attempt to improve it and address a number of issues. Examples of issues that were fixed between UML 1.1 and UML 1.3 include the absence of particular modelers, the lack of integration between certain model types, and the inconsistent naming and organization of some of the standard parts

4. How UML is used in E-Banking

Banking systems have been using UML for a long time. It is used extensively in modeling the business processes and workflows in a banking system. The use of UML helps to document, understand and improve the efficiency of the banking system.

UML is also used in developing the user interface (UI) for an e-banking system. The UI must be easy to use and understand by the customer. It should be able to handle all the transactions that a customer does on a daily basis. The use of UML helps in designing such an interface.

In addition, UML is also used in developing the security architecture of an e-banking system. The security architecture ensures that all the transactions made by the customers are safe and secure. UML helps in designing such a security architecture.

It is crucial to stress that most systems belong to more than one system type or as a mix rather than precisely fitting into one of these categories. Today's information systems, for instance, frequently have both dispersed and real-time needs. All of these system types may be modelled using UML. (Arlow, and Neustadt, 2005)

4.1. Advantages of using UML in E-Banking

UML provides many benefits for designers of e-banking systems. First, it helps to create a clear and concise representation of the system. This is especially important for complex systems with many different types of interactions. Second, UML can be used to automatically generate code from the design, which can save a lot of time and effort in development. Third, UML can help to ensure that the system is consistent and conforms to standards. Finally, UML tools can provide valuable insights into the system that can help to improve its design.

4.2. Disadvantages of using UML in E-Banking

There are a few disadvantages to using UML in e-banking systems. First, it can be difficult to understand the diagrams without prior knowledge of the Unified Modeling Language. Second, because UML is such a comprehensive modeling language, it can be challenging to create diagrams that accurately represent all aspects of an e-banking system. Finally, UML diagrams can be time-consuming to create and update, which can delay the development process. (Zarei, S. 2011)

5. Online banking

In this chapter we will present the basic history of online banking, the primary type and its progress. Apart from that we will arise the economic and social effect due to Online Banking adoption and compare with different countries by their difference and alike. Before we start designing any system, we need to understand how it works, the algorithm will help to understand this. Highlighting the main actions and the links between them, we get a detailed description of the work.

In this chapter we will present the basic history of online banking, the primary type and its progress. Apart from that we will arise the economic and social effect due to Online Banking adoption and compare with different countries by their difference and alike. Before we start designing any system, we need to understand how it works, the algorithm will help to understand this. Highlighting the main actions and the links between them, we get a detailed description of the work

5.1. UML can assist you with

Before you start writing code, create your system using a sketch or other design method. Reverse engineering: To better understand existing code, create images of it. Typically,

UML models are made up of many diagrams. Some UML diagrams display a software system's static structure. A class diagram, for instance, displays a grouping of classes, types, and interfaces along with their connections. A class diagram can display each class's operations, but it cannot represent how these operations behave in practice. A model's dynamics may be explained using behavioral diagrams like state charts and sequence diagrams. (Kurnia, Peng, & Liu2010).

These illustrations can show the whole lifecycle of an object or what occurs when an action is invoked. A full UML model will always include structural and behavioral diagrams that describe the system's static and dynamic elements. However, the study of behavioral diagrams is the main topic of this thesis.

❖ Practical Part

6. Aspects of how the cards are played

There are different types of cards that we can use for bank now a days. Some are as follow-

1. Electronic card
2. Debit card
3. Credit card
4. Master card
5. Visa card
6. Internet card
7. Chip card

6.1. Electronic card:

In our world, electronic cards are the most common kind. These only apply to transactions that are electronically validated at the card center, such as withdrawals from ATMs and payments made at businesses that use electronic payment terminals. The benefit of this

Low cost, affordable fees for restricting a stolen or misplaced card, and practically no chance of a blocked card being misused characterize these types of cards. The main drawback is their restricted usage; for instance, in the Czech Republic, you may only use them to make purchases at half the businesses that take credit cards.

6.2. Debit card:

Debit card is best for people trying to budget or prevent debt accumulation. It's linked to a checking account. May have to pay fees when using an ATM out of network or overdraft. Liability of losing funds is tied to how quickly you act to report stolen card. Funds that are

pulled directly from bank account can be used anywhere credit cards are permitted. It has higher cash withdrawal limit also easy to avail.

6.3. Credit card:

Credit card means line of credit offering borrowed funds that must be repaid. It helps build credit history. Not liable for more than $50 if credit card is lost or stolen due to fair credit billing ACT. May have to pay fees for foreign transactions, late fees and cash advances. A debt instrument for financial transactions instead of cash. It has post purchase EMI conversion as well as complimentary insurance cover for fraud transactions.

6.4. Master card:

There are four businesses that control the electronic payments market. Most card payments in the globe are processed by Visa, Mastercard, American Express, and Discover.

Due to the fact that neither Visa nor Mastercard is a business that extends credit or issues cards, they each have unique products. This indicates that every Visa and Mastercard payment card is issued as a result of a co-branding arrangement.

Although neither company extends credit nor issues cards, the two do work together to offer the widest range of goods, including alternatives for credit, debit, and prepaid cards.

6.5. Visa card:

The two biggest credit card processing networks in the world are Visa and Mastercard. In contrast to Discover and American Express, Visa and Mastercard issue cards through their member banking institutions.

Visa and Mastercard credit and debit cards are issued by member banks and credit unions directly to their clients and, frequently, through co-branded credit card alliances with airlines, lodging providers, and big-box shops.

6.6. Internet cards

Internet cards are virtual payment cards, meaning that only their number and validity are produced using CVV2/CVC2 codes. They don't actually exist in plastic like other payment cards do. The Internet Card can only be used to buy products and services online and through so-called MO/TO orders (Mail Order/Telephone Order), and cannot be used to withdraw cash from an ATM or make purchases in physical places. Because a prospective attacker would need to know all card information, including the card number, issuance date, and passcode lock CVV or CVC, it has the advantage of increased security and a decreased risk of misuse.

6.7. Chip cards:

A chip card has a microprocessor that controls access to the information contained on it, the information required to validate the client's personal code, on the front of the card. The most crucial (in terms of security) applications employ chip processor cards, which boost fraud protection.

Chip cards, commonly referred to as EMV (Europay, Mastercard, and Visa) cards, are debit and credit cards with embedded computer chips that provide unique transaction codes.

This added layer of security makes chip cards more difficult to counterfeit or clone, and has replaced the traditional magnetic stripe system in many countries.

The user normally needs a card reader or a mobile device with Near Field Communication (NFC) technology to connect with the chip in order to utilize a chip card for online banking. The user is required to input a personal identification number (PIN) or another authentication mechanism after inserting the card into the scanner or holding the card close to the mobile device. Nonetheless, it is crucial for users to keep their cards and PINs secure and avoid disclosing their login information to third parties. (Shy and Tarkka, 2002)

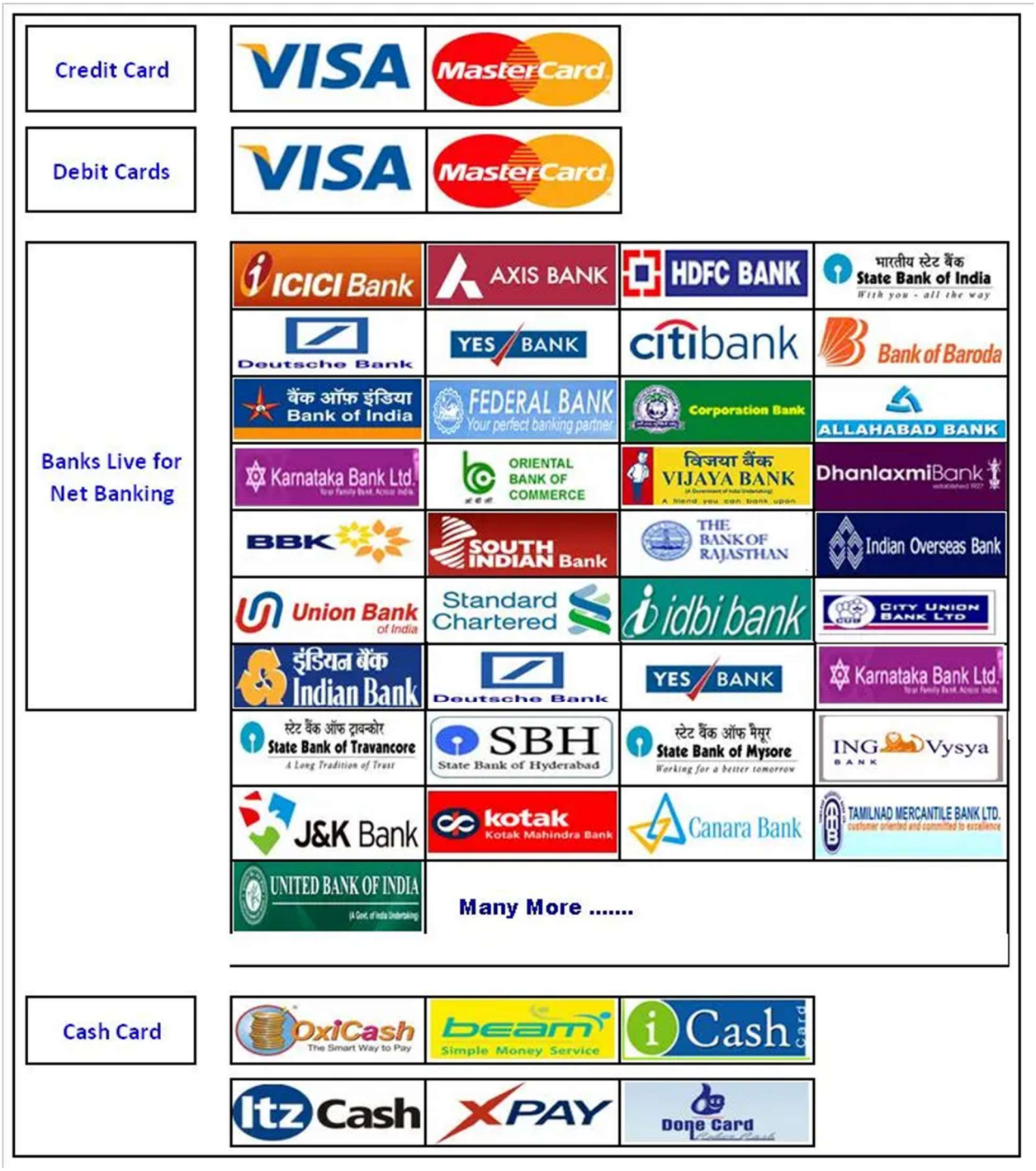

Figure: 1 Aspects of card (WordPress.com : Image. (n.d.)

7. Overview of a few electronic banking

7.1. Telephone Banking

The first company to commercially provide telephone banking was Girobank in the UK, which launched a specialized telephone banking service in 1984. Telephone banking was initially made available in the 1980s. The first wave of direct banks employed telephone banking extensively during the 1980s and the early 1990s.

7.2. GSM banking

GSM banking refers to the use of mobile phones or devices for banking transactions such as checking account balances, transferring funds, paying bills and buying goods and services. This service is provided by financial institutions and telecommunication companies that offer mobile banking applications or USSD codes that customers can use to access their bank accounts.

Mobile banking has gained popularity in recent years as a result of its accessibility and ease. Customers may conduct transactions whenever they want, anywhere they have access to the internet or a physical bank. Moreover, it offers a more secure method of banking because customers may authenticate themselves and authorize transactions using their own mobile devices.

Overall, GSM banking has revolutionized the banking industry by providing a faster, easier, and more secure way of conducting financial transactions using mobile devices.

7.3. Internet banking

Financial services are delivered via the internet through internet banking. Customers of the bank can use this system to manage their accounts, pay bills, and send money internationally

through the internet. A visual notation language used to design software systems is called UML, or Unified Modeling Language.

Systems for online banking can gain from the use of UML. To better understand the system architecture, functions, and behaviors of the online banking system, UML may be employed. It is simpler to explain the system design to stakeholders by using UML, which offers a standard method of drawing the system components and their interactions. (Bahl, 2012)

7.4. WAP banking

WAP banking refers to the mobile banking services that can be accessed through a Wireless Application protocol (WAP)- enabled mobile device, such as a smartphone or tablet. WAP is a technology that enables mobile devices to access and navigate the via a wireless network.

With WAP banking, users can perform various banking financial transactions, including checking account balances, transferring funds between accounts, paying bills and applying for loans, among others. Mobile banking apps often support WAP banking services to provide customers with flexibility in accessing their accounts anytime and anywhere they have an internet connection.

To use WAP banking services, users typically need to register for mobile banking with their bank and download the mobile banking app compatible with their device. They can then log in to the app and access the WAP banking services provided by their bank. WAP banking offers convenience and accessibility to users who prefer to carry out banking transactions on the go, without having to visit a physical bank branch.

7.5. Home banking

Home banking refers to a type of online banking that allows customers to perform various banking activities from the comfort of their homes. This service allows users to access banking services using their computer or mobile device and the internet.

Home banking is a sort of internet banking that enables users to conduct different banking tasks from the convenience of their homes. With the help of this service, customers may access banking services online using a computer, a mobile device, or both.

Users may perform various financial chores from the comfort of their homes through home banking, a type of internet banking. Customers can use a computer, a mobile device, or both to access banking services online with the use of this service.

In general, home banking is a practical and adaptable choice for clients who wish to manage their funds from any location. Customers may use it to obtain account information and carry out transactions without going to a real location.

7.6. PayPal

These are only a handful of the components that may be seen in online banking systems often. A UML component diagram shows how various components work together to create the overall system architecture.

In terms of UML (Unified Modeling Language), PayPal can be modeled with a combination of class diagrams, use case diagrams, and sequence diagrams. The class diagram would represent the various classes that make up the PayPal system, such as the User class or the Transaction class. The many activities that users can do, such as making a payment or asking for a refund, would be depicted in the use case diagram. Finally, the sequence diagram would show the steps involved in a particular transaction, such as how information is submitted and verified.

In general, PayPal is a well-liked and practical online payment option that supports user-to-user financial transactions that are safe and secure.

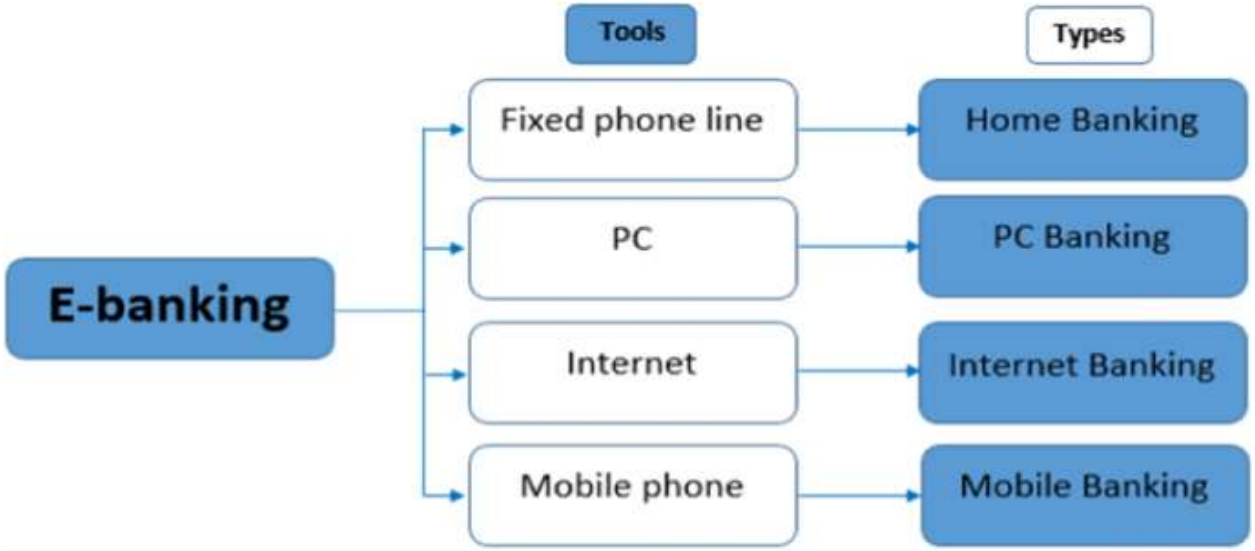

Figure 2: Overview of electronic banking (Bahl, 2012)

8. Types of UML diagram for E-banking:

1. Class diagram
2. Use case diagram
3. Collaboration diagram
4. Activity diagram
5. Statechart diagram
6. Deployment diagram.

8.1 Class diagram

In UML, class diagrams are one of six types of structural diagram. Class diagrams are fundamental to the object modeling process and model the static structure of a system. Depending on the complexity of a system, you can use a single class diagram to model an entire system, or you can use several class diagrams to model the components of a system. In bank management system class diagram works to

describes a system by visualizing the different types of objects within a system and the kinds of static relationships that exist among them. It describes the structure of a Banking Management System classes, their attributes, operations (or methods), and the relationships among objects. The main classes of the Banking management system are Customer, Employees, Accounts, Fixed Deposit, Saving Account, Current Account. (Rumpe, 2016)

You can use class diagrams to visualize, specify, and document structural features in your models. During the implementation phase of a software development cycle, you can use class diagrams to convert your models into code and to convert your code into models.
For example, during the analysis and design phases of the development cycle, you can create class diagrams to perform the following functions

- Capture and define the structure of classes and other classifiers
- Define relationships between classes and classifiers
- Illustrate the structure of a model by using attributes, operations, and signals
- Show the common classifier roles and responsibilities that define the behavior of the system
- Show the implementation classes in a package
- Show the structure and behavior of one or more classes
- Show an inheritance hierarchy among classes and classifiers
- Show the workers and entities as business object models

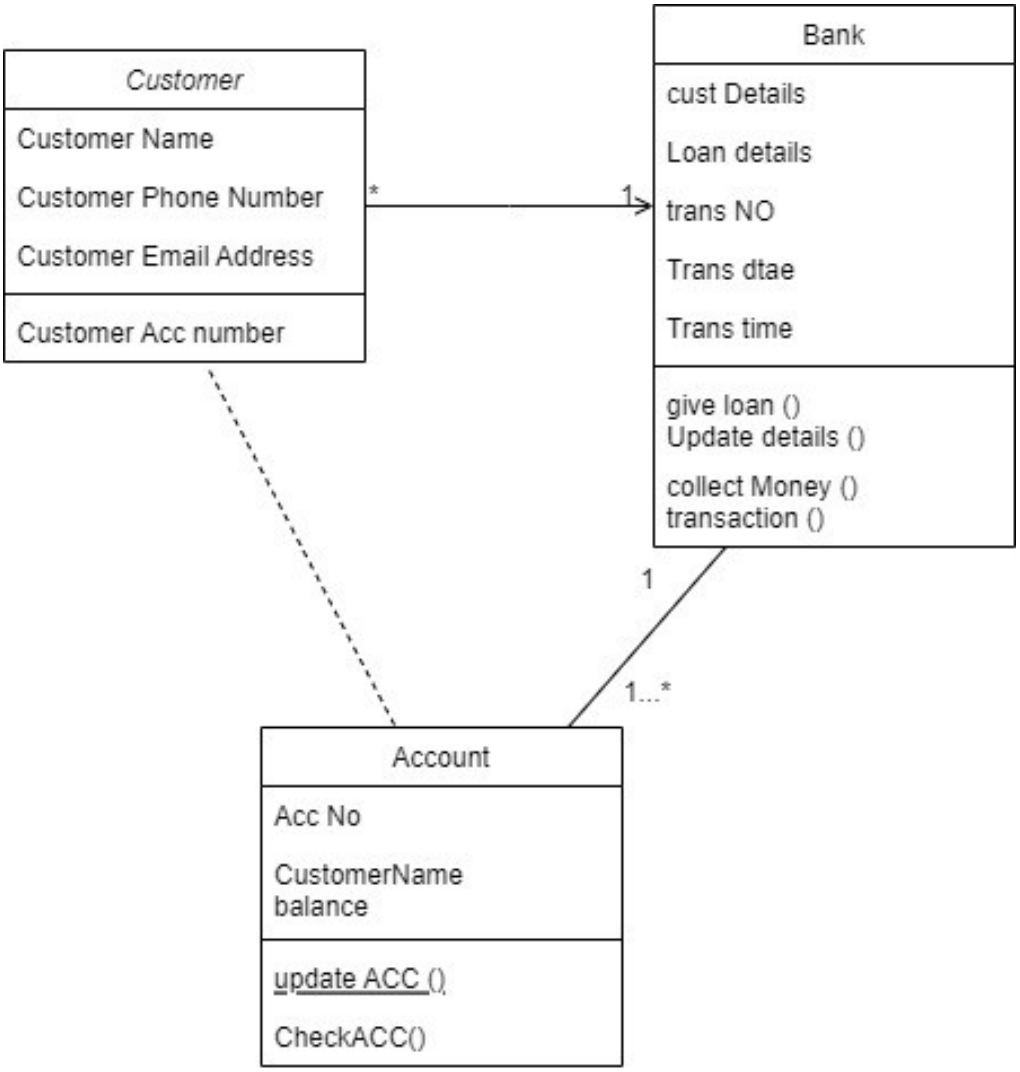

Figure 3: class diagram (source- author)

8.1.1. Classes of E-Banking system class diagram:

- Customer class: manage all the operations of customer
- Employees class: manage all the operations of Employees
- Accounts Class: Manage all the operations of accounts
- Fixed Deposit class: Manage all the operations of fixed deposit
- Saving Account Class: Manage all the operations of saving account
- Current Account class: Manage all the operations of current account

8.1.2. Classes and their attributes of E-Banking system

Customer Attributes: customer_id, customer_name, customer_mobile, customer_email, customer_username, customer_password, customer_address

Employees Attributes: employee_id, employee_name, employee_mobile, employee_email, employee_username, employee_password, employee_address

Account Attributes: account_id, account_customer_id, account_number, account_type, account_balance, account_description

Fixed Deposit Attributes: deposit_id, deposite_customer_id, deposit_amount, deposit_total,deposit_bank, deposite_type, deposit_description

Saving Account Attributes: saving_account_id, saving _account_customer_id, saving account_account_balance, saving_account_number, saving_account_description

Current account attributes: current_account_id, current_account_customer_id, current_account_balance, current_account_number, current_account_description.

8.1.3. Classes and their methods of E-banking system:

- Customer Methods: addCustomer(), editCustomer(), deleteCustomer(), updateCustomer(), saveCustomer(), searchCustomer()
- Employees Methods: addEmployees(), editEmployees(), deleteEmployees(), updateEmployees(), saveEmployees(), searchEmployees()
- Fixed Deposit Methods: addFixed Deposit(), editFixed deposit(), deleteFixed Deposit(), updateFixed Deposit(), saveFixed Deposit(), searchFixed Deposit()
- Saving Account Methods: addSaving Account(), editSaving Account(), deleteSaving Account(), updateSaving Account(), updateSaving Acoount(), updateSaving Account(), saveSaving Account(), searchSaving Account()

- Current Account Methods: addCurrent Account(), editCurrent Account(), deleteCurrent Account(), updateCurrect Account(), saveCurrent Account(), searchCurrent Account()

8.2. Use Case

Use cases describe the interactions that take place between actors and IT systems during the execution of business processes. A use case represents a part of the functionality of the IT system and enables the user (modeled as an actor) to access this functionality. This banking system use case diagram can help you: - Summarize the details of your system's users (also known as actors). - Organize a banking system's interactions. - Access the UML shape libraries. Open this template to view a detailed example of a banking system use case diagram that you can customize to your use case.
This example of UML class diagram models bank account system.
"A bank account is a financial account between a bank customer and a financial institution. A bank account can be a deposit account, a credit card, or any other type of account offered by a financial institution. The financial transactions which have occurred within a given period of time on a bank account are reported to the customer on a bank statement and the balance of the account at any point in time is the financial position of the customer with the institution. a fund that a customer has entrusted to a bank and from which the customer can make withdrawals. (GeeksforGeeks. 2021)

"Banks offer many different channels to access their banking and other services:

(1) Automated Teller Machines.
(2) A branch is a retail location.
(3) Call center.
(4) Mail: most banks accept cheque deposits via mail and use mail to communicate to their customers, by sending out statements.
(5) Mobile banking is a method of using one's mobile phone to conduct banking transactions.
(6) Online banking is a term used for performing multiple transactions, payments etc. over the Internet.

(7) Relationship Managers, mostly for private banking or business banking, often visiting customers at their homes or businesses.
(8) Telephone banking is a service which allows its customers to conduct transactions over the telephone with automated attendant or when requested with telephone operator.
(9) Video banking is a term used for performing banking transactions or professional banking consultations via a remote video and audio connection. Video banking can be performed via purpose-built banking transaction machines (similar to an Automated teller machine), or via a video conference enabled bank branch clarification.
(10) DSA is a Direct Selling Agent, who works for the bank based on a contract. Its main job is to increase the customer base for the bank." [Bank. Wikipedia]
The UML use case diagram example "Banking system" was created using the Concept Draw PRO diagramming and vector drawing software extended with the Rapid UML solution from the Software Development area of Concept Draw Solution Park.

8.2.1. E-Banking Use Case Diagram:

A use case diagram is a graphical representation of the various actions that users can take when interacting with an e-banking system. It is a useful tool for documenting and analyzing the different ways in which users can interact with the system.

The use case diagram for an e-banking system can be divided into three main sections: account management, transactions, and customer service.

Account management includes all of the actions that users can take in order to manage their account, such as creating a new account, viewing account information, and making changes to account settings.

Transactions includes all of the actions that users can take in order to conduct banking transactions, such as transferring funds, paying bills, and viewing transaction history.

Customer service includes all of the actions that users can take in order to get help from customer service representatives, such as contacting customer service, submitting a support ticket, and accessing online FAQs.

In the E- Banking system normally use case diagram is used to prepare and make understand any functional requirements of the system. Because it's specifying exact context of the software being paid. It helps to identify where each use-case specifies the behavior expected from software from the perspective of end-user as well as their relation with actors and system. Each use-case represents the function of the system which is already automated from system or maybe manual.

A use case is a detailed explanation of how visitors will use your website to complete tasks. It describes how a system behaves in response to a request from the viewpoint of a user. Each use case is described as a series of easy actions that start with the user's objective and finish when that objective is achieved. (N. 2022)

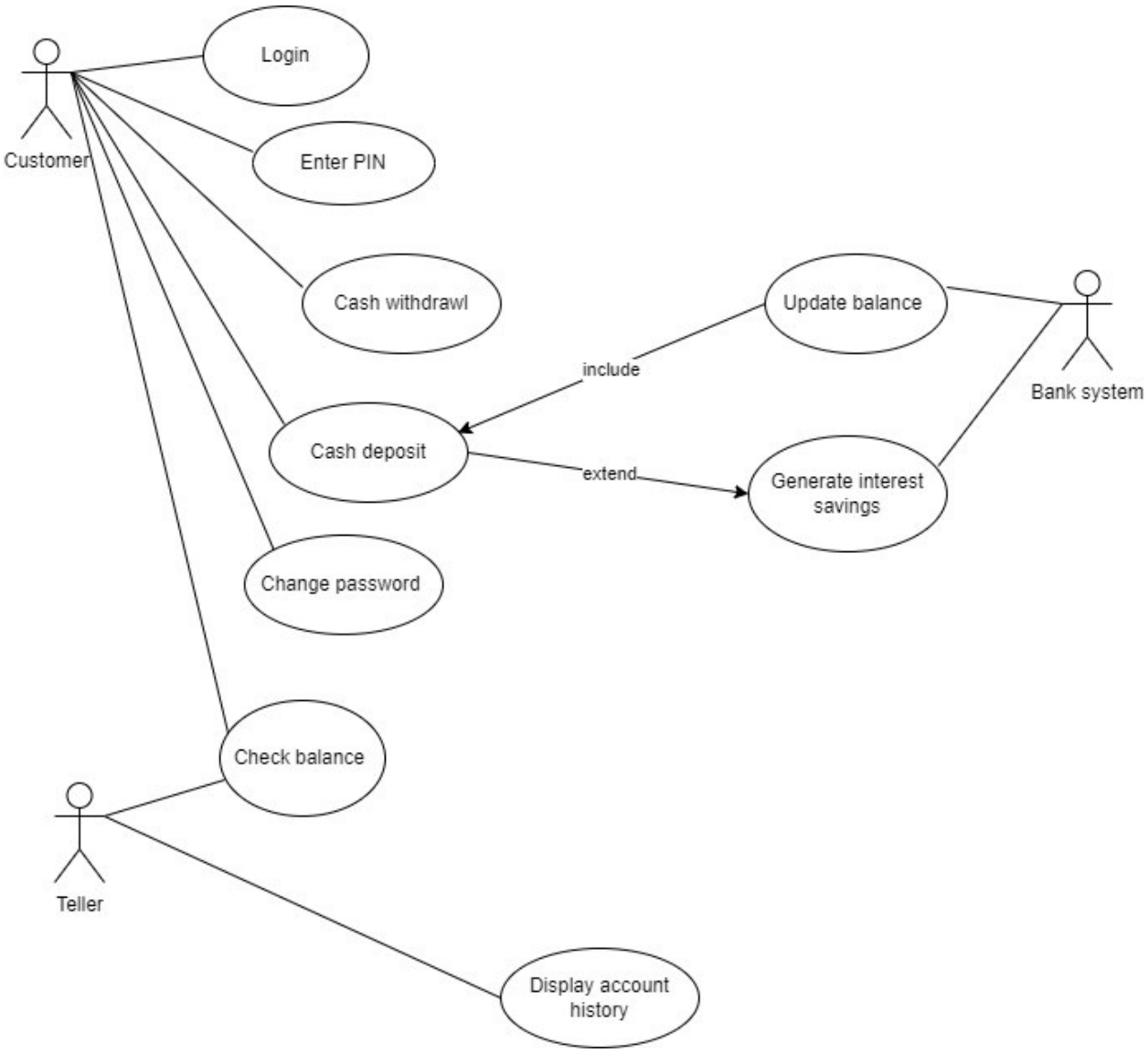

Figure 4: Use case diagram (source- author)

8.2.2. Use case decomposition using activity scenarios

Examples of potential activity situations for an online banking system include the following:

I. Signing in:

- the user's login information and password
- The system checks their credentials
- If successful, the user is logged in and shown their account dashboard.
- If not legitimate, the user is informed and given the option to try again or change their password.

II. Making a deposit:

- The user chooses their account.
- The user inputs the deposit amount.
- The system checks to make sure the account is legitimate and that the user has enough money.
- The system registers the transaction and adjusts the account balance.
- The user receives notification of the successful deposit and their updated balance.

III. Paying a bill:

- The account is selected by the user.
- The deposit amount is entered by the user.
- The system verifies that the account is valid and that the user has sufficient funds.
- The transaction is recorded by the system, and the account balance is changed.
- The user is informed of the successful deposit and the changes to their balance.

IV. Transacting money:

- User inputs the amount to transfer after choosing the accounts they wish to transfer funds from and to.
- The system checks to make sure both accounts are legitimate and that the user has enough money in their account to cover the transfer.
- The transfer process is started by the system, and the account balances are updated.
- The user receives notification of the successful transfer and their updated balance.

V. Looking at the transaction history:

- When a user chooses an account to see transaction history for, the system obtains that account's transaction history and then shows it to the user in a legible fashion.
- The transaction history can be filtered or sorted by the user as required.

These are just a few instances of how an internet banking system may be used. Depending on the particular requirements and capabilities of the system, there may be many additional use cases and activity situations to take into account in practice. Before the system is fully constructed or deployed, use case decomposition utilizing activity scenarios can assist to find possible problems, edge cases, or missing features.

8.3. Collaboration diagram:

Illustration of a UML collaboration diagram. The UML Collaboration Diagram library of the Fast UML Solution from the Software Development category of concept Draw Solution Park was used to build this sample in Concept Draw DIAGRAM, a diagramming and vector drawing program.

This sample demonstrates the contact list development process and may be used to staff hiring, employee training, and client recruitment. A collaboration diagram shows a

collaboration, which is a group of items connected by a common theme, and interaction, which is a series of messages passed back and forth among the objects to accomplish a goal.

The objects are contained in a box in a collaboration diagram. Indicate the message sent inside the specified use case, such like in a sequence diagram. The sequence is shown in a cooperation diagram by numbering the messages.

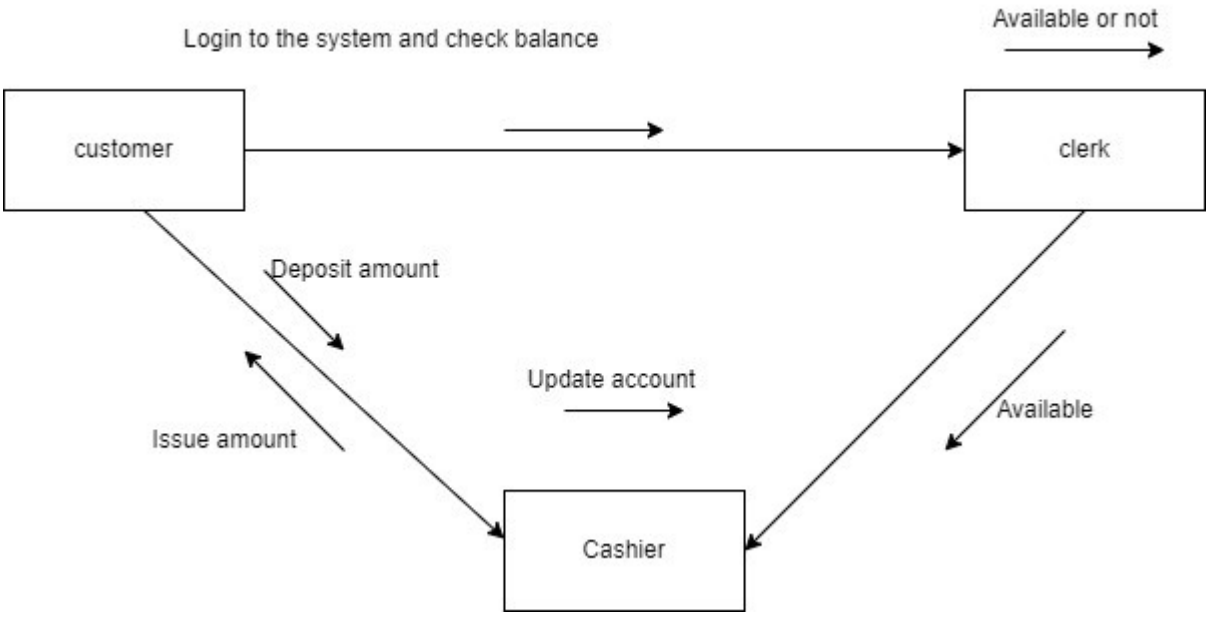

Figure 5: collaboration diagram (author)

8.4. Activity diagram for E-banking:

An activity diagram is a visual depiction of a certain process's phases from beginning to end. While using an online banking system, a transaction is normally completed after a number of user and system-initiated activities.

The Banking Management System includes the Activity Diagram for Online Banking System as a crucial component. This carefully crafted activity diagram demonstrates to the

panel and reader how the banking system would interact with its customers. With a related UML Activity Diagram that aids in understanding the activity, it streamlines the development of the banking system.

It provides a broad picture of the procedures, events, and activities that take place between users and the system. The activity diagram might also be used to check the accuracy of how the financial system functions.

8.4.1. How Activity diagram works for online banking:

In order for the programmers to have a foundation for how the software should interact with its users, the Online Banking System must include an activity diagram.

The reason for this is that the activity diagram directs the programmer as they develop the software and essential features. So, you must also finish the activity diagram if you desire nice and useful or simple-to-use software.

Here's how it might work:

1. The users start by logging in to their account on the bank's website or mobile app.
2. They select the type of transaction they want to perform (such as transferring money, paying bills or checking their balance).
3. The system verifies the user's credentials and retrieves their account information.
4. Depending on the type of transaction selected, the user may need to specify details such as the recipient's account number, the amount of money to be transferred, or the bill to be paid.
5. Once all required information is entered, the system verifies the transaction details and process the request.
6. If the transaction is successful, the system updates the user's account balance and provides a confirmation message.
7. If the transaction is not successful (due to insufficient funds, invalid account information etc.), the system will provide an error message and prompt the user to correct the issue.

This is just a basic overview of how an activity diagram for an online banking system might look like. The specific steps and actions involved would depend on the particular system and it's features.

Using activities to represent (action) states and the transitions between them, an activity diagram shows activity flows.

Activity: is a representation of an atomic activity being carried out.

Decision Block: Clearly specify the start and end state processes with start and end symbols. Block of decision-making that employs predetermined "guard conditions" to combine or branch activity flows.

Synchronization: Define where concurrent streams are formed and merged during synchronizing.

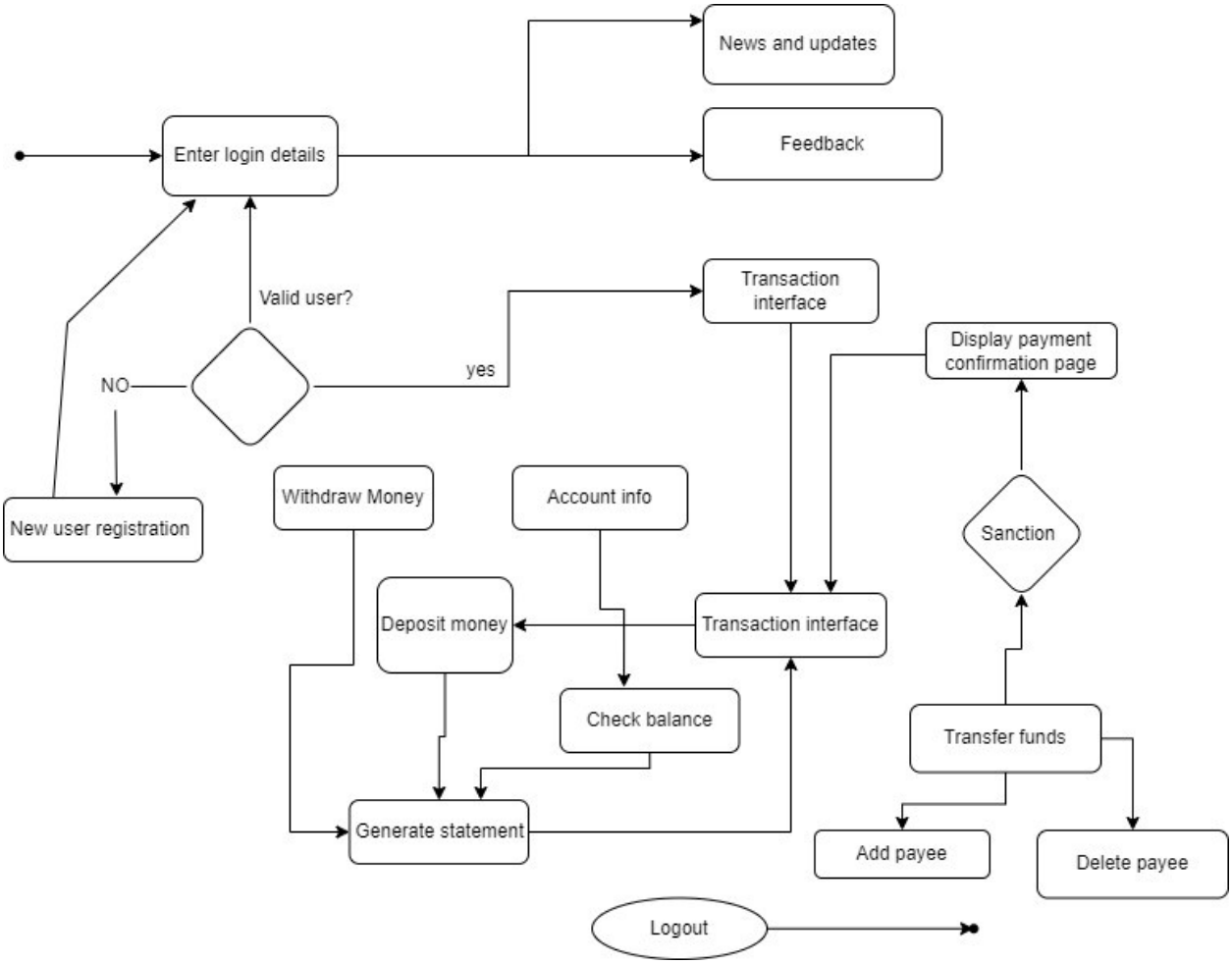

Figure 6: Activity Diagram (author)

8.5. State diagram for E- Banking:

A state diagram depicts the many states that an object can or will experience throughout the course of its existence in the system. a particular incident or occurrence the passage of time alone. Actions then result from the object's state changing. Moreover, these might be activities taken upon entering or departing the state.

Every state diagram has to start off in one condition and can end in another.

The two primary states of the state diagram shown in the following figure are the inactive state and the active state. As demonstrated, the system is capable of switching from an active state to an idle state and vice versa.

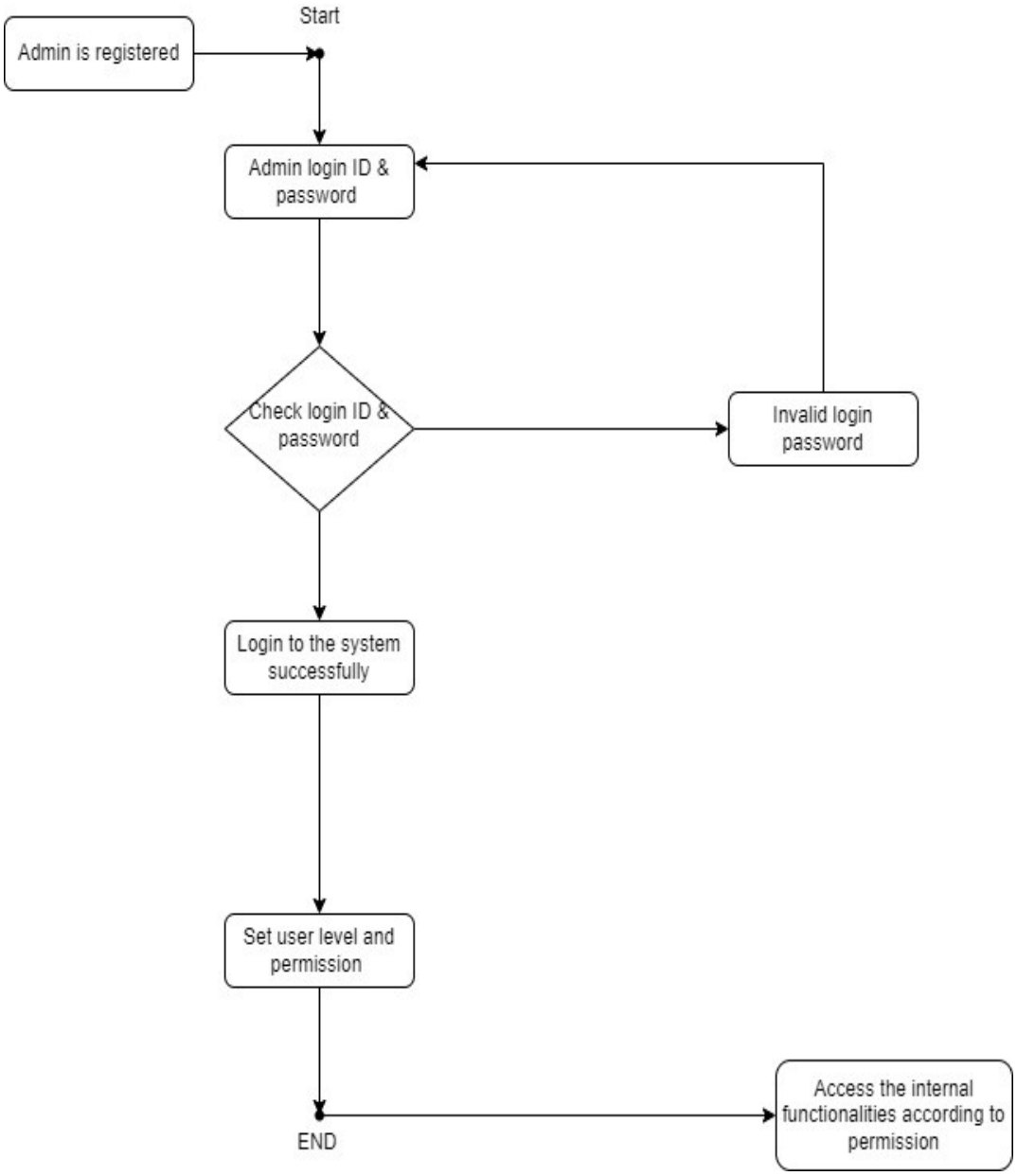

Figure 7: State diagram (source- author)

8.5.1. How state Diagram works in online banking and its steps:

A state diagram is a visual representation of the different states that a system can go through during its operation. In the case of an online banking system, it can help to map out the various stages involved in carrying out a typical transaction.

Here are some possible steps that could be included in a state diagram for an online banking system:

- Login: The user logs into their account using their username and password.
- Account selection: |The user selects which account they want to access, such as checking or savings.
- Balance check: The user views their account balance to see how much money they have available.
- Transaction selection: the user selects the type of transaction they want to perform, such as transferring funds or paying bills.
- Transaction details entry: The user enters the required details for the chosen transaction, such as the amount and recipient information.
- Verification: The system verifies that the transaction details are correct and that sufficient funds are available.
- Approval: The user approves the transaction to finalize it.
- Confirmation: The system confirms that the transaction has been completed successfully.
- Logout: The user logs out of their account to end their session.

These are just some possible steps, and depending on the specific requirements of the online banking system, there could be additional states and action involved. A well- designed state diagram can help to ensure that the system operates smoothly and efficiently, while also providing users with a clear understanding of the various stages involved in their transactions.

8.6. Deployment diagram:

The physical setup of the hardware and software elements required for the system's operation is depicted by a deployment diagram for online banking in UML. In order to support the online banking system, a network of computers, including servers, clients, and other hardware devices, must be distributed with software artifacts and system components.

The interface in the deployment diagram depicts the link between the network's hardware, system modules, and software artifacts. The interface, which regulates data flow and interactions between the various components, makes it easier for the components of the online banking system to connect with one another.

The servers, clients, databases, and other hardware devices are distributed over the system's network as shown in the deployment diagram, along with other software components and hardware devices. The deployment diagram may be used by the system administrator to control the deployment processes and make sure the system is operating properly.

In conclusion, the UML deployment diagram for online banking shows how hardware and software components are interconnected to create a network that enables communication and data flow between the various components of an online banking system. The infrastructure required for the integration of the components is provided by the interface, enabling the system to run smoothly. (Creately. n.d)

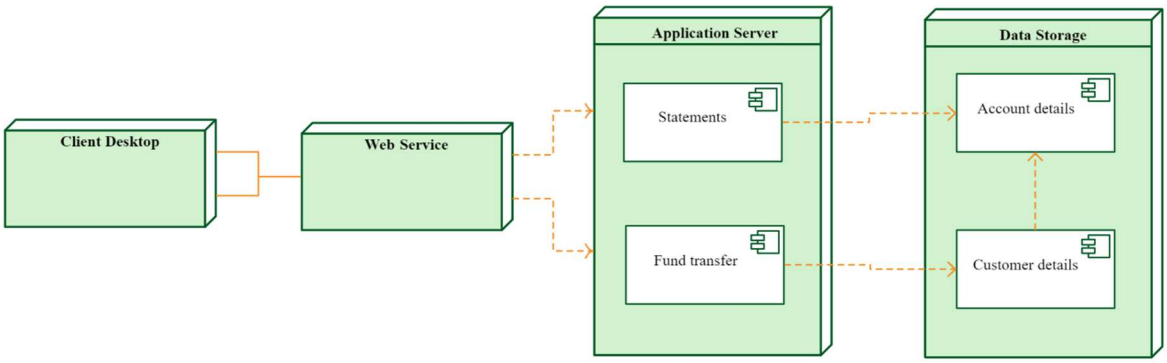

Figure 8: Deployment diagram (Creately. n.d)

In this diagram, we can see the following components:

1. Web Server: This component hosts the online banking application and serves web pages to the clients.
2. Application Server: This component handles the business logic of the online banking system, such as processing transactions and managing user accounts.
3. Database Server: This component stores the data for the online banking system, such as user account information, transaction history, and authentication data.
4. ATM Machines: These are the physical interfaces that allow customers to interact with the online banking system.
5. Mobile Devices: These are the mobile interfaces that allow customers to interact with the online banking system via mobile applications.

The connections and interfaces between the components are represented by the lines. For instance, in order to obtain and process data from the database server, the web server interacts with the application server. To send and receive data from the online banking system, the mobile devices and ATMs connect with the web server.

9. Online Banking component diagram in UML

A component diagram in UML is used to show the architecture of a system, and how its various components interact with each other. In the case of an online banking system, some of the components that might be included in the diagram are:

User Interface: This component will be responsible for interacting with the user and presenting them with the necessary banking options. This will involve web pages, forms, buttons, and other elements that make up the user interface.

Authentication: This component will be responsible for authenticating the user, so they can access their account information. This will involve reading the user's login information, verifying it, and granting them access.

Account Management: This component will handle all account transactions, including opening and closing accounts, transferring funds, and managing account balances. It may also include subcomponents for dealing with checking, savings, investment, and credit accounts.

Security: This component will be responsible for ensuring the security of the online banking system. This might include encryption, secure communication protocols, and other security measures to protect user data and transactions.

Database: This component will store all of the user's account information and transaction history. It will be responsible for quickly retrieving account information as needed for display on the user interface.

Notification: This component will be responsible for sending notifications to the user about account activity or any other relevant information.
These are just a few of the elements that are frequently seen in online banking systems. How these parts work together to build the overall system architecture is depicted in a UML component diagram. (N. 2022c)

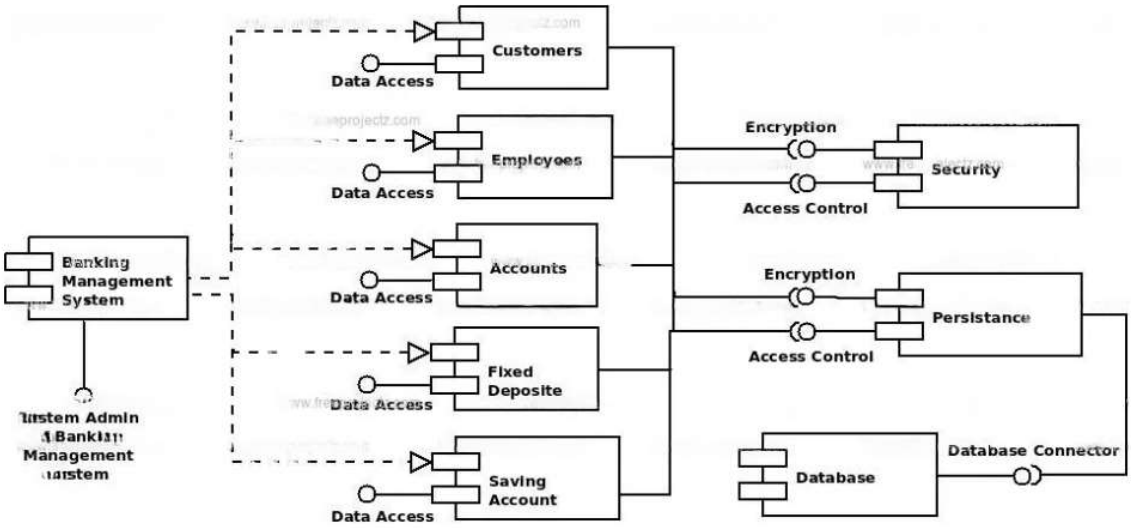

Figure 9: Component diagram (Banking Management System, 2018)

10. An Overview of UML

Booch, OMT (Rumbaugh), OOSE (Jacobson), Fusion (HP), OOA/OOD (Coad/Yourdon), Recursive Design (Shlaer/Mellor), CRC (Beck/Cunningham), Responsibility Driven Design (Wirfs-Brock), and Odell are just a few of the software development methodologies that used various modeling languages before the UML was introduced (in 1997). The "three amigos," Booch, Rumbaugh, and Jacobson, decided to develop UML by fusing the best aspects of their own efforts and adding new components. the meaning of the letter "U". (Booch, Rumbaugh and Jacobson,1998)

"Unified."

March 2000: Version 1.3
Version 1.5: 2003 March
July 2005: Version 2.0
February 2009: Version 2.2
May 2010: Version 2.3
August 2011, version 2.4.1

11. Description of scenario

11.1. Description of the activity scenario: Authorization

The user places the card in. The card's serial number is obtained by reading it on the terminal.

2. When the terminal requests a PIN, the user enters the right PIN and confirms.

3. When the terminal confirms the card number and PIN on the bank's system, the system will approve and assess.

4. The user may see the authorization outcome on the terminal.

11.2. Description of the alternative scenario of activities: Unsuccessful authorization

1. The card is inserted by the user. The card's serial number is read by the terminal from the card.

2. The user is prompted to input a PIN by the terminal, guesses the erroneous PIN, and then confirms.

3. The data is transmitted from the terminal system to the bank system.

4. The PIN is verified and the authorization is assessed by the bank's system.

5. Delivers the terminal system with the authorization outcome.

6. The terminal shows the user the authorization outcome. If the error is repeated three times, the authorization will keep the card.

11.3. Pin Change online scenario:

Here is an example scenario for a PIN change in an online banking application:

1. The user taps the "Change PIN" button after navigating to the "PIN Change" area of the online banking application.
2. The user is asked to input both their old and new PINs.
3. The user hits the "Submit" button after entering their new PIN and current PIN in the appropriate areas.

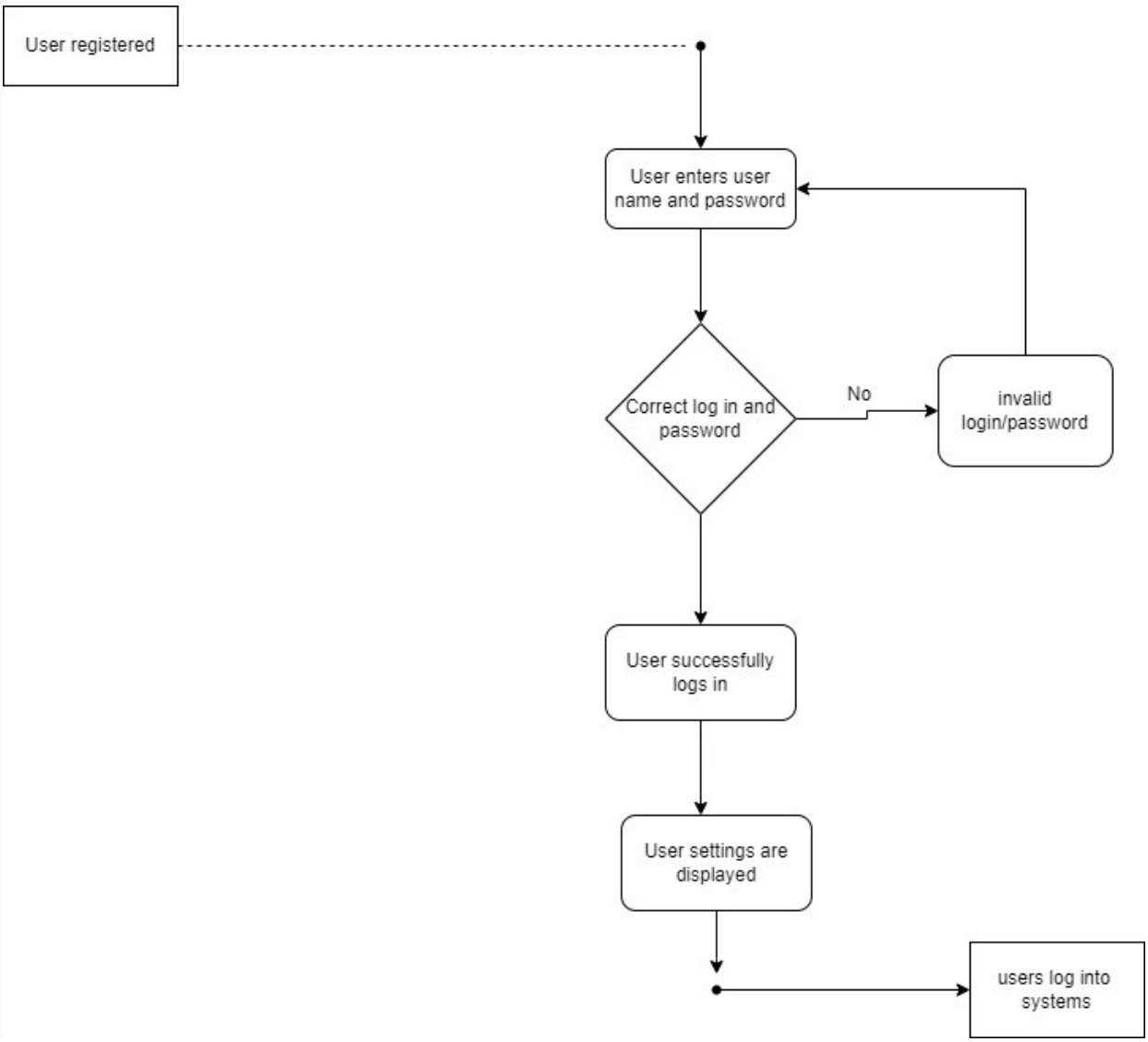

Figure 10: Pin change scenario (author)

Here are some key elements to include in your online banking login page:

Login form: This is the main component of the page, where users can enter their login credentials, such as username and password, to access their account.

Security features: Online banking systems require strong security features to protect user data. You should consider implementing features such as two-factor authentication, CAPTCHA verification, and SSL encryption to ensure the security of your users' information.

Error messages: You should include error messages that inform users if they have entered an incorrect username or password or if there is an issue with their account.

Forgot password link: It's common for users to forget their login credentials. Including a "forgot password" link can help users reset their password and regain access to their account.

Navigation menu: It's important to include a navigation menu that allows users to access other pages of the online banking application, such as account information and transaction history.

12. UML and online banking development process

Creating a digital infrastructure that enables users to conduct financial transactions and access banking services online is a key step in the development of online banking. This platform may offer functions including cash transfers, bill payment, account statement requests, account balance enquiries, and other services that are traditionally provided by a physical bank branch. (Conallen, 2002)

12.1. Waterfall development process:

The Waterfall model is a traditional software development methodology that follows a linear sequential approach with the phases of the software development lifecycle. [20] Here is an example of how the Waterfall model can be applied to develop an online banking system using UML:

1. Requirements Analysis:

Identify and gather requirements for the online banking system.

Define use cases and functional requirements using UML diagrams such as use case diagrams, activity diagrams, and sequence diagrams.

2. System Design:

Based on the requirements, design the system architecture, components, and interfaces.

Create UML diagrams such as class diagrams, component diagrams, and deployment diagrams to describe the system design

3. Implementation:

Write code to implement the system design.

Conduct unit testing to ensure that each component of the system works as expected.

4. Testing:

Conduct integration testing to ensure that all components of the system work together correctly.

Conduct system testing to validate that the system meets the requirements and functions correctly.

Conduct user acceptance testing to ensure that the system meets the needs of the users.

5. Deployment:

Deploy the system to the production environment.

Conduct post-deployment testing to ensure that the system functions correctly in the production environment.

6. Maintenance:

Maintain and support the system, including fixing bugs, adding new features, and updating the system to meet changing requirements.

UML diagrams may be used to describe and explain the system design and implementation at any point during the development process. Use case diagrams, for instance, can be used to depict system requirements, class diagrams, to depict system components and their connections, and deployment diagrams, to depict the design and deployment of the system. (Waterfall Methodology. 2023)

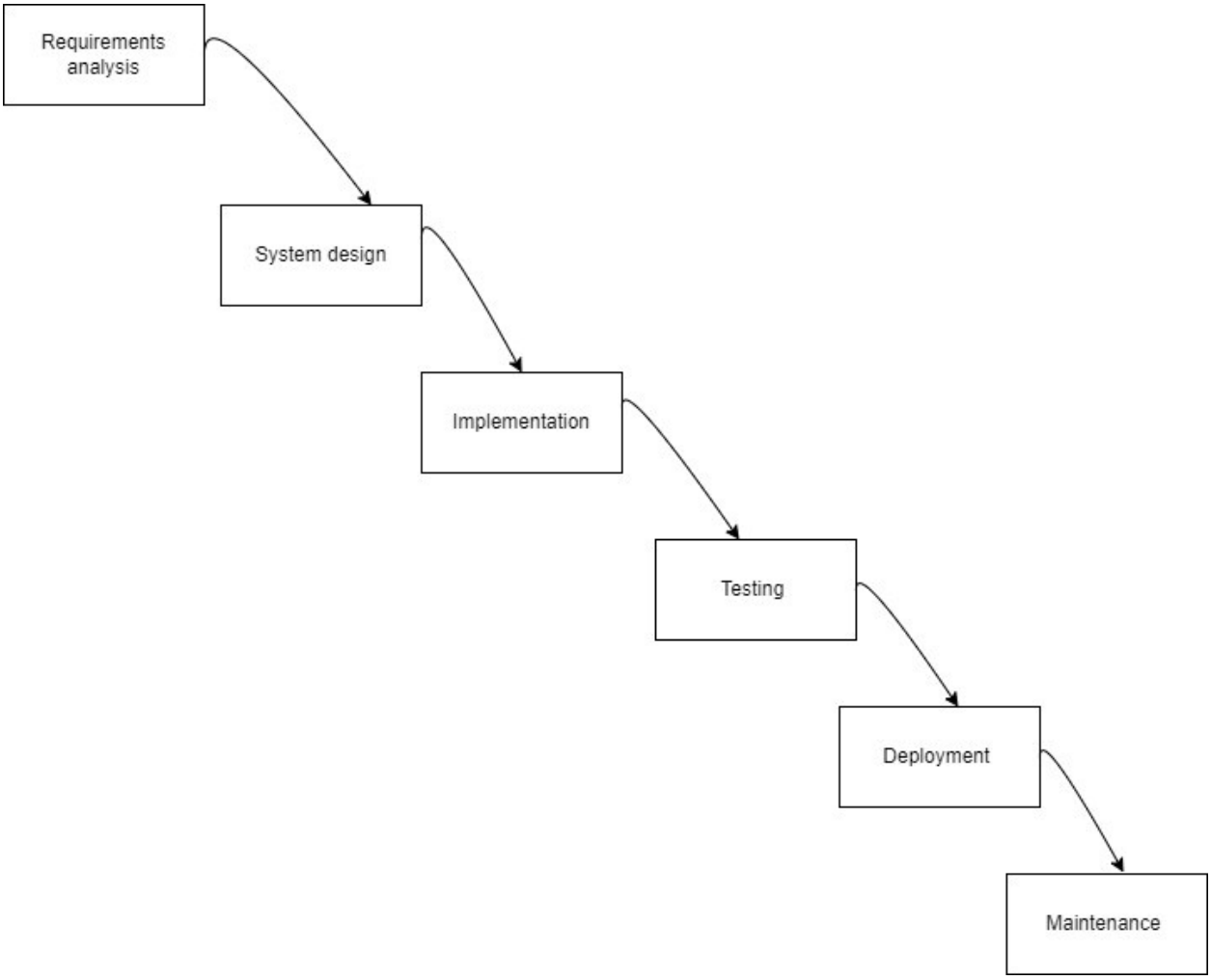

Figure 11: waterfall diagram (author)

12.2. Rapid application development for online banking:

Iterative development and fast prototyping are key components of the rapid application development (RAD) software development process. RAD enables for quick revisions and rapid delivery of features, making it a useful strategy for creating an online banking system.

The RAD approach typically involves the following phases

Requirements gathering: This involves identifying the needs and requirements of the customers and stakeholders. The requirements are typically documented in the form of user stories or use cases.

Prototyping: The development team creates a prototype of the online banking system based on the initial requirements. This prototype is used to validate the requirements and gather feedback from users.

Iterative development: Based on the feedback from users, the development team iteratively develops and refines the online banking system. The system is developed in small, incremental releases, with each release adding new features and functionality.

Testing: Each release of the online banking system is thoroughly tested to ensure that it meets the requirements and functions correctly.

Deployment: Once a release has been tested and approved, it is deployed to the production environment.

Maintenance: After the platform has been launched, it requires ongoing maintenance and support. This includes bug fixes, updates, and enhancements to the platform.

Since it enables for quick revisions and rapid delivery of functionality, the RAD method can be beneficial for online banking. This implies that new features may be swiftly added to the platform in response to shifting consumer demands and market dynamics. To guarantee the dependability and security of the platform, it is crucial to make sure that the right testing and quality assurance mechanisms are in place. (Sanjeev ,2008)

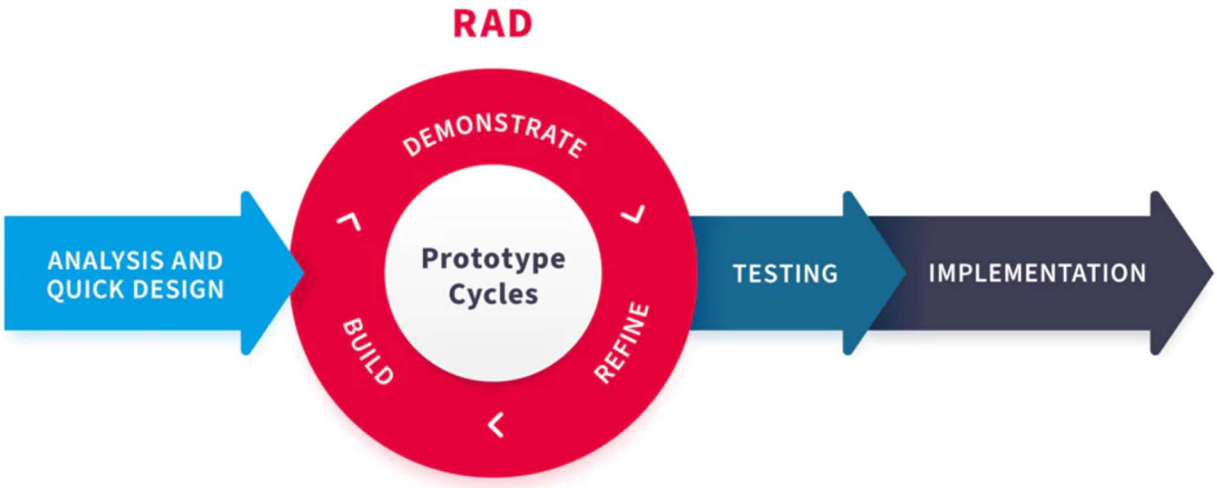

Figure 12: RAD [19]

13. Scrum

Scrum is an agile project management and completion framework. It is a well-liked process that is employed in the creation of software, including online banking systems. Scrum is a software development methodology that focuses on producing usable software in brief iterations, or "sprints," and places a strong emphasis on teamwork, self-organization, and continuous improvement.

Agile software development is an iterative and progressive method of creating software that places a focus on customer happiness, collaboration, and adaptability. Because agile enables quick iterations and rapid delivery of features based on user demands and input, it can be a successful method for creating online banking systems.

Since it enables for quick iterations and rapid delivery of products based on user demands and input, the Agile approach can be beneficial for online banking. As a result, the platform may swiftly include new features in response to shifting client demands and market dynamics. To ensure the dependability and security of the platform, it is crucial to make sure that suitable testing and quality assurance mechanisms are in place. A successful Agile development process also depends on good communication and cooperation between the development team and the client.

Here are the key roles, artifacts, and processes in Scrum for online banking development:

Roles: The key roles in Scrum are the Product Owner, Scrum Master, and Development Team.

The Product Owner is responsible for defining and prioritizing the features and requirements of the online banking system, based on customer needs and business goals.

The Scrum Master is responsible for facilitating the Scrum process and ensuring that the team is following the Scrum framework.

The Development Team is responsible for designing, developing, testing, and delivering the online banking system. (A. 2022)

Artifacts: The key artifacts in Scrum for online banking development are the Product Backlog, Sprint Backlog, and Increment.

The Product Backlog is a prioritized list of features and requirements for the online banking system.

The Sprint Backlog is a list of features and requirements that the Development Team plans to deliver during a specific sprint.

The Increment is the working version of the online banking system that is delivered at the end of each sprint.

Processes: The key processes in Scrum for online banking development are Sprint Planning, Daily Scrum, Sprint Review, and Sprint Retrospective.

Sprint Planning is a meeting where the Product Owner and Development Team collaborate to define the goals and scope of the upcoming sprint and create a Sprint Backlog.

Daily Scrum is a short daily meeting where the Development Team discusses progress, challenges, and plans for the day.

Sprint Review is a meeting where the Development Team presents the Increment to the Product Owner and other stakeholders and receives feedback.

Sprint Retrospective is a meeting where the Scrum Team reflects on the previous sprint and identifies opportunities for improvement.

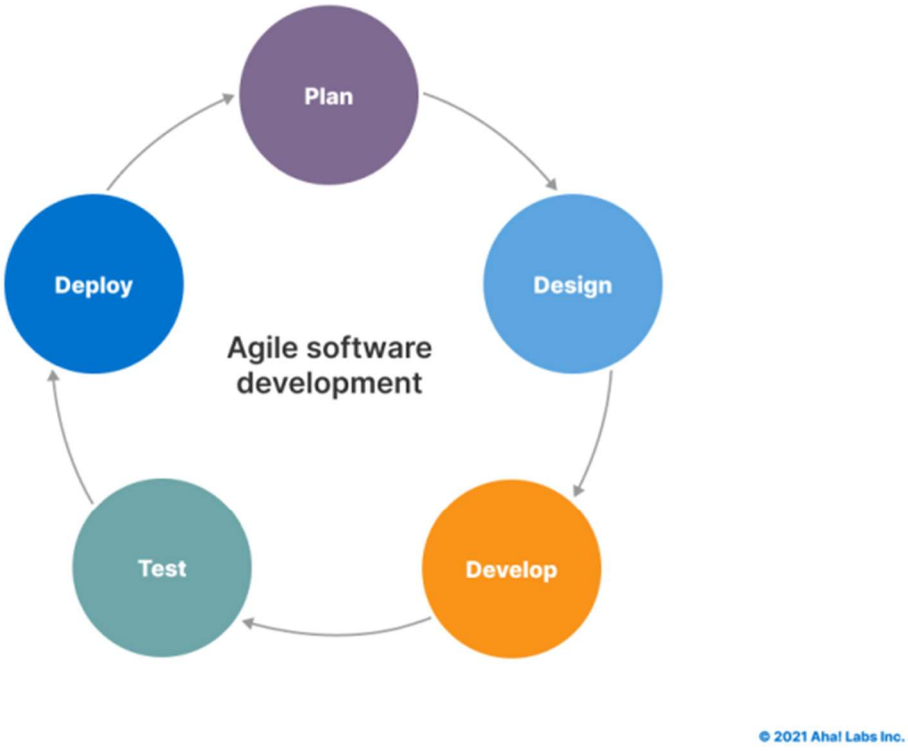

Figure 13: Scrum (A. 2022)

Scrum may be a successful methodology for creating online banking systems because it enables fast iterations, ongoing feedback, and communication between the development team and stakeholders. To guarantee the dependability and security of the system, it is crucial to make sure that the right testing and quality assurance mechanisms are in place.

14. System Architecture For E-Banking

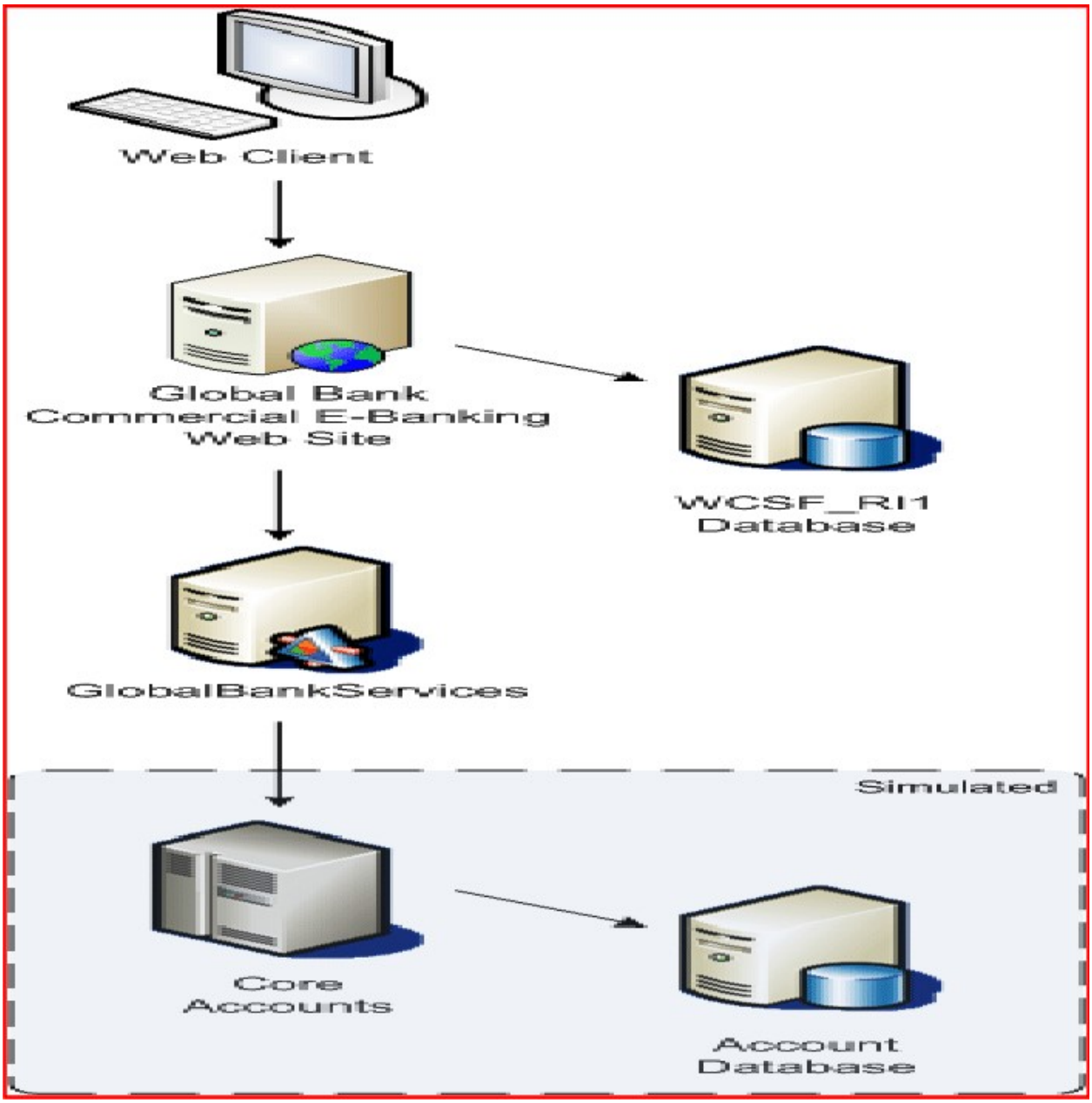

Figure 14: E-Banking Architecture (Beren)

15. Methods:

UML (Unified Modeling Language) is a standard way to create diagrams of software designs. UML can be used for many different types of diagrams, including class diagrams, component diagrams, and use case diagrams.

In an E- Banking system, the main goal is to design a system that will automate the processes and transactions associated with banking. This can include everything from online banking and bill pay to mobile banking and ATM transactions.

When designing an E- Banking system using UML, it is important to first understand the business requirements of the bank or financial institution. Once these requirements are understood, they can be translated into UML diagrams which will show the various components of the system and how they interact with each other.

Some common UML diagrams used in E- Banking systems include class diagrams, component diagrams, and use case diagrams. Class diagrams show the various objects that make up the system and their relationships with each other. Component diagrams show how those objects are implemented as software components. Use case diagrams show how those software components interact with each other to provide functionality for the end users.

Once the initial UML design is complete, it can be further refined and expanded upon as needed. It is also important to keep in mind that UML designs are not set in stone – they can be adapted and changed as needed during the development process.

16. Conclusion:

The Unified Modeling Language (UML) is a modeling language that has been designed to provide a standard way to visualize the design of a system. The UML provides a common vocabulary that can be used by analysts, designers, and developers to create models of software systems. In this article, we have looked at how the UML can be used in the context of designing an E-Banking system. We have also seen how the different diagrams in the UML can be used to model different aspects of the system.

E-banking is now the most cutting-edge method of delivering financial services and is already regarded as a common route of distribution. belongs to the financial services with the fastest growing market.

The goal of this diploma thesis was to assess the requirements of the bank's self-service terminal and develop an information system to meet those demands.

Any self-service terminal can use the analysis and solution design since they are not particular to any one target environment. System design is envisioned as the creation of distinct components. Therefore, using only a portion of the system's features on a self-service terminal won't be an issue. (Willcocks, & Lester, 1996 and Dawson 1998)

A thorough study covers several aspects of the system. The fundamental perspective is a functional view that discusses the system's process side. In context diagrams and use cases, the functioning of the system is described. The data view includes data entities. A data view displays a class diagram that includes all object classes and their connections to the system. The time-dependent system actions are described by the dynamic view. The state diagram and the activity diagram both depict a dynamic perspective. The component diagram and deployment diagram, which serve as representations of the different system components, are displayed in the implementation preview. The system's constituent components are depicted in the component diagram.

The job of utilizing UML to analyze electronic banking was performed in this diploma thesis. We utilized the Unified Software Development Methodology Process to conduct a thorough study. Due to its connection to the UML modelling language, I choose the technique.

This diploma thesis's output is a thorough investigation of the self-service bank terminal's system and screen design.

17. References:

1. Ambler, S. W., 2004. The Object Primer: Agile Model-Driven Development with UML 2.0, Cambridge, England: Cambridge University Press.
2. Booch, G., Rumbaugh, J. and Jacobson, I., 1998. The Unified Modeling Language User Guide, Boston, MA: Addison Wesley.
3. Conallen, J., 2002. Building Web Applications with UML, Boston, MA: Addison Wesley
4. Arlow, J. and Neustadt, I., 2005. UML 2 and the Unified Process: Practical Object• Oriented Analysis and Design, (2nd ed.). Boston, MA: Addison Wesley
5. Hamilton, K. and Miles, R., 2006. Learning UML 2.0, Boston, MA: O'Reilly
6. Jacobson, I. and Cook, S. The Road Aheadfor UML [online]. [Accessed 10 January 2018].
7. OMG, 2017. OMG Unified Modeling Language (OMG UML): Infrastruct ure specification, Version 2.5.1, formal/2017-12-05, s.l.: Object Management Group.
8. Karim, A. J., 2011. The Significance of Management Information Systems for Enhancing Strategic and Tactical Planning. Journal of Information Systems and Technology Management, 8(2), p. 459-470.
9. MIT Press. Information Systems: Introduction and Concepts [online]. [Accessed 13September 2017].
10. Waterfall Project Management Software. (2023, March 24).
11. Banking System | UML use case diagram - Banking system | Bank System | Diagram for Banking System. (n.d.).
12. Deployment Diagram for Online Banking Transaction System [classic] | Creately. (n.d)
13. UML 2 Activity Diagramming Guidelines. Agile Modeling (AM) Home Page: Effective
14. Practices for Modeling and Documentation [online].
15. Peppard, J. and Ward, J., 2016. The Strategic Management of Information Systems: Building a Digital Strategy, (4th ed.). Hoboken, NJ: John Wiley & Sons, Inc.

16. Rosenberg, D. and Stephens, M., 2007. Use Case Driven Object Modeling with UML: Theory and Practice, New York, NY: Apress.
17. Rumpe, B., 2016. Modeling with UML: Language, Concepts and Methods, Cham, Switzerland: Springer.
18. N. (2022, August 5). Online Banking System UML Diagram. Itsourcecode.com.
19. IBM Documentation. (n.d.)..
20. GeeksforGeeks. (2021, September 6). Use Case Diagram for Online Banking System
21. Piluta, R. (2023, March 16). Rapid Application Development: Advantages and Disadvantages – NIX United. NIX United – Custom Software Development Company in US.
22. SDLC - Waterfall Model. (n.d.).
23. Try QA. (n.d.). https://tryqa.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it/
24. Sebastian, K. (2019, November 21). Five reasons why banks should adopt Agile Product Development. Penser.
25. Waterfall Methodology. (2023, March 27). ProjectManager.
26. A. (2022a, May 13). What Is Agile Software Development? Aha! Software.
27. Figure 1. E-Banking Architecture. (n.d.). ResearchGate.
28. Kurnia, S., Peng, F., & Liu, Y. R. (2010). Understanding the adoption of electronic banking in China. In 43rd Hawaii International Conference on System Sciences, Honolulu, Hawaii, USA, pp. 1–10.
29. LOW TDR PAYMENT GATEWAY DIRECTLY FROM BANK on WordPress.com : Image. (n.d.).
30. Bahl, D. S. (2012). E-banking: Challenges and policy implications. International Journal of Computing & Business Research, 229–6166.
31. Zarei, S. (2011). Risk management of internet banking. In 10th WSEAS International conference on Artificial Intelligence, Knowledge Engineering and Data Bases, Cambridge, UK, pp. 134–139.
32. sanjeev banzal, “Mobile Banking and M-Commerce and Related Issues,” 2008.
33. Shy, Oz, and Tarkka, J. (2002), The Market for Electronic Cash Cards, Journal of Money, Credit and Banking, Vol. 34, pp. 299-314.
34. Willcocks, L. P. & Lester, S. (1996) and Dawson (1998) "Beyond the IT Productivity Paradox", European Management Journal, 14 (3), pp. 279-290.

35. Berentsen, A. (1998), Monetary Policy Implications of Digital Money, Kyklos, Vol. 51, pp. 89 117.
36. N. (2022c, August 2). Component Diagram for Bank Management System | UML. Itsourcecode.com.
37. Banking Management System Component UML Diagram | FreeProjectz. (2018, February 8). https://www.freeprojectz.com/uml-diagram/banking-management-system-component-diagram