|  |
| --- |
| Optimizing User Experience An In-Depth Look at ATM Simulation Systems |
| A CAPSTONE PROJECT  Submitted By |
|  |
| K M Haneef  [192211905] |
| In Partial Fulfillment for the completion of the course |
| CSA0912  PROGRAMMING IN **JAVA FOR ACCESSING DATABASE**  SEP 2024 |
|  |
| SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES  CHENNAI - 602105  TAMIL NADU, INDIA |



# **BONAFIDE CERTIFICATE**

This is to certify that the project report entitled **OPTIMIZING USER EXPERIENCE AN IN-DEPTH LOOK AT ATM SIMULATION SYSTEMS** submitted by K M Haneef 192211905 to Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai, is a record of bonafide work carried out by him/her under my guidance. The project fulfills the requirements as per the regulations of this institution and in my appraisal meets the required standards for submission.

Dr K. Jayasakthi velmurugan

**COURSE FACULTY**

*SIMATS, Chennai – 602105*

Saveetha school of engineering

# **ACKNOWLEDGEMENT**

This project work would not have been possible without the contribution of many people. It gives me immense pleasure to express my profound gratitude to our Honorable Chancellor **Dr. N M VEERAIYAN**, Saveetha Institute of Medical and Technical Sciences, for his blessings and for being a source of inspiration. I sincerely thank our Director of Academics **Dr. DEEPAK NALLASWAMY,** SIMATS, for his visionary thoughts and support. I am indebted to extend my gratitude to our Director **Dr. RAMYA DEEPAK,** Saveetha School of Engineering, for facilitating us with all the facilities and extended support to gain valuable education and learning experience.

I register my special thanks to **Dr. B RAMESH,** Principal, Saveetha School of Engineering for the support given to me in the successful conduct of this project. I wish to express my sincere gratitude to my Course faculty **K**,**Jayasakthi velumurugan** for his inspiring guidance, personal involvement and constant encouragement during the entire course of this work.

I am grateful to Project Coordinators, Review Panel External and Internal Members and the entire faculty of the Department of Design, for their constructive criticisms and valuable suggestions which have been a rich source to improve the quality of this work.

**INDEX**

[BONAFIDE CERTIFICATE](#_jamwo2607bmz) 1

[ACKNOWLEDGEMENT](#_8yiu0rn5tdsd) 2

[1. ABSTRACT](#_u3bnbdop1w2) 5

[2. INTRODUCTION](#_ytjahc6y3luo) 6

[3. ARCHITECTURE DIAGRAM](#_wgnqfbsa5xqn) 7

[4. FLOWCHART](#_uylcc3c0vobr) 8

[5. UML DIAGRAM](#_p5otxqr3lwgf) 9

[6. CLASS DIAGRAM](#_mwjh6gk15gp3) 10

[7. CODE IMPLEMENTATION](#_ack6ofphfxze) 11

[7.1 JAVA CODE](#_es8tj0pw01co) 11

[7.2 HTML CODE](#_lpv4bxmmf4f6) 12

[7.3 CSS CODE](#_p66bhylntj5) 13

[8. OUTPUT SCREENSHOT](#_c61jzdgd0maj) 14

[9. CONCLUSION](#_3admx128b1cd) 16

[10. REFERENCES](#_weq4at85bunu) 17

# **1. ABSTRACT**

ATM (Automated Teller Machine) systems play a pivotal role in modern banking, providing users with quick and convenient access to financial transactions. As technology evolves, so does the need to optimize user experiences and enhance the efficiency of these systems. This capstone project focuses on the intricacies of ATM simulation systems, examining the architecture, flow, and user interactions that contribute to a seamless experience. By employing Java for enterprise applications, this study delves into the development and simulation of ATM functionalities, offering insights into how user interfaces, system architectures, and transaction processes can be streamlined for optimal performance. The study begins by outlining the essential components of an ATM system, including user interaction, server communication, database management, and transaction handling. A detailed architecture diagram illustrates the connections between these components, providing a framework for understanding how ATM systems function in real-world scenarios. The project's flowchart further elucidates the sequence of operations, from user authentication through PIN entry to the selection of transaction types and the processing of deposits and withdrawals. By analyzing these processes, the study identifies potential bottlenecks and areas for improvement, offering recommendations for enhancing system responsiveness and user satisfaction. Java code implementations are presented, highlighting the practical aspects of building an ATM simulation system. The code snippets demonstrate the logic behind PIN validation, balance inquiries, and transaction updates, emphasizing the importance of robust error handling and security measures.

# **2. INTRODUCTION**

The evolution of banking technology has brought about significant changes in how financial transactions are conducted. Automated Teller Machines (ATMs) have become a cornerstone of modern banking, providing customers with a convenient means to access their accounts, withdraw cash, and perform various financial transactions without the need for human assistance. As the demand for seamless and efficient banking experiences grows, optimizing ATM systems to meet user expectations has become increasingly important.

This capstone project explores the optimization of user experiences in ATM simulation systems, with a focus on enhancing system architecture, user interactions, and transaction processes. The study aims to provide a detailed understanding of how ATM systems operate, identifying key components and their interactions to propose improvements that can lead to more efficient and user-friendly systems.

The project begins by examining the fundamental components of ATM systems, including the user interface, web application, server, database, and transaction modules. Each component plays a crucial role in ensuring smooth and secure transactions, and understanding their interactions is essential for optimizing system performance.

A detailed architecture diagram is presented to illustrate the relationships between these components, providing a visual representation of how ATM systems function. The diagram serves as a blueprint for analyzing system operations, highlighting areas where improvements can be made to enhance user experiences.

The flowchart of ATM operations further breaks down the sequence of interactions, from the initial user authentication through PIN entry to the selection of transaction options and the processing of deposits and withdrawals. By examining each step in detail, the study identifies potential challenges and proposes solutions to streamline operations and reduce user wait times.

Java programming language is employed in this project to simulate ATM functionalities, showcasing the practical aspects of developing a robust and efficient system. The code implementation section provides insights into the logic behind key operations, emphasizing the importance of secure coding practices and error handling to prevent unauthorized access and ensure accurate transactions.

Ultimately, this capstone project seeks to contribute to the field of ATM system development by offering a comprehensive analysis of user experience optimization techniques. By understanding the intricacies of ATM operations and identifying areas for improvement, the study aims to provide valuable insights for developers, financial institutions, and stakeholders interested in enhancing the efficiency and user-friendliness of ATM systems. As technology continues to evolve, the need for innovative solutions in banking technology remains paramount, and this project serves as a step towards achieving that goal.

# **3. ARCHITECTURE DIAGRAM**

|  |
| --- |
| Figure 3.1 : Architecture Diagram |
| The architecture diagram provided represents the structure and interactions within an ATM system. It outlines how different components and modules within the system communicate with each other to process transactions, manage user interactions, and maintain account balances. Here is a detailed explanation of each component and their relationships: **Components of the ATM System Architecture**  1. **User**:    * The user is the individual interacting with the ATM. They input commands and receive feedback from the system. 2. **Web Application**:    * The Web Application serves as the interface between the user and the server. It handles user inputs, displays information, and forwards requests to the server for further processing. 3. **Server**:    * The server is the core component that processes requests from the Web Application. It coordinates between various modules to ensure the correct execution of transactions. 4. **Database**:    * The Database stores all the data related to user accounts, including balance information, transaction history, and user authentication details. It is accessed by the server to retrieve and update account information. 5. **ATM System**:    * The ATM System is a high-level abstraction that encompasses various functionalities and modules required for the operation of the ATM. It acts as the orchestrator, managing user interactions, transactions, and balance updates. 6. **Transaction Module**:    * The Transaction Module is responsible for handling specific transaction operations such as deposits and withdrawals. It validates the transaction amounts and interacts with the Balance Module to update the user's account balance. 7. **Balance Module**:    * The Balance Module manages the user's account balance. It performs operations to update the balance based on the transaction type (deposit or withdrawal) and ensures that the account remains accurate.  **Interactions and Relationships**  1. **User to Web Application**:    * The user interacts directly with the Web Application, entering their PIN, selecting transaction options, and inputting transaction amounts. The Web Application serves as the interface through which the user communicates with the system. 2. **Web Application to Server**:    * The Web Application forwards user inputs and requests to the server. For example, when the user enters their PIN or selects a transaction option, the Web Application sends these inputs to the server for processing. 3. **Server to Database**:    * The server interacts with the Database to retrieve and update account information. For instance, during PIN validation, the server queries the Database to verify the entered PIN against stored user credentials. Similarly, for balance inquiries and transaction updates, the server accesses and modifies the relevant data in the Database. 4. **Web Application to ATM System**:    * The Web Application forwards user requests to the ATM System for further processing. This includes actions such as showing the transaction form or validating the entered amount. 5. **ATM System to Transaction Module**:    * The ATM System delegates specific transaction-related tasks to the Transaction Module. For example, when the user enters an amount to deposit or withdraw, the ATM System passes this information to the Transaction Module for validation and processing. 6. **ATM System to Balance Module**:    * The ATM System also interacts with the Balance Module to update the user's account balance. Depending on the transaction type, the ATM System instructs the Balance Module to either add the deposited amount or subtract the withdrawn amount from the balance. |

# **4. FLOWCHART**

|  |
| --- |
| Figure 4.1 : Flowchart |
| The flowchart provided depicts the sequence of operations within an ATM system, from the initial interaction with the user to the final transaction completion. It illustrates how the system handles various scenarios, including PIN validation, transaction selection, amount validation, and balance updates. Here is a detailed explanation of each step and decision point in the flowchart: **Flowchart Nodes and Paths**  1. **Start**:    * The process begins at the "Start" node, initiating the sequence of interactions between the user and the ATM system. 2. **Enter PIN**:    * The first user action is to enter their Personal Identification Number (PIN) into the ATM. This input is crucial for user authentication. 3. **Is PIN Correct?**:    * The system checks if the entered PIN is correct. This is a decision node where the process can branch into different paths based on the PIN validation result. 4. **PIN Incorrect**:    * If the PIN is incorrect, the system prompts the user to re-enter their PIN. This loop continues until the user enters the correct PIN or chooses to exit. 5. **Choose an Option**:    * Once the PIN is validated, the user is presented with various transaction options such as checking the balance, depositing money, or withdrawing money. 6. **Check Balance**:    * If the user selects the "Check Balance" option, the system retrieves and displays the user's current balance. This process then concludes the session for this transaction type. 7. **Show Transaction Form**:    * For deposit or withdrawal transactions, the system displays a transaction form where the user can enter the desired amount. This node leads to a subsequent decision node where the entered amount is validated. 8. **Enter Amount**:    * The user inputs the amount they wish to deposit or withdraw. 9. **Is Amount Valid?**:    * The system checks whether the entered amount is valid. This includes checking for non-negative values and, in the case of withdrawals, ensuring the amount does not exceed the available balance. 10. **Invalid Amount**:     * If the amount is invalid (e.g., negative value, exceeds balance), the system prompts the user to re-enter the amount. This loop continues until a valid amount is entered. 11. **Deposit Amount**:     * If the user chooses to deposit money and the amount is valid, the system adds the entered amount to the user's balance. This transaction type then concludes the session. 12. **Withdraw Amount**:     * If the user chooses to withdraw money and the amount is valid, the system checks if there are sufficient funds. 13. **Insufficient Funds**:     * If the balance is insufficient for the requested withdrawal, the system notifies the user of insufficient funds, and the session concludes. 14. **Sufficient Funds**:     * If the balance is sufficient, the system deducts the requested amount from the user's balance and completes the withdrawal transaction. 15. **End**:     * Each of the above paths eventually leads to the "End" node, indicating the completion of the user's session with the ATM system. |

# **5. UML DIAGRAM**

|  |
| --- |
| Figure 5.1 : UML Diagram |
| The UML sequence diagram provided illustrates the interactions between a user and an ATM system during a transaction process. It highlights the key steps and interactions involved in verifying a user's PIN, selecting a transaction, validating the transaction amount, and updating the user's balance. **Actors and Participants**  1. **User**: The person interacting with the ATM system. 2. **ATM System**: The central system that manages the ATM operations, interacts with the user, and coordinates with other modules. 3. **Transaction Module**: A component within the ATM system that handles transaction-related operations such as validating and processing deposits or withdrawals. 4. **Balance Module**: Another component within the ATM system responsible for managing and updating the user's account balance.  **Steps in the Sequence Diagram**  1. **User Enters PIN**:    * The process begins with the user entering their PIN into the ATM system. This interaction is initiated by the user and is sent to the ATM system for validation. 2. **ATM System Validates PIN**:    * The ATM system receives the PIN entered by the user and validates it. This step is internal to the ATM system and does not involve interaction with other participants.    * If the PIN is correct, the ATM system sends a message back to the user indicating that the PIN is accepted. If the PIN is incorrect, the user is prompted to re-enter their PIN. 3. **User Chooses Option**:    * After the PIN is accepted, the user is presented with options such as checking the balance, depositing money, or withdrawing money. The user selects one of these options and the selection is sent to the ATM system. 4. **ATM System Shows Transaction Form**:    * Based on the user's selection, the ATM system displays the appropriate transaction form. This involves prompting the user to enter the amount they wish to deposit or withdraw. 5. **User Enters Amount**:    * The user enters the transaction amount into the ATM system. This amount is then sent to the Transaction Module for validation. 6. **Transaction Module Validates Amount**:    * The Transaction Module receives the entered amount and validates it. This involves checking if the amount is a valid number and, in the case of a withdrawal, ensuring that the amount does not exceed the user's balance.    * The validation result (valid or invalid) is sent back to the ATM system. 7. **Updating Balance**:    * If the transaction amount is valid, the Transaction Module proceeds to update the balance. This involves either adding the deposit amount to the user's balance or subtracting the withdrawal amount from the user's balance. The updated balance information is sent to the Balance Module. 8. **Display New Balance**:    * Finally, the ATM system communicates the new balance to the user, concluding the transaction process. |

# **6. CLASS DIAGRAM**

|  |
| --- |
| Figure 6.1 : Class Diagram |
| The class diagram provided offers a structured representation of the ATM system's components and their relationships. It includes the classes for the main system functionalities, user interactions, and transaction processes. Here's a detailed explanation of each class and its attributes, methods, and interactions: **Classes in the ATM System**  1. **ATM System**:    * Attributes:      + **balance**: This attribute represents the current balance in the user's account. It is a floating-point number.    * Methods:      + **enterPin():** This method handles the process of entering and validating the user's PIN.      + **checkBalance():** This method retrieves and displays the user's current balance.      + **showTransactionForm():** This method displays the form for entering transaction details such as deposit or withdrawal amounts.      + **submitTransaction():** This method processes the transaction based on the entered details and updates the balance accordingly. 2. User Interaction:    * Methods:      + **enterPin():** This method allows the user to enter their PIN.      + **prompt():** This method displays prompts to the user, asking for input or providing information.      + **displayMessage():** This method displays messages to the user, such as errors or confirmations. 3. Transaction:    * Attributes:      + **amount:** This attribute represents the amount of money involved in a transaction. It is a floating-point number.    * Methods:      + **validateAmount():** This method checks if the entered transaction amount is valid.      + **deposit():** This method processes a deposit transaction by adding the entered amount to the balance.      + **withdraw():** This method processes a withdrawal transaction by subtracting the entered amount from the balance if there are sufficient funds.  Relationships Between Classes  1. Inheritance:    * The **UserInteraction** and **Transaction** classes inherit from the **ATM System** class. This inheritance signifies that these classes are specialized versions of the **ATM System** class, each adding specific functionality.    * The **Transaction** class also has an association with the **validateAmount** method, indicating that validation of the transaction amount is a critical part of the transaction process.  Interactions and Workflow  1. User Enters PIN:    * The user starts by entering their PIN through the **UserInteraction** class. The **enterPin** method in the **ATM System** class is called to validate the PIN. 2. Checking Balance:    * If the user chooses to check their balance, the **checkBalance** method in the **ATM System** class retrieves and displays the current balance. 3. Transactions:    * For deposit or withdrawal transactions, the **showTransactionForm** method in the **ATM System** class displays the form for entering transaction details.    * The **submitTransaction** method processes the transaction. If it's a deposit, the **deposit** method in the **Transaction** class is called to add the amount to the balance. If it's a withdrawal, the **withdraw** method checks for sufficient funds and deducts the amount from the balance if possible. |

# **7. CODE IMPLEMENTATION**

## **7.1 JAVA CODE**

let balance = 1000; // Example starting balance

function enterPin() {

const pin = prompt("Please enter your PIN:");

if (pin === "1234") {

document.getElementById('message').innerText = "PIN accepted. Choose an option.";

} else {

document.getElementById('message').innerText = "Incorrect PIN. Try again.";

}

}

function checkBalance() {

document.getElementById('message').innerText = Your balance is $${balance};

}

function showTransactionForm(type) {

document.getElementById('transaction-form').style.display = 'block';

document.getElementById('transaction-form').dataset.transactionType = type;

document.getElementById('message').innerText = Enter amount to ${type};

}

function submitTransaction() {

const amount = parseFloat(document.getElementById('amount').value);

const transactionType = document.getElementById('transaction-form').dataset.transactionType;

if (isNaN(amount) || amount <= 0) {

document.getElementById('message').innerText = "Please enter a valid amount.";

return;

}

if (transactionType === 'deposit') {

balance += amount;

document.getElementById('message').innerText = Deposited $${amount}. Your new balance is $${balance}.;

} else if (transactionType === 'withdraw') {

if (amount > balance) {

document.getElementById('message').innerText = "Insufficient funds.";

} else {

balance -= amount;

document.getElementById('message').innerText = Withdrew $${amount}. Your new balance is $${balance}.;

}

}

document.getElementById('transaction-form').style.display = 'none';

document.getElementById('amount').value = '';

}

## **7.2 HTML CODE**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Saveetha ATM Simulation System</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<div id="atm-container">

<h1>Saveetha ATM</h1>

<div id="screen">

<p id="message">Welcome to Saveetha ATM</p>

</div>

<div id="keypad">

<button onclick="enterPin()">Enter PIN</button>

<button onclick="checkBalance()">Check Balance</button>

<button onclick="showTransactionForm('deposit')">Deposit</button>

<button onclick="showTransactionForm('withdraw')">Withdraw</button>

</div>

<div id="transaction-form" style="display:none;">

<input type="number" id="amount" placeholder="Enter amount">

<button onclick="submitTransaction()">Submit</button>

</div>

</div>

<script src="atm.js"></script>

</body>

</html>

## **7.3 CSS CODE**

body {

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

margin: 0;

background-color: #f0f0f0;

font-family: Arial, sans-serif;

}

#atm-container {

background-color: #333;

color: #fff;

border-radius: 10px;

padding: 20px;

width: 300px;

text-align: center;

}

#screen {

background-color: #000;

color: #0f0;

padding: 20px;

border-radius: 5px;

margin-bottom: 20px;

}

#keypad button {

background-color: #444;

color: #fff;

border: none;

border-radius: 5px;

padding: 10px;

font-size: 16px;

margin: 5px;

cursor: pointer;

transition: background-color 0.3s;

}

#keypad button:hover {

background-color: #555;

}

#transaction-form {

margin-top: 20px;

}

#transaction-form input {

padding: 10px;

font-size: 16px;

margin-bottom: 10px;

width: calc(100% - 22px);

}

#transaction-form button {

background-color: #444;

color: #fff;

border: none;

border-radius: 5px;

padding: 10px;

font-size: 16px;

cursor: pointer;

transition: background-color 0.3s;

}

#transaction-form button:hover {

background-color: #555;

}

# **8. OUTPUT SCREENSHOT**

|  |
| --- |
| Figure 8.1 : Login Interface. |

|  |
| --- |
| Figure 8.2 : Entering pin. |

|  |
| --- |
| Figure 8.3 : Pin Accepted. |

|  |
| --- |
| Figure 8.4 : Checking Balance. |

|  |
| --- |
| Figure 8.5 : Deposit Money. |

|  |
| --- |
| Figure 8.6 : Withdraw Money. |

# **9. CONCLUSION**

In conclusion, the optimization of user experiences in ATM simulation systems is a multifaceted challenge that requires a thorough understanding of system architecture, user interactions, and transaction processes. This capstone project has explored these aspects in detail, offering insights into how ATM systems can be enhanced to provide more efficient and user-friendly services. Through a comprehensive analysis of ATM components and their interactions, the study has identified key areas for improvement. The architecture diagram and flowchart have provided a clear understanding of how ATM systems operate, highlighting potential bottlenecks and opportunities for optimization. By examining these elements, the study has proposed solutions to streamline operations, reduce user wait times, and enhance overall system performance. The implementation of Java programming in this project has demonstrated the practical aspects of developing ATM simulation systems. The code snippets presented in the study highlight the importance of robust error handling and security measures, ensuring that transactions are conducted smoothly and securely. The emphasis on secure coding practices underscores the need for vigilance in preventing unauthorized access and maintaining the integrity of financial transactions. In summary, optimizing user experiences in ATM simulation systems is a critical aspect of modern banking technology. By understanding the intricacies of system operations and identifying areas for improvement, this study offers a roadmap for enhancing the efficiency and user-friendliness of ATM systems. The insights gained from this project pave the way for future advancements in banking technology, ensuring that ATM systems remain a vital component of the financial ecosystem.

# **10. REFERENCES**

|  |  |
| --- | --- |
| [1] | Jurado, Raquel Delgado-Aguilera, et al. "Safety and Resilience Bayesian Belief Network (SR-BBN) for ATM." *Journal of Physics: Conference Series*. Vol. 2716. No. 1. IOP Publishing, 2024. |
| [2] | Jurado, Raquel Delgado-Aguilera, Víctor Fernando Gómez Comendador, María Zamarreño Suárez, Francisco Pérez Moreno, Christian Eduardo Verdonk Gallego, and Rosa María Arnaldo Valdés. "Safety and Resilience Bayesian Belief Network (SR-BBN) for ATM." In *Journal of Physics: Conference Series*, vol. 2716, no. 1, p. 0120[(Gattani 2021)](https://paperpile.com/c/EiYii2/15cB)78. IOP Publishing, 2024. |
| [3] | Jurado, R.D.A., Comendador, V.F.G., Suárez, M.Z., Moreno, F.P., Gallego, C.E.V. and Valdés, R.M.A., 2024, March. Safety and Resilience Bayesian Belief Network (SR-BBN) for ATM. In *Journal of Physics: Conference Series* (Vol. 2716, No. 1, p. 012078). IOP Publishing. |
| [4] | Jurado RD, Comendador VF, Suárez MZ, Moreno FP, Gallego CE, Valdés RM. Safety and Resilience Bayesian Belief Network (SR-BBN) for ATM. InJournal of Physics: Conference Series 2024 Mar 1 (Vol. 2716, No. 1, p. 012078). IOP Publishing. |
| [5] | Zhu, Junda, et al. "ATM: Adversarial Tuning Multi-agent System Makes a Robust Retrieval-Augmented Generator." *arXiv preprint arXiv:2405.18111* (2024). |
| [6] | Zhu, J., Yan, L., Shi, H., Yin, D., & Sha, L. (2024). ATM: Adversarial Tuning Multi-agent System Makes a Robust Retrieval-Augmented Generator. *arXiv preprint arXiv:2405.18111*. |
| [7] | Zhu, Junda, Lingyong Yan, Haibo Shi, Dawei Yin, and Lei Sha. "ATM: Adversarial Tuning Multi-agent System Makes a Robust Retrieval-Augmented Generator." *arXiv preprint arXiv:2405.18111* (2024). |
| [8] | Zhu J, Yan L, Shi H, Yin D, Sha L. ATM: Adversarial Tuning Multi-agent System Makes a Robust Retrieval-Augmented Generator. arXiv preprint arXiv:2405.18111. 2024 May 28. |
| [9] | Holubnychyi, Oleksii, et al. "Self-organization Technique with a Norm Transformation Based Filtering for Sustainable Infocommunications Within CNS/ATM Systems." *International Workshop on Advances in Civil Aviation Systems Development*. Cham: Springer Nature Switzerland, 2024. |
| [10] | Holubnychyi, O., Zaliskyi, M., Ostroumov, I., Sushchenko, O., Solomentsev, O., Averyanova, Y., ... & Kuznetsov, B. (2024, March). Self-organization Technique with a Norm Transformation Based Filtering for Sustainable Infocommunications Within CNS/ATM Systems. In *International Workshop on Advances in Civil Aviation Systems Development* (pp. 262-278). Cham: Springer Nature Switzerland. |
| [11] | Holubnychyi, Oleksii, Maksym Zaliskyi, Ivan Ostroumov, Olha Sushchenko, Oleksandr Solomentsev, Yuliya Averyanova, Yurii Bezkorovainyi et al. "Self-organization Technique with a Norm Transformation Based Filtering for Sustainable Infocommunications Within CNS/ATM Systems." In *International Workshop on Advances in Civil Aviation Systems Development*, pp. 262-278. Cham: Springer Nature Switzerland, 2024. |
| [12] | Holubnychyi, O., Zaliskyi, M., Ostroumov, I., Sushchenko, O., Solomentsev, O., Averyanova, Y., Bezkorovainyi, Y., Sokolova, O., Voliansky, R., Bovdui, I. and Ivannikova, V., 2024, March. Self-organization Technique with a Norm Transformation Based Filtering for Sustainable Infocommunications Within CNS/ATM Systems. In *International Workshop on Advances in Civil Aviation Systems Development* (pp. 262-278). Cham: Springer Nature Switzerland. |
| [13] | Holubnychyi O, Zaliskyi M, Ostroumov I, Sushchenko O, Solomentsev O, Averyanova Y, Bezkorovainyi Y, Sokolova O, Voliansky R, Bovdui I, Ivannikova V. Self-organization Technique with a Norm Transformation Based Filtering for Sustainable Infocommunications Within CNS/ATM Systems. InInternational Workshop on Advances in Civil Aviation Systems Development 2024 Mar 26 (pp. 262-278). Cham: Springer Nature Switzerland. |
| [14] | Kolotusha, Volodymyr, Tetiana Shmelova, and Dmytriy Bondarev. "Multi-criterion Evaluation of the Criteria of the Information Model Conformity of the Air Traffic Controller Simulator to the Real System." *International Workshop on Advances in Civil Aviation Systems Development*. Cham: Springer Nature Switzerland, 2024. |
| [15] | Kolotusha, V., Shmelova, T., & Bondarev, D. (2024, March). Multi-criterion Evaluation of the Criteria of the Information Model Conformity of the Air Traffic Controller Simulator to the Real System. In *International Workshop on Advances in Civil Aviation Systems Development* (pp. 364-384). Cham: Springer Nature Switzerland. |