

ARCHITECTURAL FRAMEWORK FOR DEVELOPING A TRAVEL AGENCY INFORMATION SYSTEM

Dragana Knežević¹ Mladen Stojić² Milorad Murić³

Abstract: The growing complexity of operations in the tourism industry demands efficient and scalable information systems. The entire process can be divided into two phases: (1) creating a plan (draft) for the application and analyzing the needs and functionalities, so-called the pre-development phase, and (2) the application development phase. This paper is the first of two that presents the development of an application—an information system tailored to the needs of a travel agency, focusing on its architectural framework. The focus is on the pre-development phase, including system requirements analysis, business process modeling, and the definition of use cases. Key diagrams, such as use case, class, and sequence diagrams, are utilized to capture functional and non-functional requirements, ensuring seamless alignment with business goals. The proposed framework emphasizes modularity, data integrity, and scalability, providing a structured approach for future implementation. This research offers a foundation for the subsequent development of a WPF application that will enhance service delivery and operational efficiency in travel agencies.

Key words: UML, System design, Travel agency software, Software architecture.

1. INTRODUCTION

The modern tourism sector requires comprehensive information systems to manage bookings, customers, and offers in real time. Designing an information system for a travel agency involves several critical stages, including requirements gathering, business process modeling, and system architecture design. This paper focuses on developing an architectural framework that ensures modularity, scalability, and alignment with the agency's operational needs. The framework will guide future development efforts, with a particular focus on implementing a WPF application. Various diagrams and models will be used to illustrate the system's structure and interactions, ensuring clarity and coherence throughout the development process.

The goal of this paper is to prepare all necessary pre-development steps, laying a solid foundation for the eventual coding phase. By identifying functional and non-functional requirements and visualizing processes through diagrams, the framework will streamline the software's development, ensuring a robust and efficient system for travel agency operations.

2. RESEARCH LANDSCAPE AND RELEVANT APPLICATIONS

In the field of software development for the travel and tourism industry, several studies and projects have addressed the challenges of designing efficient, scalable, and user-friendly information systems. These systems often aim to automate complex processes, such as booking management, customer data handling, and payment processing, while providing a seamless user experience.

One such example is the work by Tesoriero and Mohamad [1], who explored the application of Business Process Modeling Notation (BPMN) and Unified Modeling Language (UML) to design travel agency systems. Their research highlighted the importance of visualizing business processes and system interactions, aligning closely with the approach taken in this paper to utilize use case diagrams, class diagrams, and sequence diagrams for system design. This study demonstrates the utility of UML in documenting and communicating both high-level and detailed system requirements for travel agencies, making it a valuable reference for the current work.

Another notable study is that of Zhao and Stojanovic [2], who performed a comparative analysis of web-based and client-server systems within the travel and tourism industry. Their research explores

¹PhD, Western Serbia Academy of Applied Studies – Užice Department, Užice, Serbia, email: dragana.knezevic@vpts.edu.rs

²BSc, Western Serbia Academy of Applied Studies – Užice Department, Užice, Serbia, email: mladen.stojic@vpts.edu.rs

³MSc, Western Serbia Academy of Applied Studies – Užice Department, Užice, Serbia, email: milorad.muric@vpts.edu.rs

the scalability and performance challenges associated with these systems, offering insights that are highly relevant to the architectural framework outlined in this paper. As travel agencies continue to adopt web-based solutions, evaluating the technical trade-offs and benefits of various system architectures provides a valuable basis for making informed decisions on deployment and scalability strategies.

Beyond domain-specific applications, several seminal works have established UML as a foundational tool for system design. Booch, Rumbaugh, and Jacobson [3] provide a comprehensive guide to UML, offering detailed explanations of all diagram types and their practical applications. Fowler [4] complements this with a concise introduction to UML, focusing on the most commonly used diagrams, making it a useful reference for the preliminary stages of system modeling. Similarly, Larman [5] bridges UML with object-oriented analysis and agile development practices, providing practical examples that are relevant to contemporary system design projects.

In terms of database design, Bertino et al. [6] discusses key considerations in database security, a crucial aspect when handling sensitive data such as customer information and payment details. Their work serves as a basis for ensuring the proposed system adheres to best practices in data security and privacy protection, aligning with the focus on securing the travel agency's data within the pre-development phase of this project.

MacDonald [7] provides an in-depth exploration of business process modeling techniques, emphasizing the importance of detailed activity diagrams to represent complex workflows in organizations. His research highlights the significance of modeling as a means of enhancing operational efficiency—a main goal of the system described in this paper. This perspective supports the methodology employed here, where business process diagrams are used to model the agency's critical operations, such as bookings and payments.

These studies, coupled with the foundational literature on UML and modern database practices, significantly contribute to the development of information systems in the travel sector. The current research builds upon these works, incorporating modern software engineering principles and practices to create a robust, efficient, and scalable information system tailored to the specific needs of a travel agency.

Furthermore, the Entity-Relationship Model (ER Model) serves as a foundational data modeling technique for representing entities, attributes, and relationships within a database [8]. An in-depth understanding of ER modeling principles is crucial for designing an efficient and scalable database schema for the travel agency system.

Software development methodologies, as outlined by Pressman [9], play a vital role in structuring the development process. These methodologies provide a framework for various activities, such as requirement gathering, system design, implementation, testing, and deployment. By adhering to established methodologies, the development team can ensure a well-defined and controlled approach to building the travel agency system.

Object-oriented design patterns, as documented by Gamma et al. [10], offer reusable solutions to common software design problems. Integrating these patterns into the system's architecture can promote code reusability, maintainability, and overall code quality.

Chapin [11] describes software architecture as the foundational blueprint that outlines a system's structure, including its components, their interactions, and the principles governing these interactions. A thoughtfully designed architecture ensures the travel agency system remains modular, scalable, and easy to maintain over time.

When considering the user interface (UI) development, technologies like Windows Presentation Foundation (WPF) [12] from Microsoft offer a robust framework for building visually appealing and user-friendly interfaces. WPF's capabilities in data binding, layout management, and animation can contribute to a positive user experience within the travel agency system.

3. MAIN POINTS AND FOCUS AREAS

The pre-development phase consists of a comprehensive set of sub-phases aimed at establishing a solid foundation for system development. It begins with the analysis of requirements, where both functional needs—such as booking management, customer profiles, and payment processing—and non-functional needs, like performance, security, and scalability, are identified and documented. Stakeholders, including agents, administrators, and customers, are defined to ensure their roles and expectations are clearly addressed. Feasibility studies are then conducted to evaluate the technical, financial, and operational viability of the project, followed by conceptualization, where high-level ideas are organized into a structured concept, often paired with a preliminary system design.

This phase also includes use case modeling, where interactions between users and the system are visualized through diagrams and scenarios, such as creating bookings, updating travel offers, and managing user roles. Business process modeling further refines workflows by mapping key operational flows, like booking and payment processes, to enhance efficiency. System architecture design focuses on defining core components, such as classes, attributes, and their relationships, while also visualizing interactions and deployment strategies, whether on-premises or in the cloud. Behavioral modeling deepens the understanding of system dynamics by mapping interactions and object lifecycles through sequence and state diagrams.

Database design ensures robust data management by modeling schemas, establishing relationships, and normalizing data to maintain integrity and eliminate redundancy. Security and access control measures are planned, detailing roles, permissions, and mechanisms for authentication and authorization to safeguard sensitive information. Finally, strategies for scalability and maintainability are outlined, emphasizing modular design, cloud integration, and adaptability to accommodate future business needs, providing a clear roadmap for the subsequent development phases.

4. VISUAL MODELING USING UML: REPRESENTING SYSTEM FLEXIBILITY

UML is a language used to describe the functionality of a system and the interactions between its elements and participants through a set of strict definitions established by the Object Management Group [9, 10]. The first version of UML was developed in 1994 [11]. Diagrams serve as the primary tool for representing a system's structure [12]. Composed of graphical elements, they are designed to describe specific components or functionalities within the system. Since no single diagram can capture the entirety of a system, multiple diagrams are used to illustrate its various aspects.

This section showcases the key visual models created during the pre-development phase, which serve as a blueprint for understanding and designing the system. These models play a pivotal role in translating business requirements into a structured and detailed technical framework. The case study in this paper focuses on the travel agency "MS". The agency's offerings include organizing tours to the world's largest cities, such as Rome, Venice, Milan, Paris, Madrid, Vienna, Istanbul, Munich, Amsterdam, Lisbon, Moscow, and others. It organizes packages for the summer and winter seasons, cruises, and excursions, as well as both group and individual programs. The agency offers a wide variety of travel options – from classic one-day and multi-day trips to family vacations. The main business processes of the agency are:

1. Client interactions:

- Presenting the offers to the client,
- Signing the contract,
- Collecting payments from the client,
- Managing the reservation.

2. Record keeping:

- Keeping records of available packages, adding new ones, modifying, and removing existing packages from the offer,

- Keeping records of tourists, collecting their data for organizing trips,
- Keeping records of completed trips

4.1. Defining the System Requirements

First of all, we need to define the main participants in the system. In this case, these are: Client – Tourist, User – Travel Agent, and System Administrator. The main functionalities for these participants are:

Client:

- Visits the agency and reviews the offers,
- If interested in an offer, asks the travel agent to make a reservation and sign the contract,
- Makes payment immediately on-site or before the trip departs,
- If the client changes their mind, they can request the cancellation of the reservation.

Travel Agent:

- Keeps records, adds new offers, and modifies existing ones,
- Creates reservations and signs contracts with clients,
- Maintains records of all reservations,
- Sends requests to the administrator to create a new account or modify an existing one.

Administrator:

- Reviews, adds, and modifies application users.

The real project consists of several diagrams describing such a system. Below, only some of them will be presented, starting with the first of the agency's two main business processes: Client interactions.

4.2. System Design Insights: UML Diagrams for Architecture and Dynamics

4.2.1. Class and Object Diagram:

The Class Diagram defines the system's structure by outlining its primary entities (Figure 1), such as: Contract, Client, Employee, Reservation, Tourist, Administrator, Agent, and Destination. A client has attributes such as first name and last name. A client can sign one or more contracts. The employee class contains two subclasses: Administrator and Agent. An employee (agent) can sign one or more contracts.

A contract can contain only one travel package. A travel package can contain only one destination. Multiple tourists can be registered for a single travel package. A client can reserve a trip for themselves, but they are not required to.

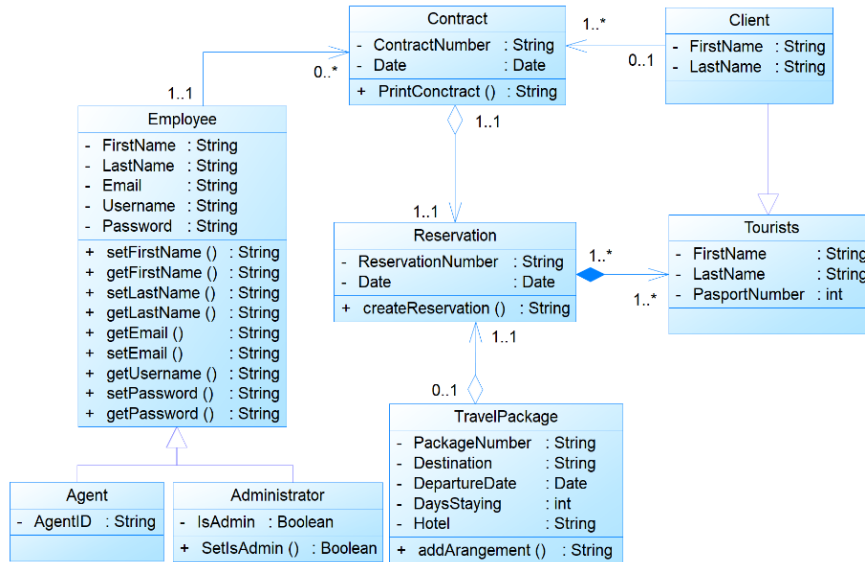


Figure 1. – A class diagram for MS Travel Agency (Source: Author)

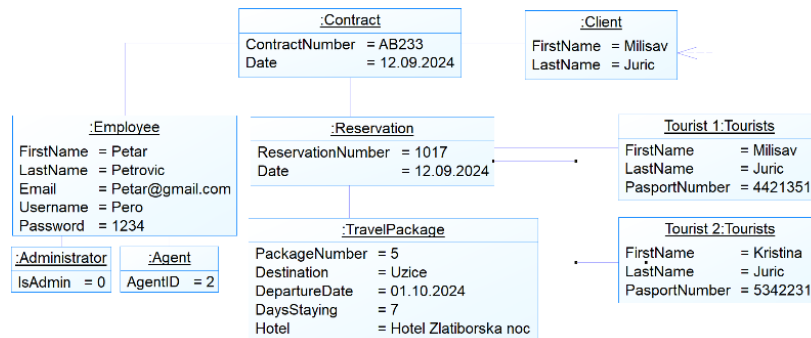


Figure 2. – An object diagram for MS Travel Agency (Source: Author)

Similarly, we can create an object diagram (Figure 2) that illustrates a specific example of client interaction with the travel agency. Client *Milisav Jurić* signed a contract on *September 9, 2024*, under number *AB233*, thereby booking a trip for two passengers. He booked a trip to *Užice* on *October 1, 2024*, staying at the *Hotel Zlatiborska noć* for a duration of *7 days*. The contract was prepared by the travel agent *Petar Petrović*.

4.2.2. State Diagram

A State Diagram represents the dynamic behavior of a system by modeling the possible states of an object and the transitions between those states based on events or conditions. In the case of the travel agency, state diagrams can be used to depict different behaviors of the system during various business processes, such as booking new reservations, managing the catalog, or managing employee accounts. The state diagram (Figure 3) illustrates the client's behavior when selecting and booking a destination. Initially, the client reviews the available destinations in the catalog. If no destination is selected, the client exits the booking process. Subsequently, the system checks the availability of the chosen destination. If unavailable, the process reverts to the initial step, allowing the client to select another destination or exit. Conversely, if the destination is available, a reservation request is submitted. The system then validates the entered data. If errors are detected, the client is prompted to correct the data and resubmit. Upon successful data validation, the reservation is confirmed, and the payment process is initiated.

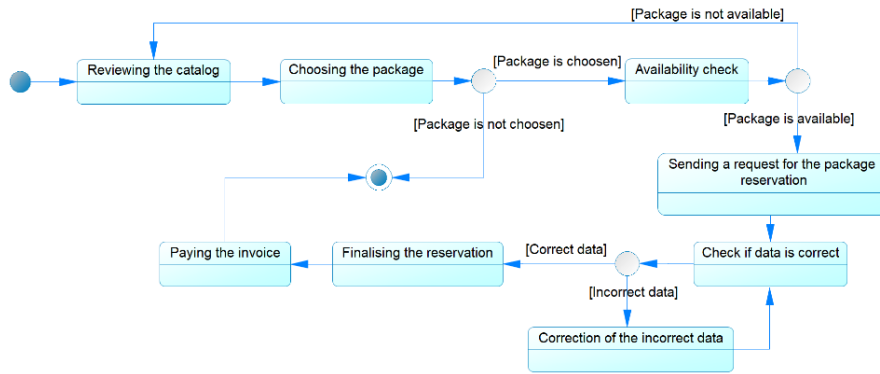


Figure 3. – A State diagram representing the client interactions with the System (Source: Author)

4.2.3. Use Case Diagram

The Use Case Diagram offers a high-level overview of the system’s interactions with users, including agents, administrators, and customers. It highlights core functionalities such as booking management, updating travel offers, and user role administration, ensuring all essential operations are captured.

The following diagram (Figure 4) demonstrates the client’s interaction with the system. Clients can browse the travel agency's catalog in search of a suitable destination. If a client decides on a destination, they can request the creation of a reservation. Creating a reservation may involve selecting a destination, departure date, and the number of days for the stay. Creating a reservation requires that the agent be logged into the system so that the system can process the reservation. The agent signs a contract with the client based on the created reservation. In addition to the contract, the client can pay for the trip on the spot. The client can also request that the reservation be canceled. In such a case, if the trip has already been paid for, a refund will be issued.

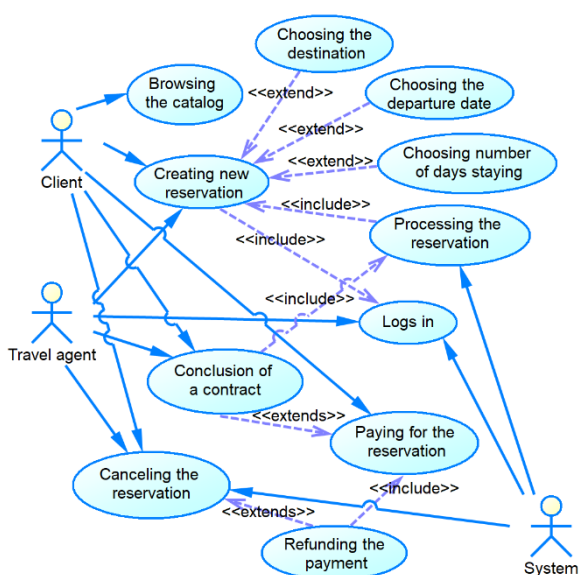


Figure 4. – A use case diagram - the client’s interaction with the System (Source: Author)

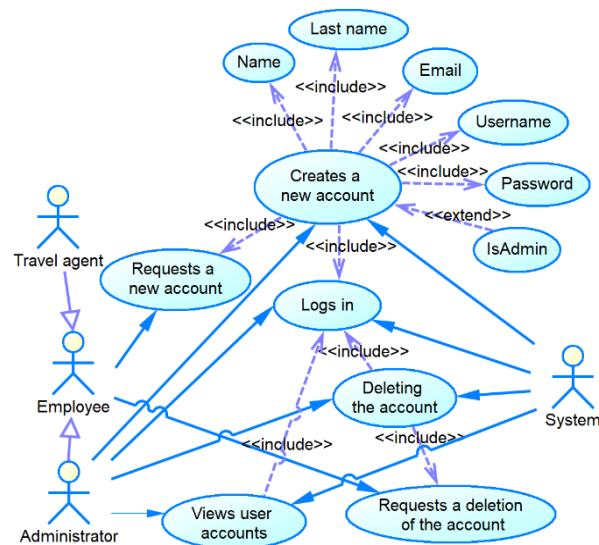


Figure 5. – A use case diagram demonstrating the administrator’s interaction with the System (Source: Author)

Figure 5 shows another very important use case diagram, demonstrating the administrator’s interaction with the system. Existing or new employees can request the creation of a new account through which they can access the system. The actor who manages the accounts is the administrator. Since this involves the administrator, it is necessary for the administrator to be logged into the system in order to create a new system user. Once logged in, the administrator performs the account creation procedure based on the request.

Account creation involves entering the user's details, such as first name, last name, email, username, and password. Based on these parameters, a new travel agent's account is created.

The administrator can also create accounts for other administrators. In addition to the previously mentioned parameters, the administrator can designate the new or existing account as an administrator account. Besides requesting the creation of a new account, employees can also request the deletion of their accounts. The administrator can view the list and basic details of all registered users in the system.

4.2.4. Activity Diagram

The Activity Diagram illustrates workflows of critical business processes, such as booking and employee's account management. By visualizing these processes step-by-step, it identifies opportunities for optimization and ensures smooth execution. Specifically, key stages in the booking process, like the travel package selection, a validation of client information, reservation confirmation, and payment, are detailed, enabling the system to operate both efficiently and logically.

The following example (Figure 6) presents an Activity Diagram that models a client’s interaction with the system during the booking process.

The workflow starts when the client requests a catalog of destinations from the agent. The agent provides the catalog, and the client begins searching for a desired destination. If no destination is chosen, the activity ends. Otherwise, the client submits a reservation request to the agent. The agent then reviews the request and verifies the correctness of the submitted information. If the information is incorrect, the agent asks the client to revise it. Upon receiving accurate data, the agent processes the reservation and enters it into the system. After entering the reservation, the agent prints the contract and hands it to the client for signing. Once the client signs the contract, the payment process is initiated, completing the workflow.

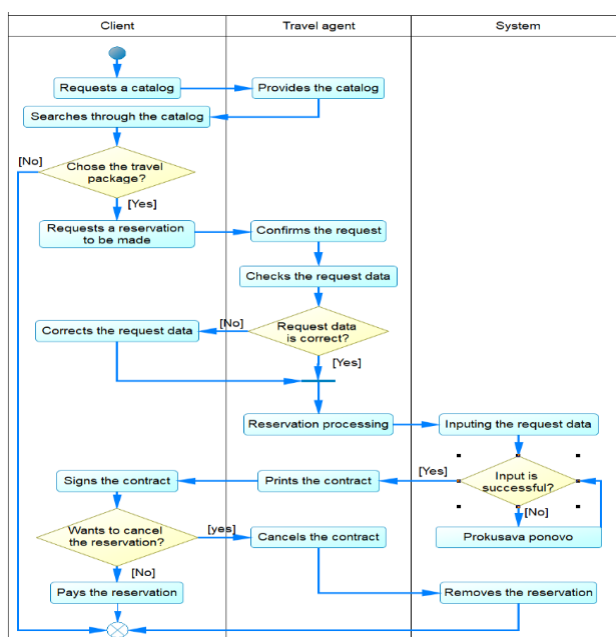


Figure 6. – An activity diagram that shows possible interactions between the client and the System (Source: Author)

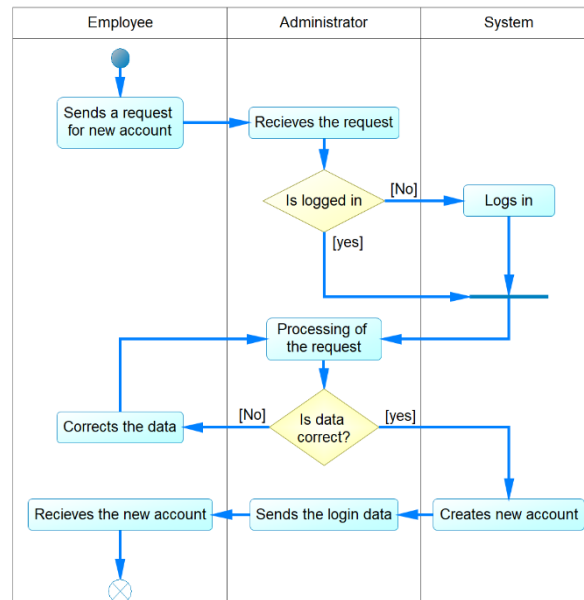


Figure 7. – An activity diagram that shows the employee’s interactions with the System during account creation (Source: Author)

Figure 7 represents an activity diagram that shows the employee’s interaction with the system during account creation. The employee submits a request for the creation of a new account and forwards it to the administrator. The request contains the data required to create a new account. The administrator must be logged in to create a new account, and the system checks if they are logged in. If not, the administrator will log in.

The administrator verifies the correctness of the information in the request; if it is correct, the new user is registered, and the login details are sent to them. Otherwise, the administrator asks the employee to correct the errors and verifies the correctness again.

4.2.5. Sequence Diagram

The Sequence Diagram focuses on the dynamic behavior of the system by modeling the sequence of interactions between its components. Figure 8 represents the order of events in case the client decides to make a reservation for a trip to a specific destination. The client requests the catalog of travel packages from the travel agent. The agent presents the agency's offer.

When the client decides on a destination, they ask the agent to make a reservation. The agent then enters the details of the selected destination in the system. Once the destination details are entered, the agent requests the details of the passengers who will travel to the selected destination. The data for each passenger is entered through a loop.

Once the passenger data entry is complete, the agent creates a contract based on the previously entered information, which is printed and forwarded to the client. The client signs the contract on-site and pays the reservation costs. Afterward, the client receives a copy of the contract and a receipt for the transaction.

Figure 9 shows the second sequence diagram representing a sequence of events during account creation. The employee sends a request to the administrator to create a new account. The administrator responds that the request has been accepted. In order to create a new account, the administrator must be logged in. The administrator attempts to log in, and if the login is unsuccessful, they will continue attempting until they are logged in. The employee sends the registration details. After logging in, the administrator verifies the information. If the data is incorrect, the employee is asked to correct it. With the correct data, the administrator completes the registration. Upon finishing the registration, the administrator forwards the login details to the employee.

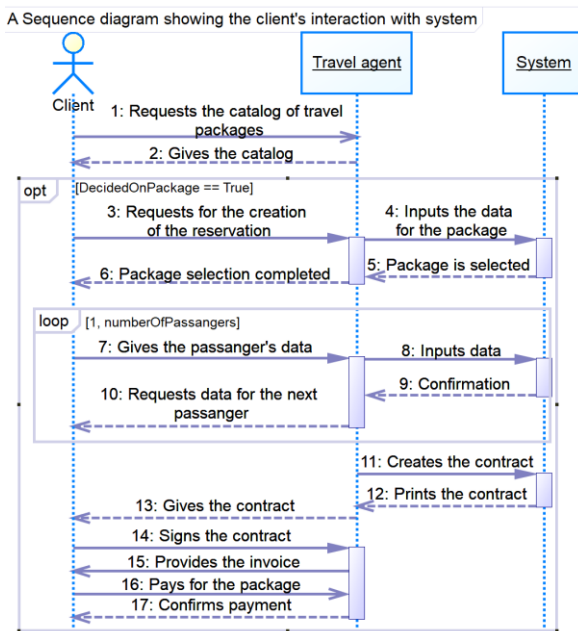


Figure 8. – A sequence diagram showing the sequence of the client's interactions with the System (Source: Author)

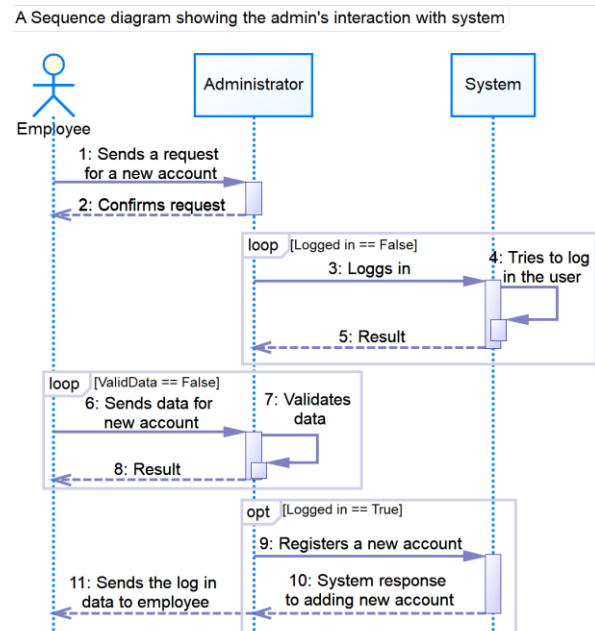


Figure 9. – A sequence diagram showing sequences during the creation of the new account (Source: Author)

5. CONCLUSION

The paper concludes with a reflection on the importance of proper planning and architectural design in software development. The preparation outlined in this framework ensures that the future WPF application will be robust, user-friendly, and capable of meeting the dynamic needs of a travel agency. Further research will focus on the actual development and implementation of the WPF solution, building upon the foundation established by this architectural framework.

The presented visual models provide a detailed representation of the system's structural and behavioral aspects. These diagrams are essential for bridging the gap between business requirements and technical implementation, ensuring that the system design aligns with the goals and needs of the travel agency. By capturing both high-level interactions and intricate processes, the models foster a shared understanding among stakeholders and developers. This visual framework lays a solid foundation for the development phase, offering clarity, consistency, and scalability as the project progresses toward implementation.

Collectively, these visual models offer a comprehensive guide for stakeholders and developers, outlining the system's structure, behavior, and architecture. This will help inform the development phase and ensure that all parties are aligned.

In the following paper, the application developed using C# and XAML code (WPF application) will be described, detailing the implementation of the system based on the architectural framework established in this phase.

6. REFERENCES

- [1] Tesoriero, R.; Mohamad, M.: *Business Process Modeling Notation (BPMN) and UML: The Case of a Travel Agency*, International Journal of Business Information Systems, vol. 7, no. 4, 2011, pp. 319–332.
- [2] Zhao, X.; Stojanovic, A.: *A Comparative Analysis of Web-Based and Client-Server Systems in Tourism*, International Journal of Software Engineering and Applications, vol. 10, no. 3, 2017, pp. 45–60.

- [3] Booch, G.; Rumbaugh, J.; Jacobson, I.: *The Unified Modeling Language User Guide*. Addison-Wesley, 1999.
- [4] Fowler, M.: *UML Distilled: A Brief Guide to the Standard Object Modeling Language*, 3rd ed. Addison-Wesley, 2003.
- [5] Larman, C.: *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process*. Prentice Hall, 2004.
- [6] Bertino, E.; Sandhu, R.; Thuraisingham, B.: *Database Security: Concepts, Approaches, and Challenges*, IEEE Transactions on Software Engineering, vol. 31, no. 9, 2005, pp. 749–761.
- [7] MacDonald, C.: *Modeling Business Processes: A Petri Net-Oriented Approach*, Springer, 2012.
- [8] Oracle Documentation, *Entity-Relationship Model (ER Model)*, 2023. [Online]. Available: https://docs.oracle.com/cd/B28359_01/server.111/b28310/datamodel.htm, Accessed: January 2024.
- [9] Pressman, R. S.: *Software Engineering: A Practitioner's Approach*, 9th ed. McGraw-Hill Education, 2014.
- [10] Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J.: *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1994.
- [11] Chapin, R.: *Software Architecture for Developers: Volume 1 - Technical Leadership and the Software Architecture Practice*, Leanpub, 2010.
- [12] Microsoft Documentation, *Windows Presentation Foundation (WPF)*, 2023. [Online]. Available: <https://learn.microsoft.com/en-us/dotnet/desktop/wpf/>, Accessed: January 2024.