Monopoly assistance

ElHadi73

Development report and documentation

ElHadi73
Mars 28, 2023

Contents

1 Introduction

The classic board game Monopoly has been enjoyed by millions of people of all ages for over eight decades. It is a game that involves buying, selling, and trading properties with the objective of becoming the wealthiest player. Monopoly has brought joy to countless families and friends over the years. However, the game's reliance on paper money and too many physical game pieces can make it cumbersome and time-consuming to play.

Monopoly Assistantante is a digital app designed to enhance and simplify the gameplay experience of the classic board game Monopoly for an efficient and convenient way to play the game by providing a digital system for managing players money and properties, reducing the amount of paper used during gameplay, and tracking game progress in real-time, and make it less time-consuming, with customizable rules featuer with multiplayer support, and try to achive that without sacrificing the social and fun aspects of playing in person.

These features aim to simplify the gameplay experience for players while also enhancing it by providing real-time updates, detailed statistics, and customizable options. In addition, the app's digital tracking capabilities allow players to easily view their progress throughout the game, including their assets, properties, and overall net worth. This provides players with a clear understanding of their current position and helps them make more informed gameplay decisions. By providing players with a comprehensive overview of the game, Monopoly Assistantante can also help players develop their strategic thinking and decision-making skill, which can lead to a more immersive and engaging experience, where players can fully explore the strategic depth of the game and gain valuable experience in theory.

The thesis will evaluate the effectiveness of the app in improving the Monopoly gameplay experience, with a focus on how it impacts gameplay mechanics, and the flow and speed of the game, facilitates more strategic gameplay, player interaction, and overall enjoyment, as well as to identify areas for potential future development.

It is important to note that while the app is designed to reduce the amount of paper used and make gameplay more efficient, it is not intended to replace the physical presence of the players and the board game itself. The fun and social aspects of playing Monopoly in person are an essential part of the game's appeal, and the app is designed to complement and enhance these aspects, rather than replace them.

The thesis will begin by reviewing the literature on board games and digital technology, as well as discussing the design and development process of the app. Next, the thesis will present the results of user testing and surveys to evaluate the app's usability and user experience. Finally, the thesis will conclude with a discussion of the findings and potential directions for future research in this area. Through an analysis of Monopoly Assistantante's features and impact on gameplay, this thesis aims to explore the ways in which digital tools can enhance and modernize traditional board game experiences. By examining the app's impact on gameplay mechanics, player interaction, and enjoyment, this thesis seeks to provide valuable insights into the role of technology in transforming classic board games into efficient and engaging digital experiences.

2 Problematic

Playing monopoly is really fun, but it take a lot of time, which causes many of your friends refuses to play it, that what will make playing it not fun. one of the things that causes it to take very long is the money transactions between players, or with the bank/goverment, espacaly when u dont have the change to pay the right amount of money.

3 Objectives

in this web app we will minimize the time it takes to make a transaction between two players or between a player and the bank/goverment, where the transactions going to be made digitly, and the money holding will be virtually.

4 System Requirements and Actors

4.1 Introduction

To make sure that the web application replaces the paper money its full uses in the game, it is important to identify the diffrent user of the system

4.2 User Actors

4.2.1 Admin

An external user who is an administrator of the system and can access a set of use cases related to managing the game

4.2.2 Guest

An external user who is not a registered player or administrator and can access a limited set of use cases

4.2.3 Player

An external user who is a registered player and can access a set of use cases related to playing the game

4.3 Static Context Diagram

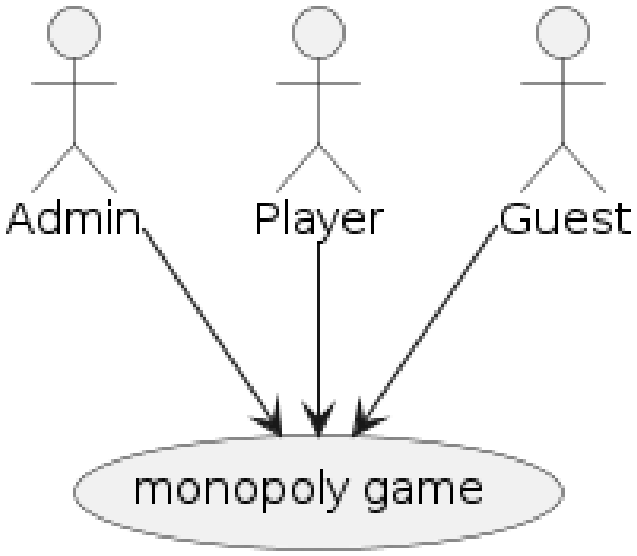

Figure 1: Static Context Diagram

4.4 Functional Requirements

4.4.1 Global

-The system will be capable of handling all money transactions between all element of the game according to the game rules -Each game will have an administrator -A player or Admin must go through guest fase

4.4.2 Guest

-Guest Can choose between hosting a new game or joining an already started game

4.4.3 Admin

-Can initiate transactions of money between players and the Bank

4.4.4 Player

-Can initiate money transaction to other entities of the game

4.5 Non-Functional Requirements

4.5.1 Game rules

-All transaction must be logged as histroy of transacion -The histroy of transacion must be available of all players to see -Any money transaction must be accepted or refused by admin of the game before it is made -In futur Version all game rules can be customizable by Admin of the game -All transaction must be valudated first by admin -The admin is not the same person, but in futur version the admin can switch between player and admin roles -All user must loggin to use the system -The UI/UX must be responsive

4.5.2 Security

-Authentication and authorization: The website must have a robust authentication and authorization mechansime in place to ensure that only authorized users can access there data or perform critical actions that they are allowed to do. -Protection against SqlInjections Attack

4.5.3 Usability

4.5.4 Performance

4.5.5 Relability

4.5.6 Scalability

4.5.7 Maintainability

4.5.8 Compatibility

4.6 Conclusion

5 understanding the monopoly game

5.1 the money flow in monopoly

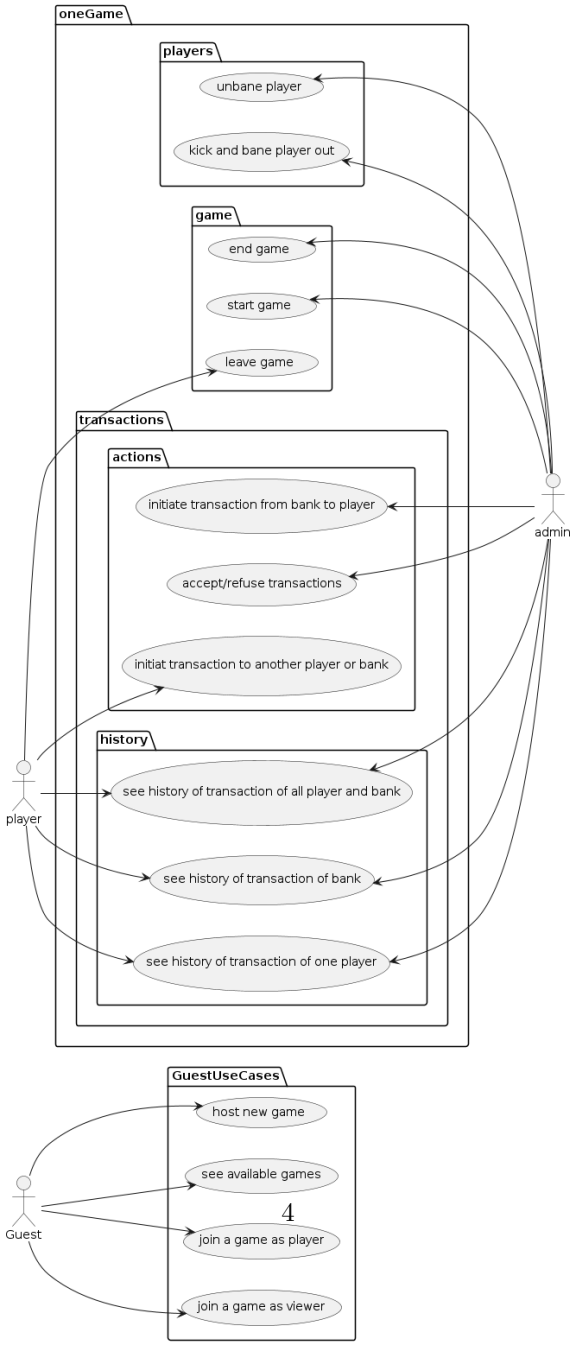

Figure 2: Global Usecase Diagram

6 Architecture and Design

6.1 Introduction

6.2 Global Usecase Diagram

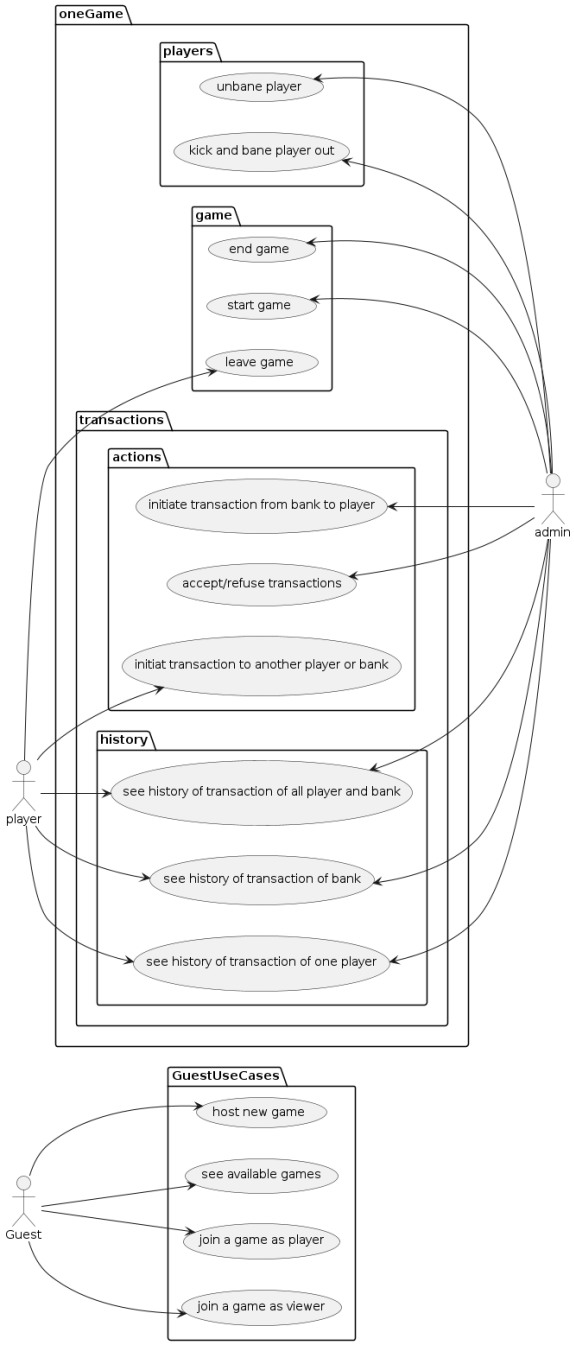

Figure 3: Global Usecase Diagram

6.2.1 Guest Usecase Diagram

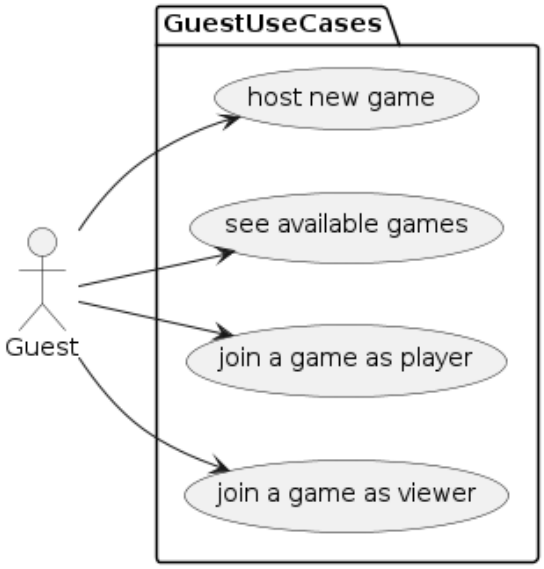

Figure 4: Guest Usecase Diagram

6.2.2 Admin Usecase Diagram

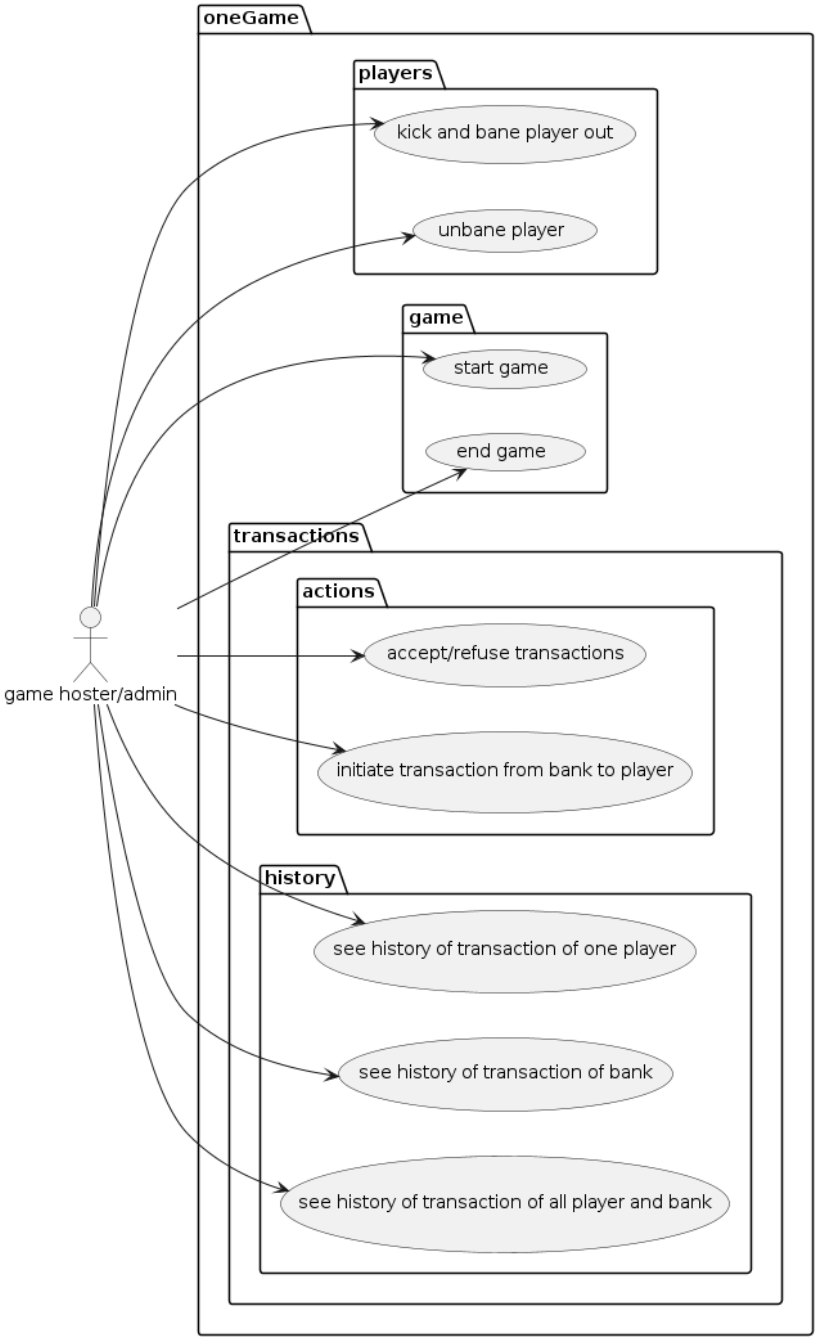

Figure 5: Admin Usecase Diagram

6.2.3 Player Usecase Diagram

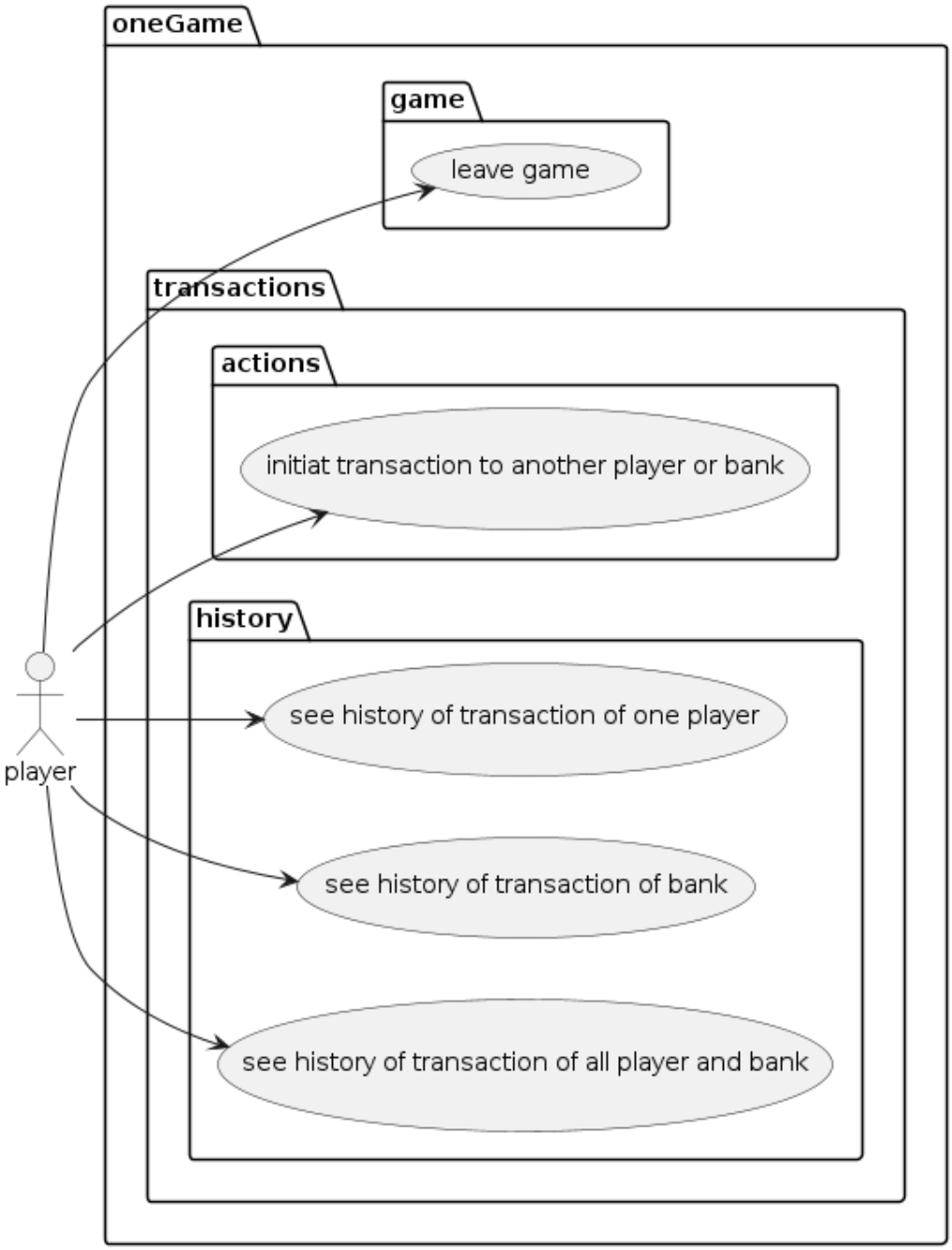

Figure 6: Player Usecase Diagram

6.3 Usecases Study

6.3.1 Host new game by the guest actor usecase

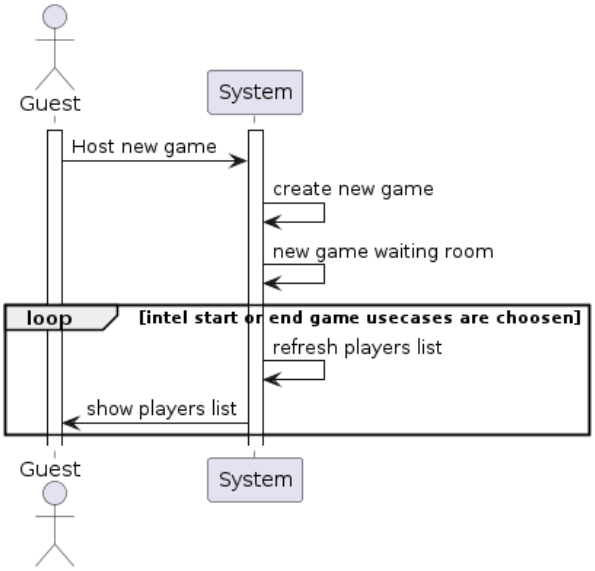

Figure 7: Host new game by the guest actor Sequence Diagram

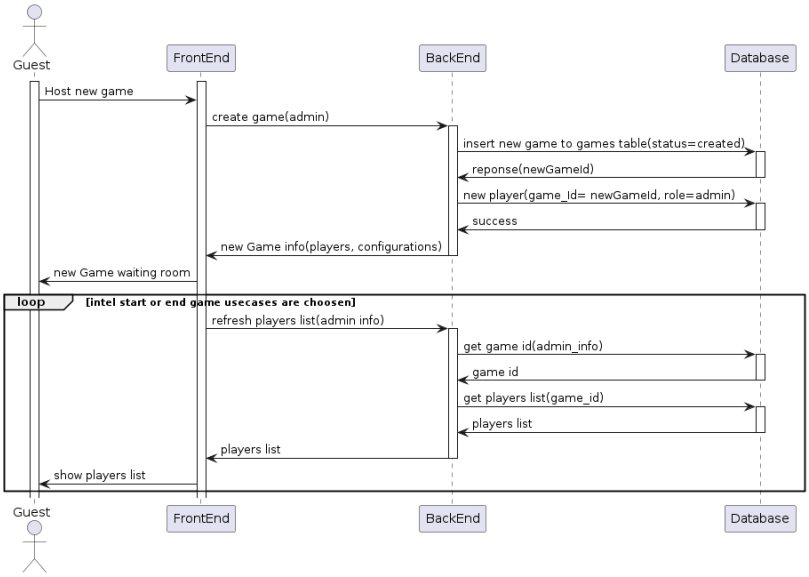

Figure 8: Host new game by the guest actor detailed Sequence Diagram

6.3.2 show available games by the guest actor usecase

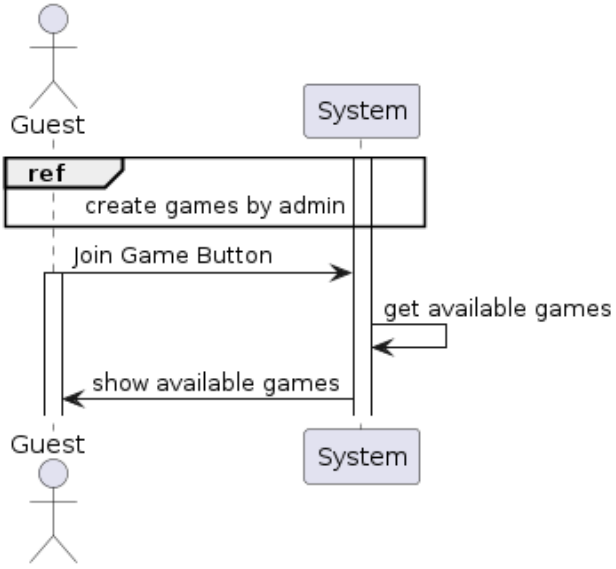

Figure 9: Show available games by the guest actor Sequence Diagram

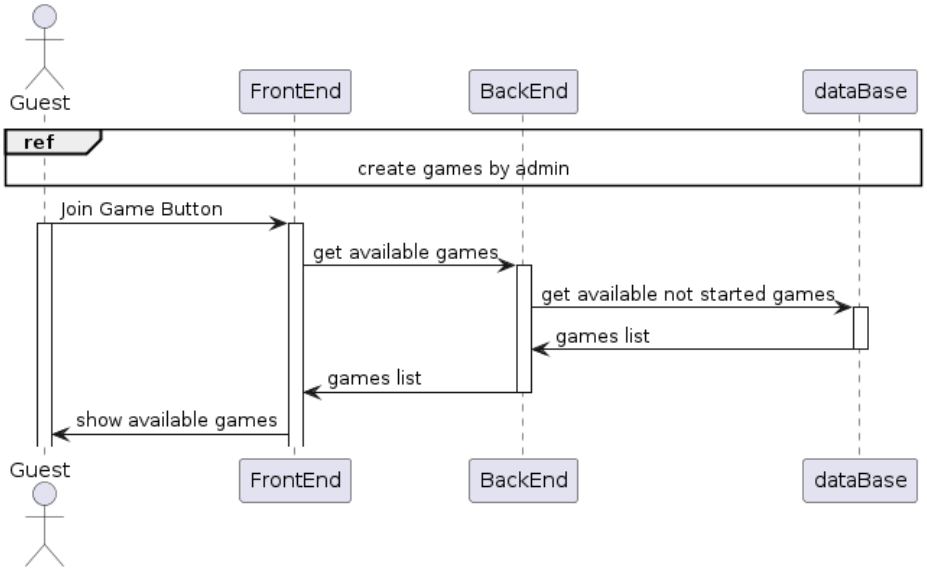

Figure 10: Show available games by the guest actor detailed Sequence Diagram

6.3.3 join game by the guest actor usecase

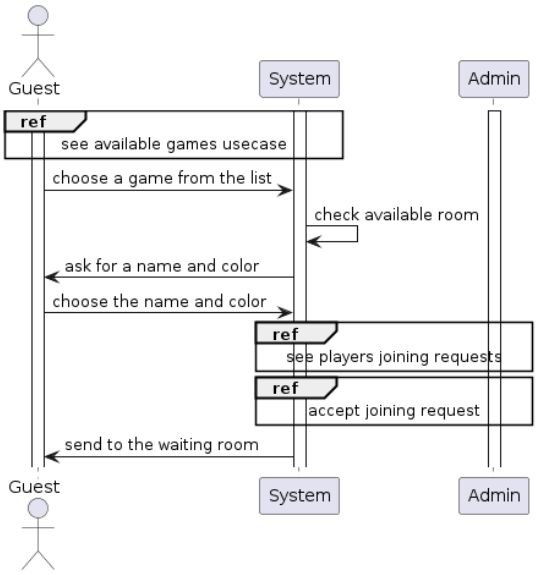

Figure 11: Join game by the guest actor Sequence Diagram

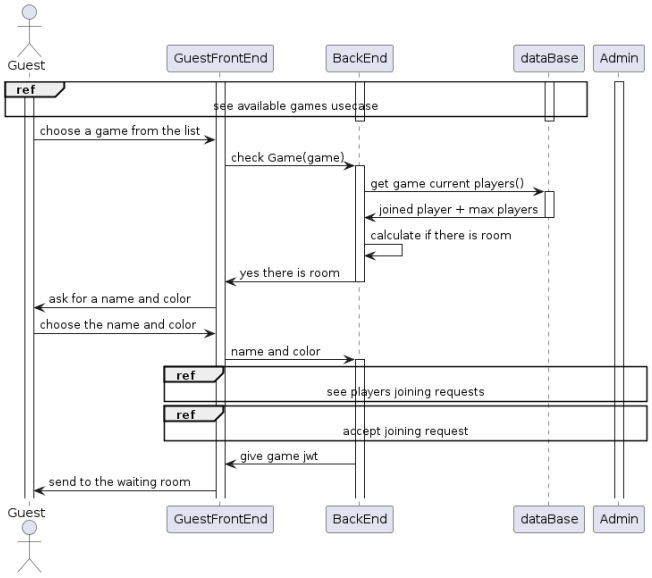

Figure 12: Join game by the guest actor detailed Sequence Diagram

6.3.4 start game by the admin actor usecase

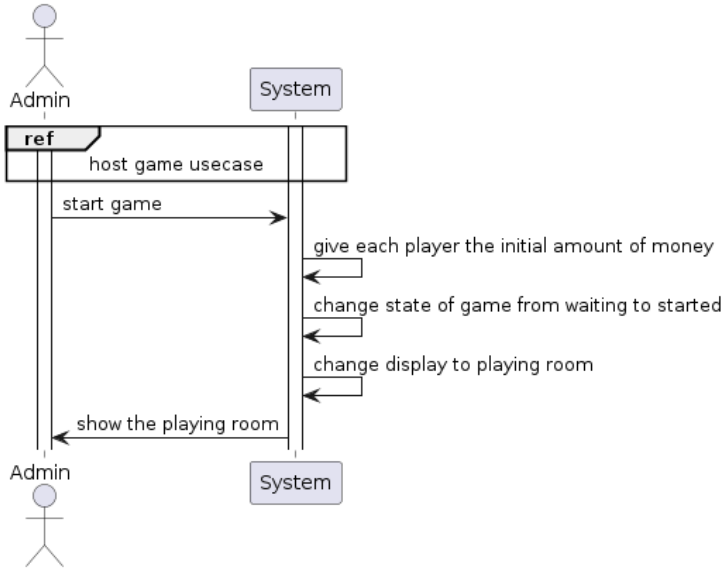

Figure 13: Start game by the admin actor Sequence Diagram

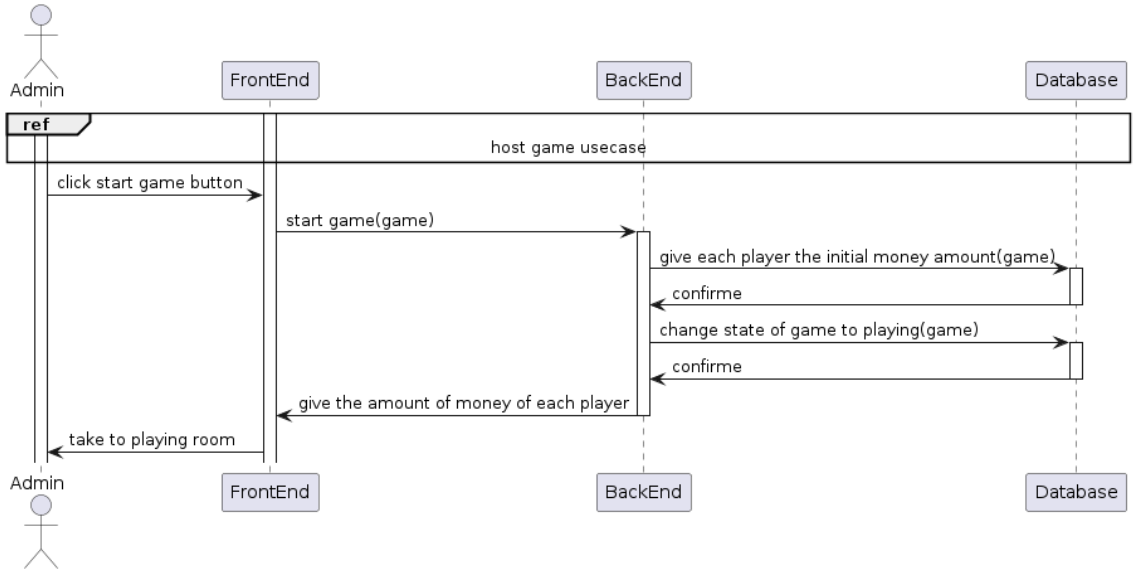

Figure 14: Start game by the admin actor detailed Sequence Diagram

6.3.5 kick player out by the admin actor usecase

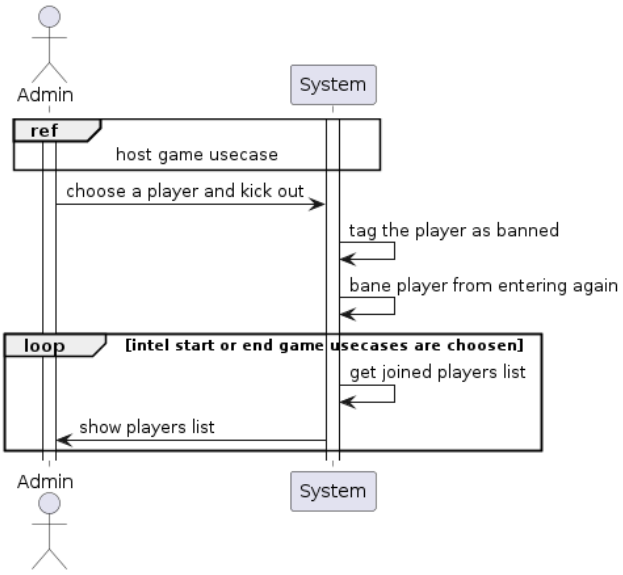

Figure 15: kick player out by the admin actor Sequence Diagram

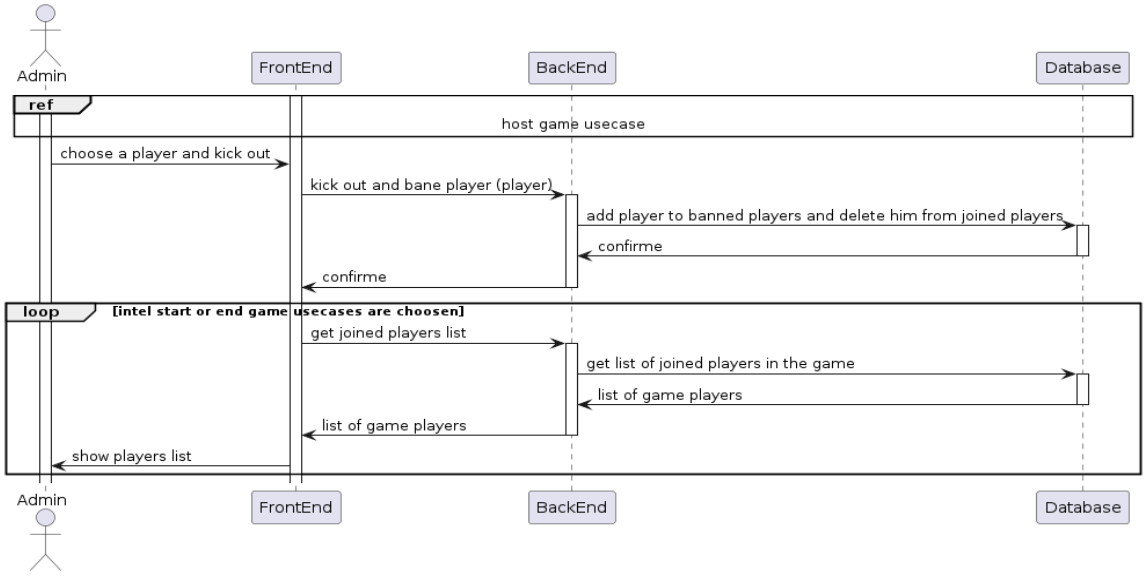

Figure 16: kick player out by the admin actor detailed Sequence Diagram

6.3.6 unbane player by the admin actor usecase

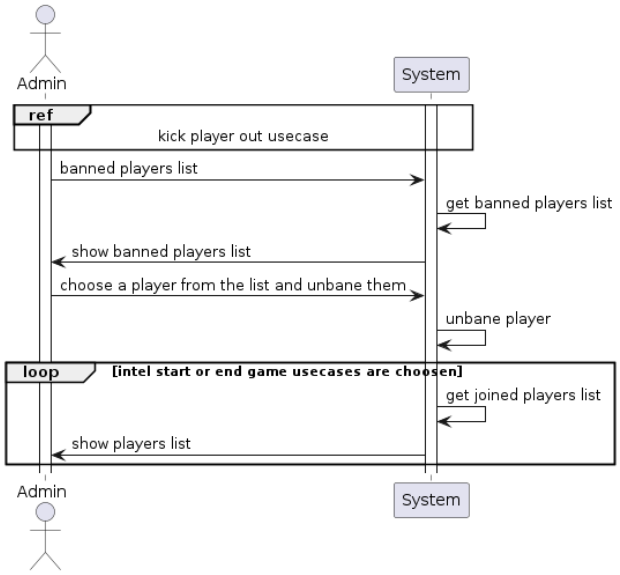

Figure 17: unbane player by the admin actor Sequence Diagram

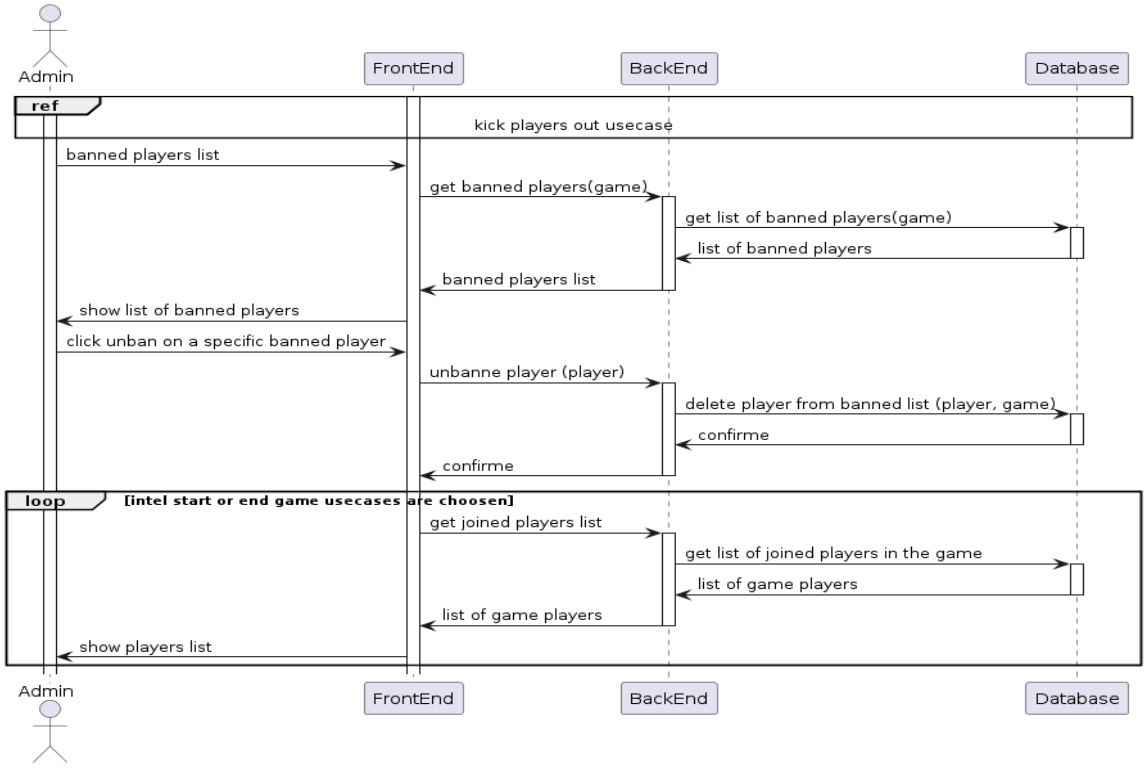

Figure 18: unbane player by the admin actor detailed Sequence Diagram

6.3.7 end game by the admin actor usecase

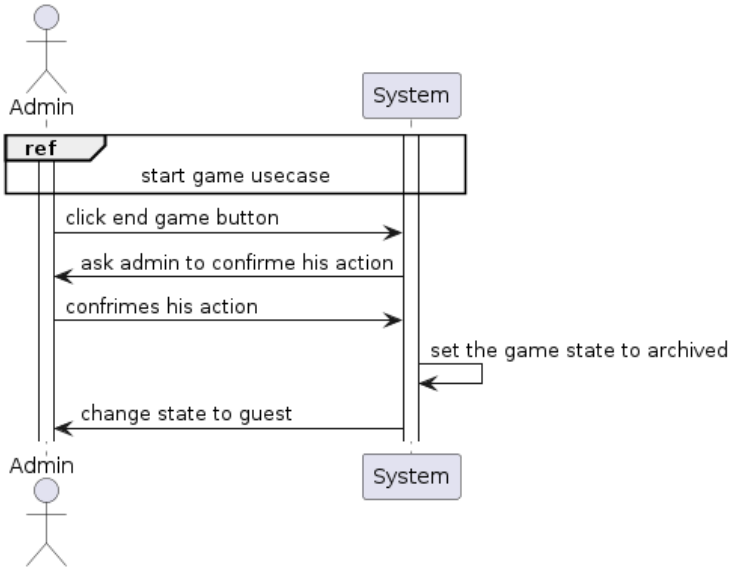

Figure 19: end game by the admin actor Sequence Diagram

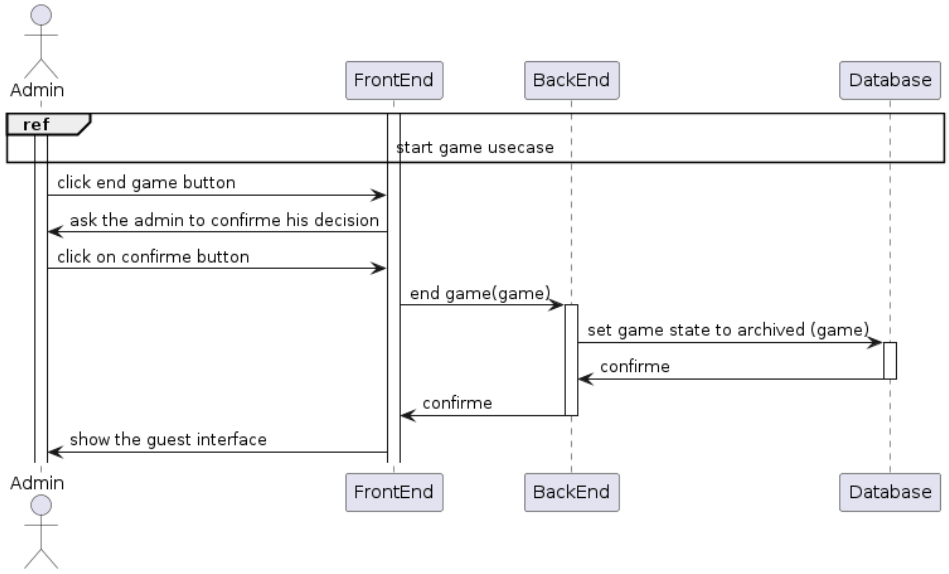

Figure 20: end game by the admin actor detailed Sequence Diagram

6.3.8 accept or refuse transaction by the admin actor usecase

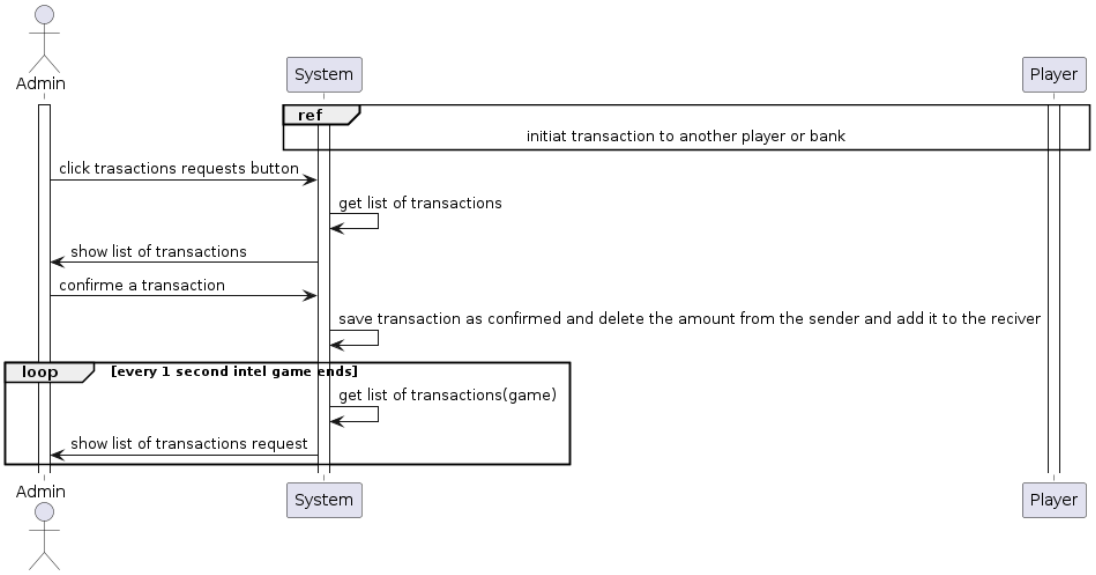

Figure 21: accept or rufuse transaction by the admin actor Sequence Diagram

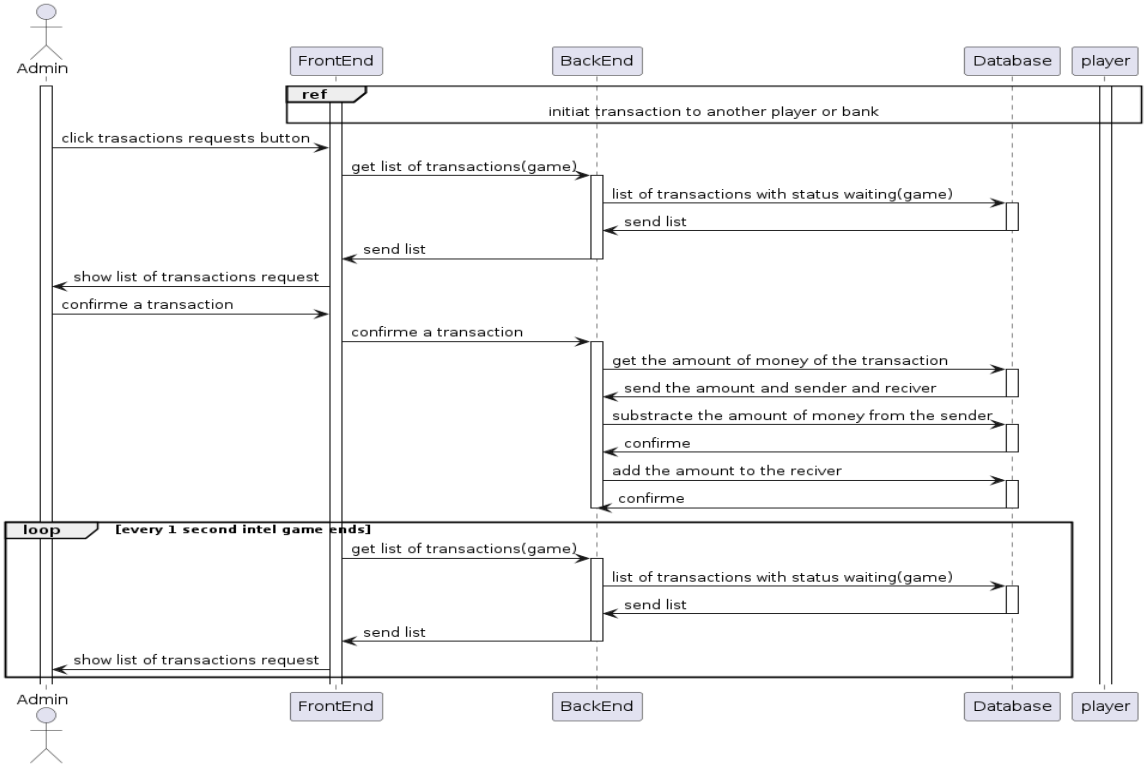

Figure 22: accept or rufuse transaction by the admin actor detailed Sequence Diagram

6.3.9 initiate transaction from the bank to another player by the admin actor usecase

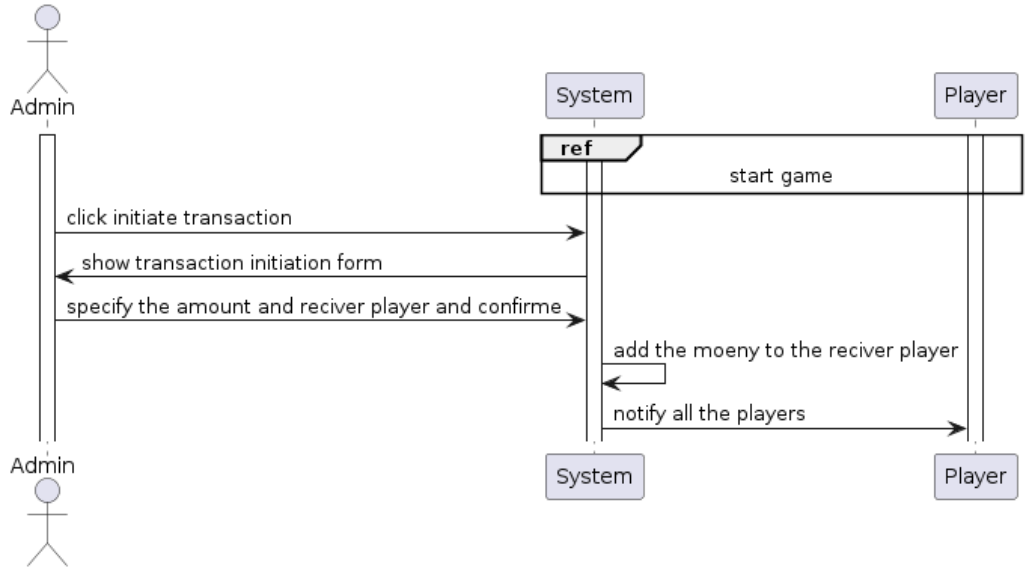

Figure 23: initiate transaction from the bank to another player by the admin actor Sequence Diagram

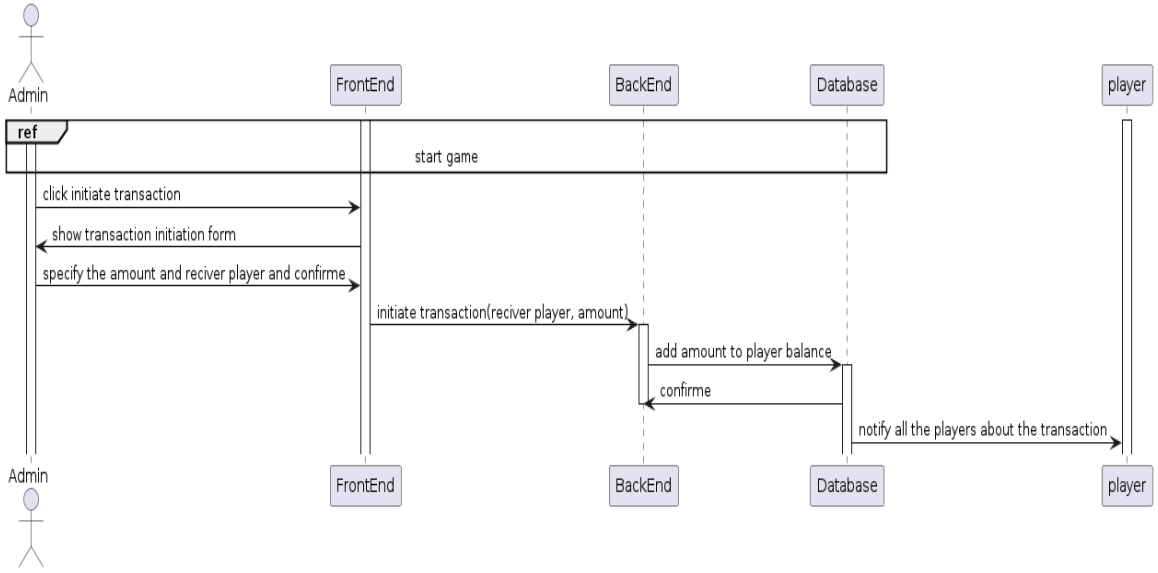

Figure 24: initiate transaction from the bank to another player by the admin actor detailed Sequence Diagram

6.3.10 see transactions full histroy by the admin actor usecase

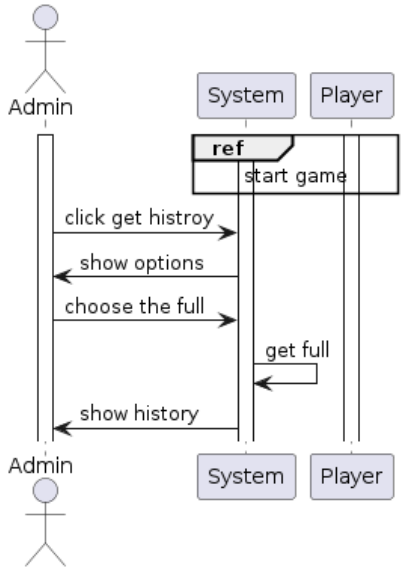

Figure 25: see transactions full histroy by the admin actor Sequence Diagram

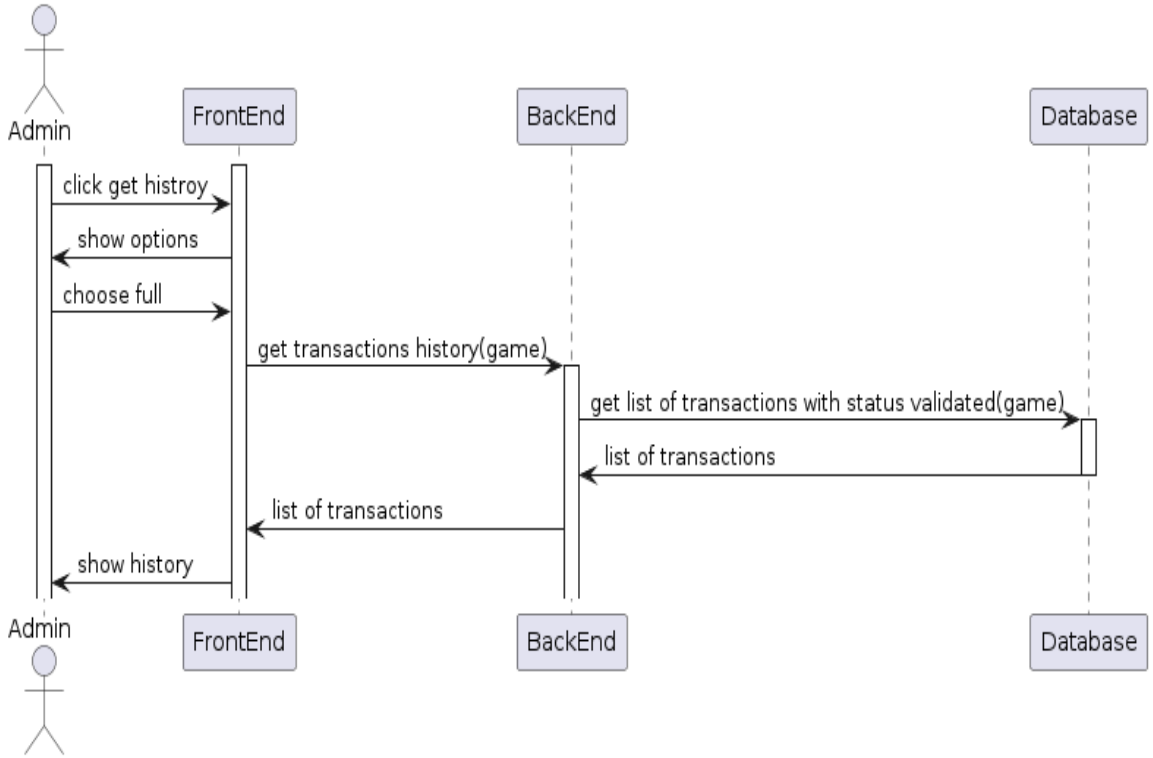

Figure 26: see transactions full histroy by the admin actor detailed Sequence Diagram

6.3.11 see transaction histroy of the bank by the admin actor usecase

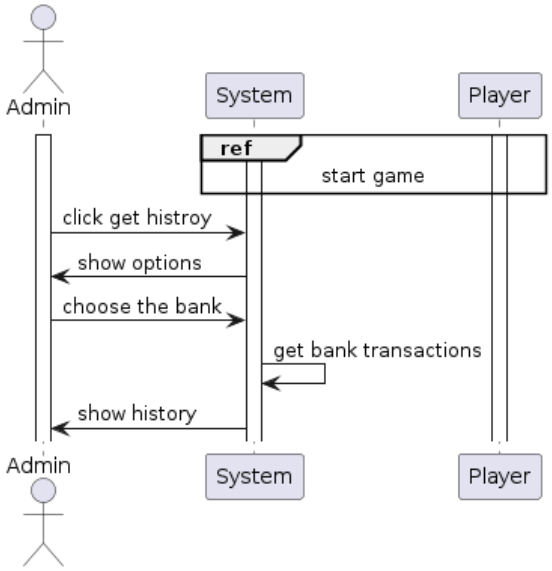

Figure 27: see transaction histroy of the bank by the admin actor Sequence Diagram

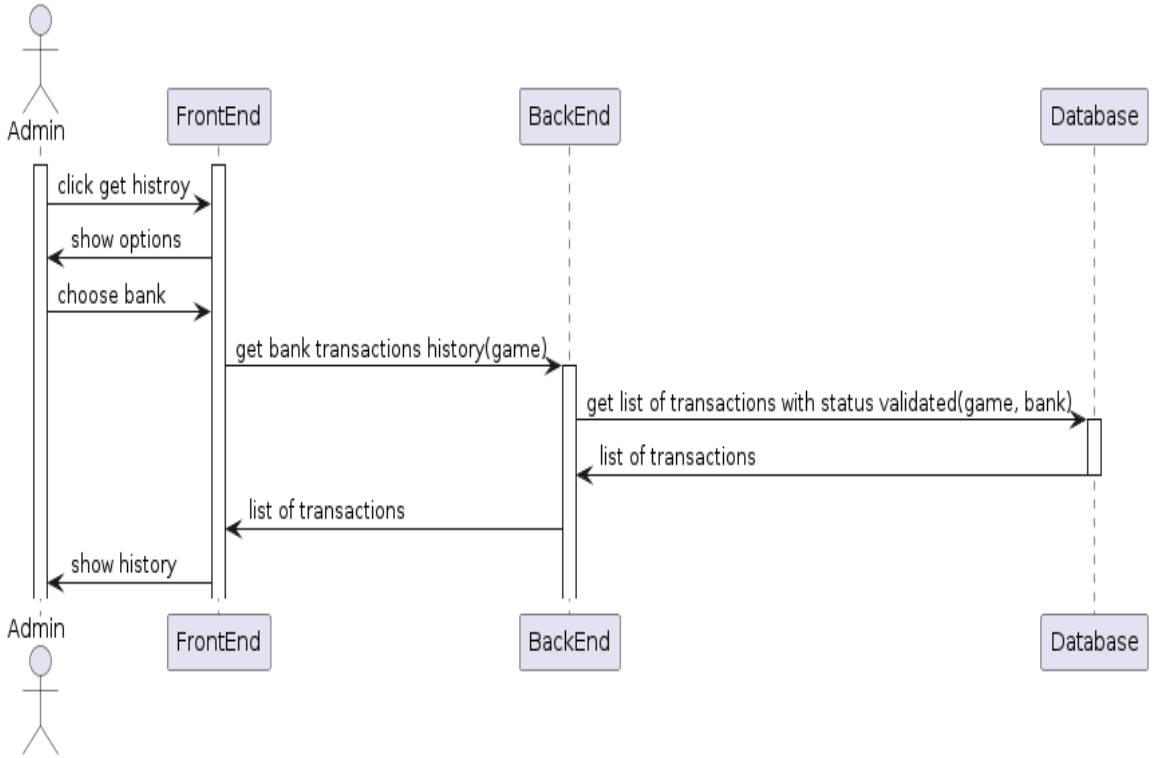

Figure 28: see transaction histroy of the bank by the admin actor detailed Sequence Diagram

6.3.12 see transaction histroy of one player by the admin actor usecase

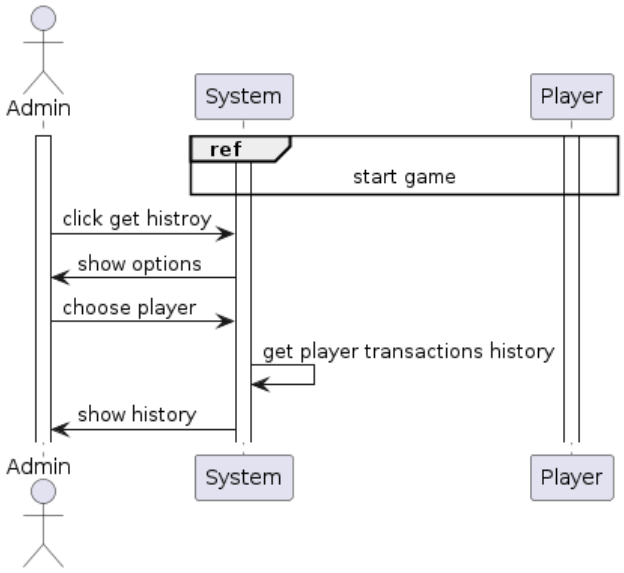

Figure 29: see transaction histroy of one player by the admin actor Sequence Diagram

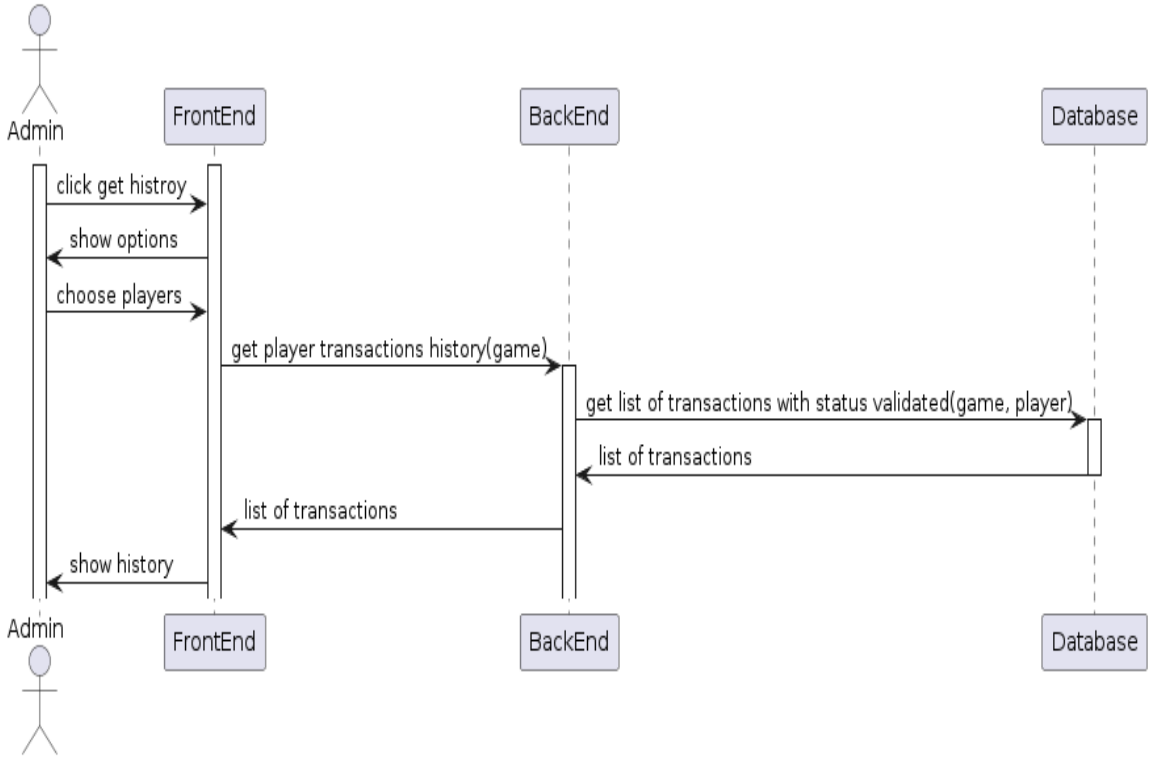

Figure 30: see transaction histroy of one player by the admin actor detailed Sequence Diagram