Software Design Description

for

CHESS

Version 4.0 Approved
March 4, 2010.

Prepared by:

Mayra Aguas, Alfred Blackman, George D'Andrea,

Paul Vu, and Gurvinder Singh.

Drexel University

Revision History

Name	Date	Reason for Changes	Version
Mayra Aguas, Alfred Blackman, George D'Andrea, Paul Vu, and Gurvinder Singh.	15 Feb 2010	Initial Version	1.0
Mayra Aguas, Alfred Blackman, George D'Andrea, Paul Vu, and Gurvinder Singh.	22 Feb 2010	Correct SRS	2.0
Mayra Aguas, Alfred Blackman, George D'Andrea, Paul Vu, and Gurvinder Singh.	1 Mar 2010	Correct Class and Sequence Diagrams	3.0
Mayra Aguas, Alfred Blackman, George D'Andrea, Paul Vu, and Gurvinder Singh.	4 Mar 2010	Correct Class and Sequence Diagrams	4.0

Contents

1. Introduction

1.1. Purpose

This document specifies the software design for CHESS. Software Design Document follows the SRS. The design decisions are based on the functionality, constrains, features and interfaces of the system.

University students need an entertainment tool to enjoy and play with friends over the network. CHESS intend to fill this need by providing a software allows entertainment with friends and over the network.

1.2. Scope

This document specifies the software design for CHESS for the implementation of the system and initial release of CHESS, version 1.0.

Class diagram, Sequential diagrams and Requirements Traceability Table are based on the software requirements specifications provided by the customer.The intended audience of this document exclusively includes the developers, programmers, and testers of the CHESS system.

1.3. Definitions, Acronyms, and Abbreviations

Bishop: one of two pieces of the same color that may be moved any number squares diagonally, as long as no other piece blocks its way. One piece always remains on White squares and the other always on Black.

Castling: to move the king two squares horizontally and bring the appropriate rook to the square the king has passed over.

Check: To make a move that puts the opponents King under direct attack.

Checkmate: a situation in which an opponent's king is in check and it cannot avoid being captured. This then brings the game to a victorious result.

Chess Board: A board you need to play Chess. Have 64 black and white square.

Chess: A game played by 2 people on a chessboard with 16 pieces each.

Client system interface: Interface to the user's client to receiver user input and moves selections for the game.

Connect interface: Interface used by players and display player information .As players make moves, the screen is updated to reflect the moves made in the game.

Communication protocol: Rules determining the format and transmission of data

Internet: A system connecting computers around the world using **TCP/IP,** which stands for Transmission Control Protocol/Internet Protocol, a set of standards for transmitting and receiving digital data. The Internet consists primarily of the collection of billions of interconnected webpages that are transferred using HTTP (Hypertext Transfer Protocol), and are collectively known as the World Wide Web. The Internet also uses FTP (File Transfer Protocol) to transfer files, and SMTP (Simple Mail Transfer Protocol) to transfer e-mail.

Internet IP: Internet protocol: a code used to label packets of data sent across the Internet, identifying both the sending and the receiving computers

En Passant: a method by which a pawn that is moved two squares can be captured by an opponent's pawn commanding the square that was passed

King: The main piece of the game, checkmating this piece is the object of the game. It can move 1 space in any direction.

Knight: This piece can move 1 space vertically and 2 spaces horizontally or 2 spaces vertically and 1 space horizontally. This piece looks like a horse. This piece can also jump over other pieces.

Network: A system of computers and peripherals, such as printers, that are linked together. A network can consist of as few as two computers connected with cables or millions of computers that are spread over a large geographical area and are connected by telephone lines, fiberoptic cables, or radio waves. The Internet is an example of very large network

Network system interface: Interface to the network in order to transmit information and connect players.

Network software interface: Interfaces with the user computer and expect that it is capable of use TCP connections

Pawn: One of eight men of one color and of the lowest value usually moved one square at a time vertically and capturing diagonally.

Player or user: A user will be the person that is playing the chess game.

Queen: This piece can move in any number of spaces in any direction as long as no other piece is in its way.

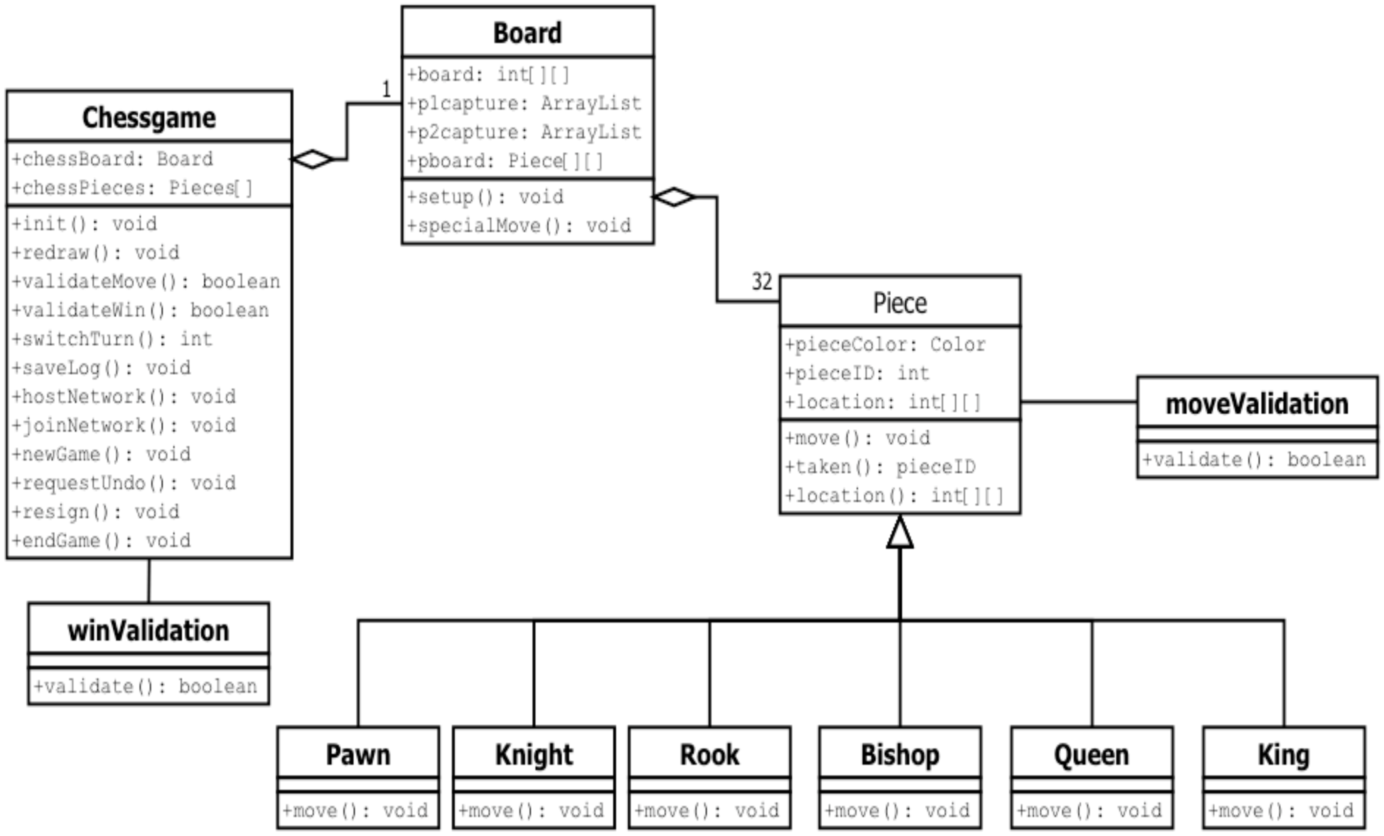
Rook: one of two pieces of the same color that may be moved any number squares horizontally or vertically, as long as no other piece blocks its way.

Stalemate: A situation in which a player’s king is not in check, but that player can make no move. This then result is a stalemate, which is a draw.

TCP/IP: Transmission Control Protocol/Internet Protocol. Standard Internet communications protocols that allow digital computers to communicate over long distances.

2. Static Design Structure Modeling

2.1. Class Diagram



2.2. Class Description

2.2.1. Chessgame

The main class of the chess program – here the board will be set up, the players will connect and it is how the players will interact with the system.

2.2.2. Board

The playing area – The board will be an 8x8 playing area upon which the chess pieces are moved by the players.

2.2.3. winValidation

The class that checks for a winning move – when a piece is moved, winValidation will check the board to see if the game is over or not according to the rules of chess.

2.2.4. Piece

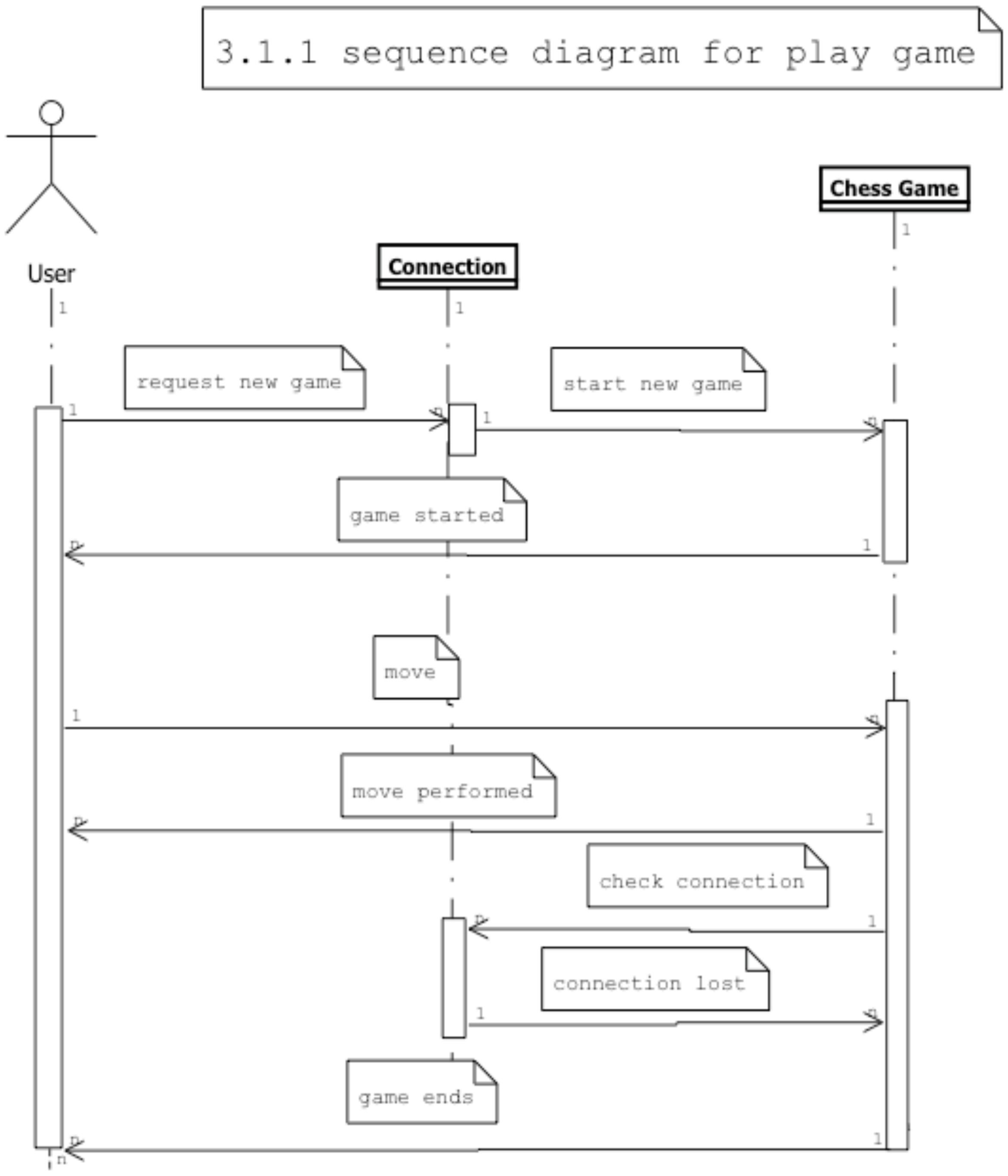
The class that encompasses the piece types – Pieces can be pawns, knights, rooks, bishops, queens, and king. The difference between them is appearance and available moves.

2.2.5. moveValidation

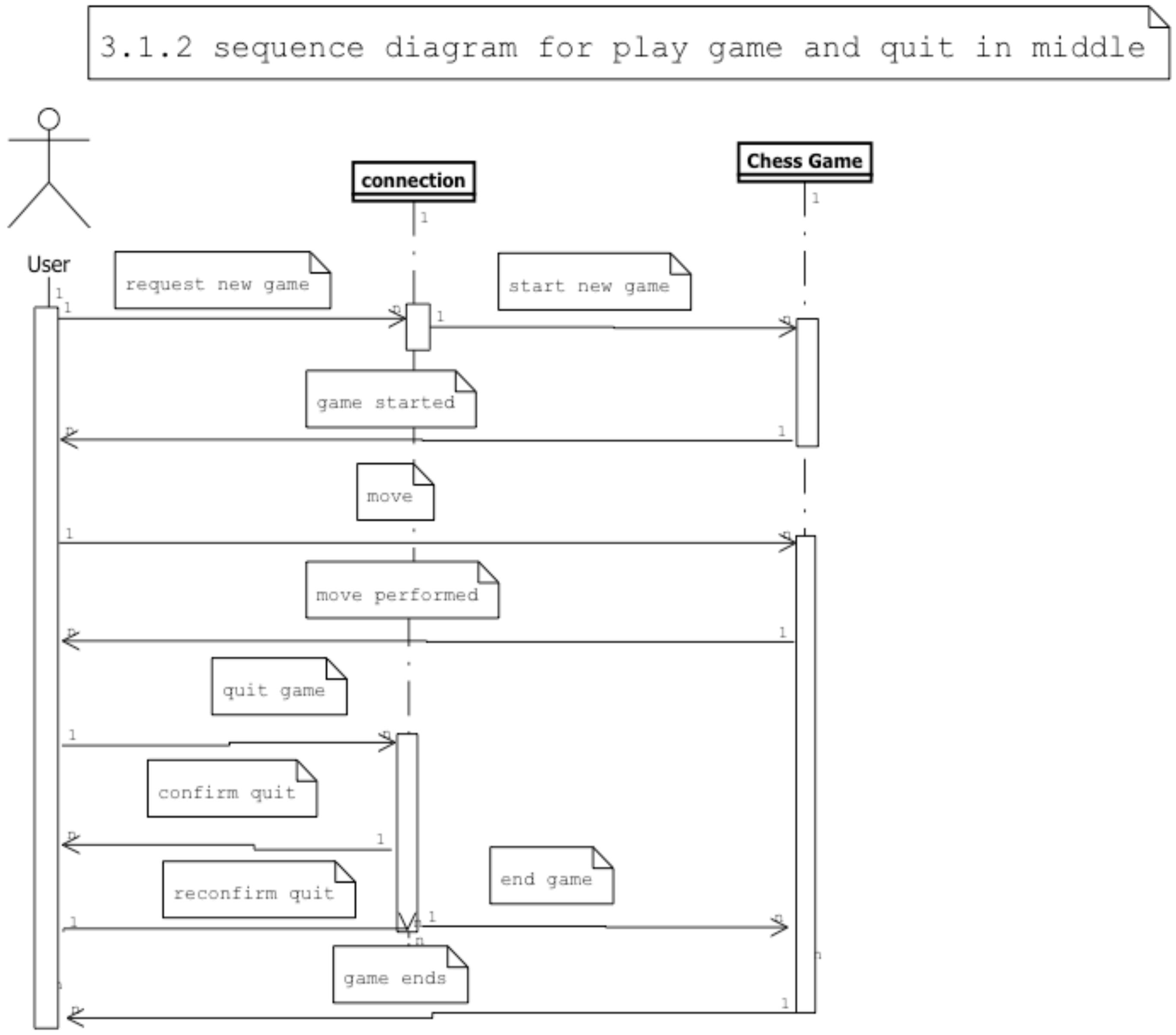
The class that checks pieces for valid moves – When a piece is selected, moveValidation will assure the only place it can be placed is a valid one as per the piece's class.

3. Dynamic Behavior Modeling

3.1.1. Play Game Sequence Diagram

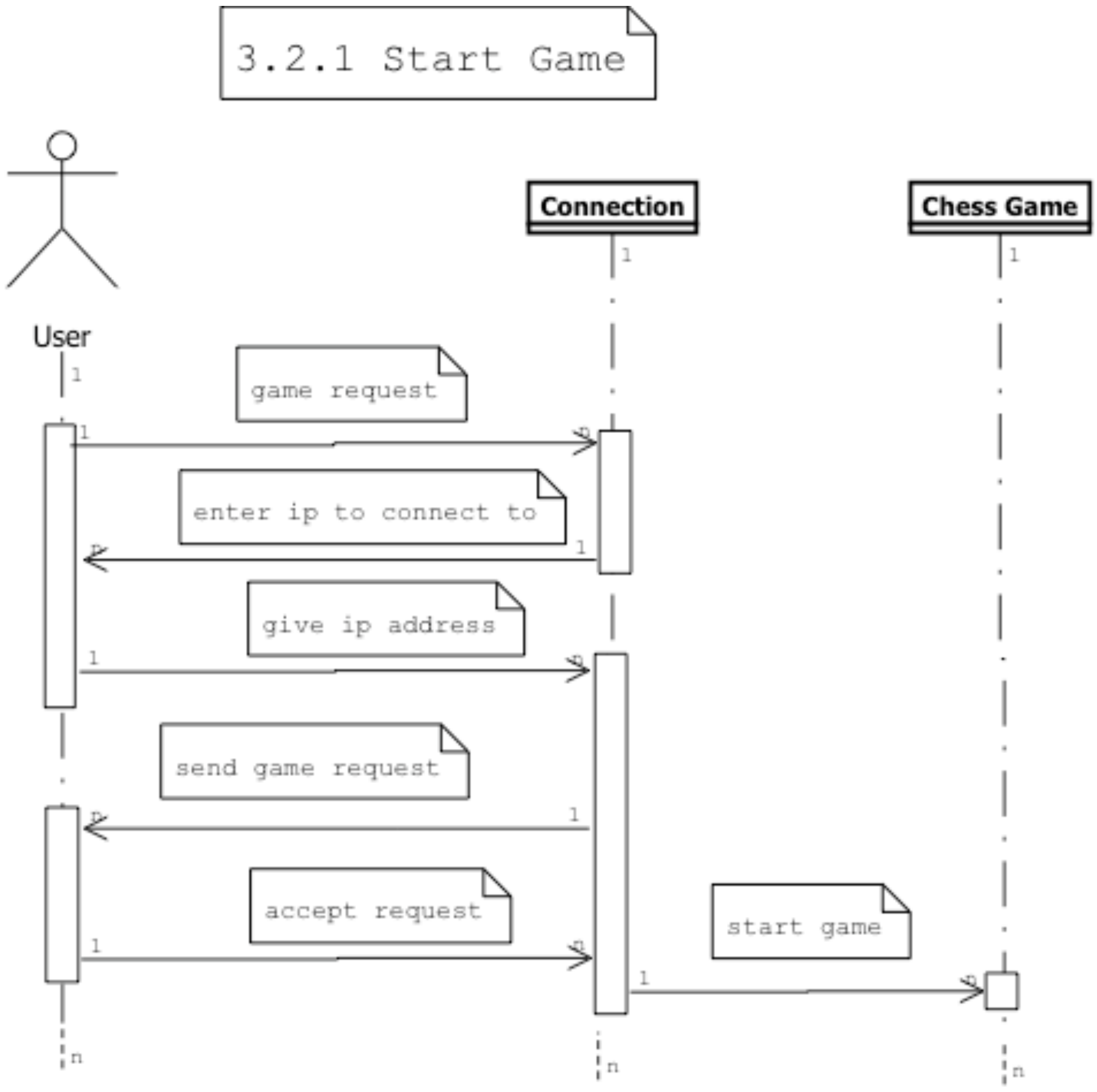


3.1.2. Sequence diagram for play game and quit in the middle

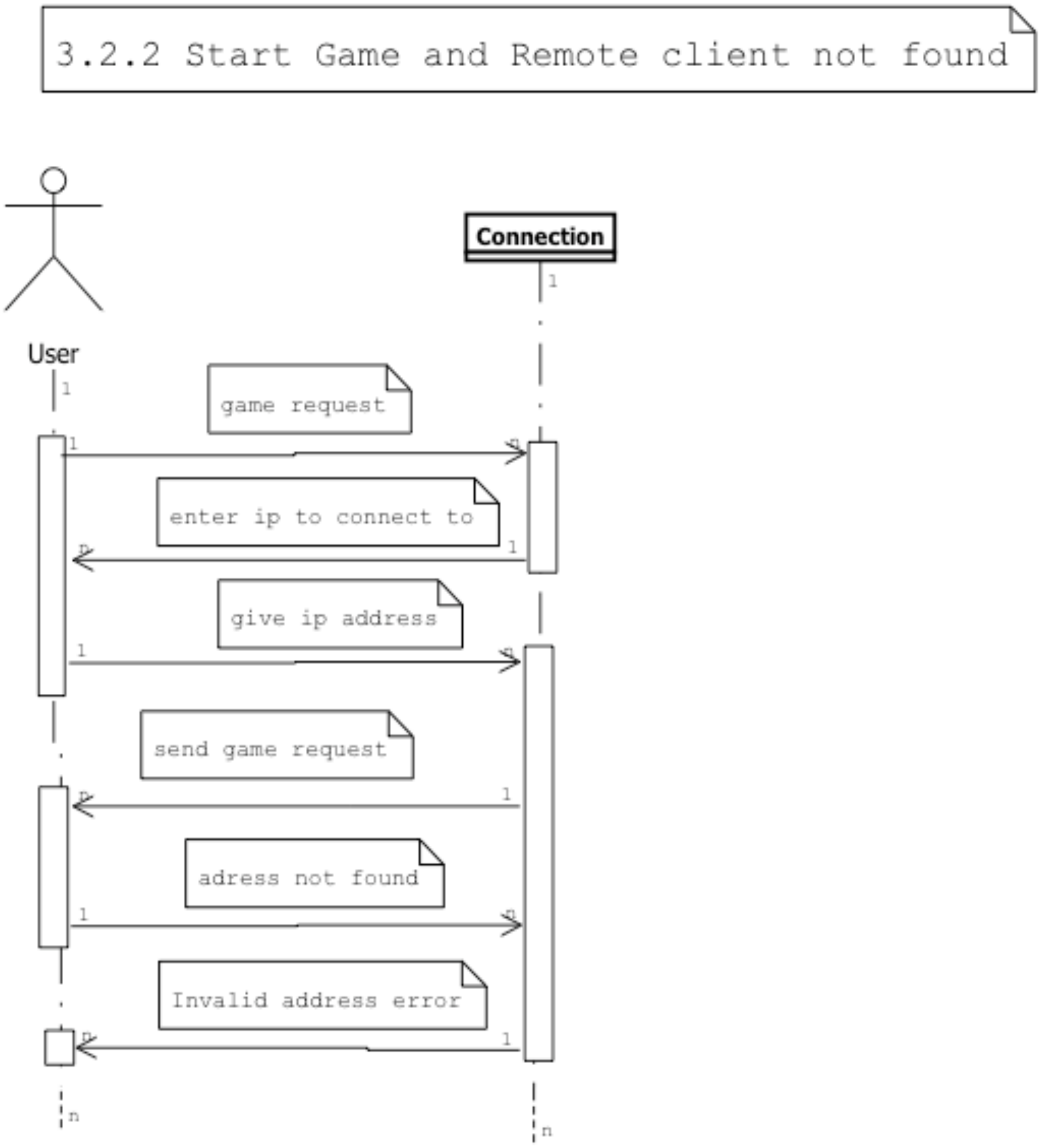


3.2. Start Game

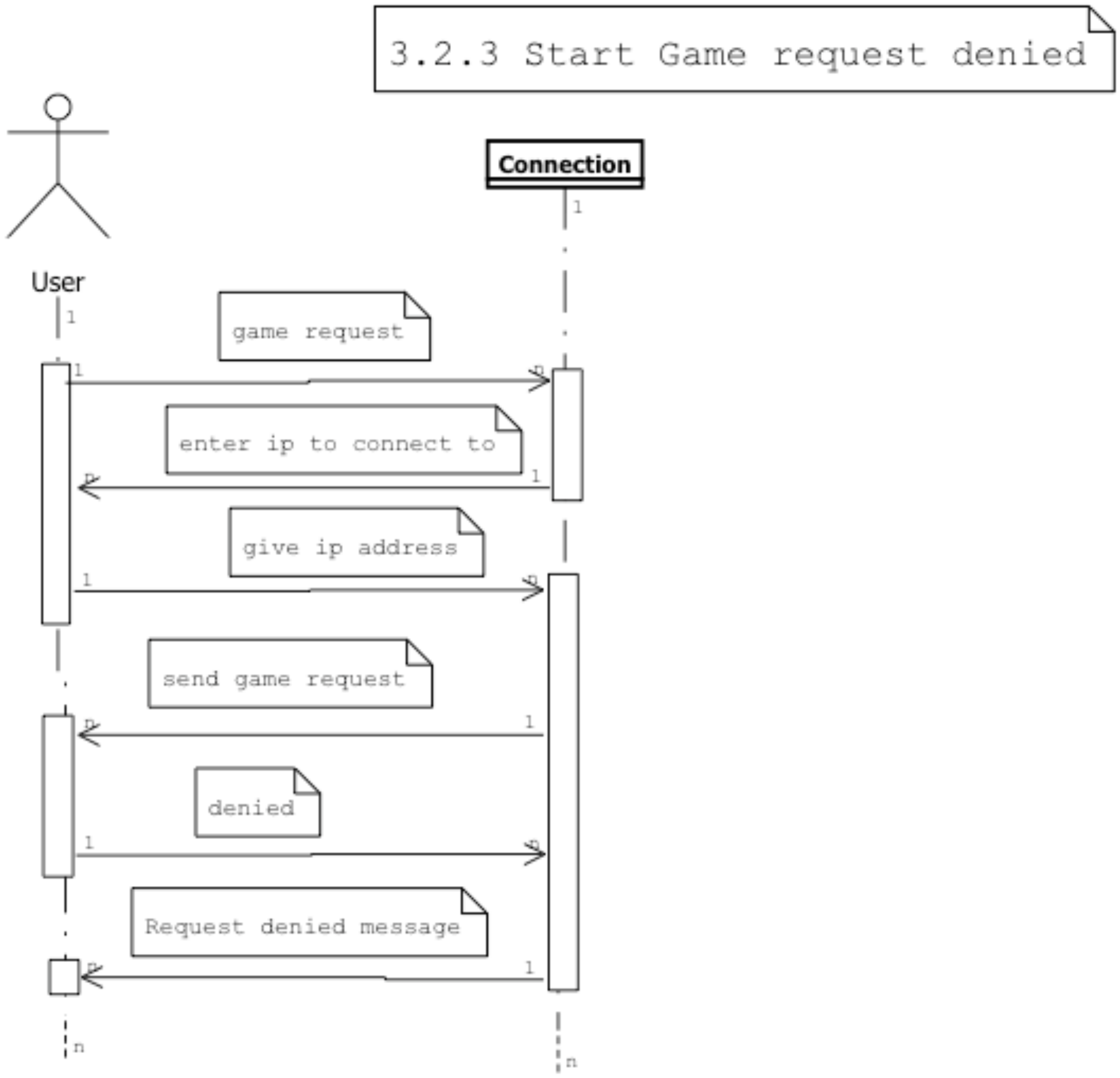
3.2.1. Sequence diagram for Start Game



3.2.2. Sequence diagram for Start Game and Remote client not found

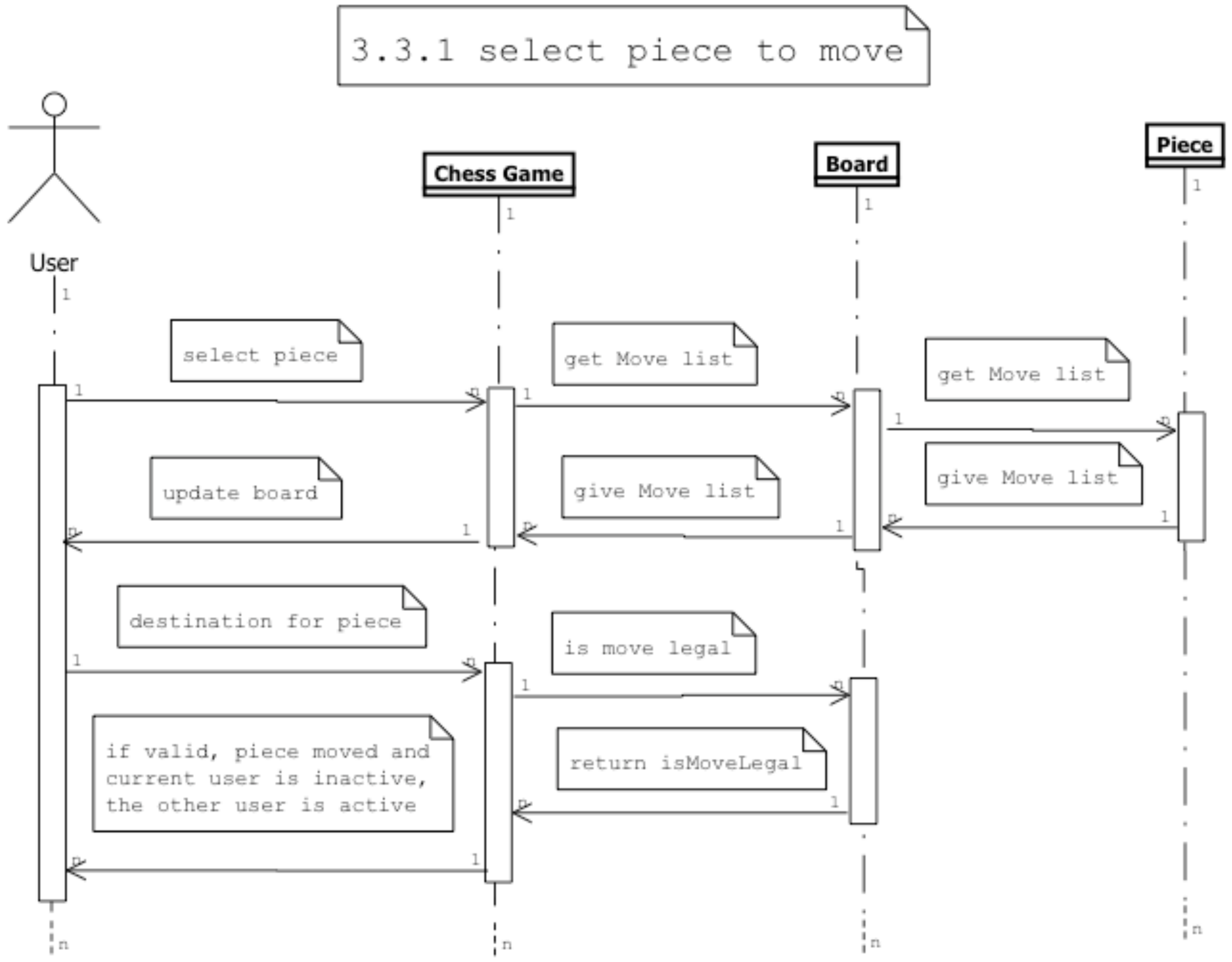


3.2.3. Sequence diagram for Start Game request denied

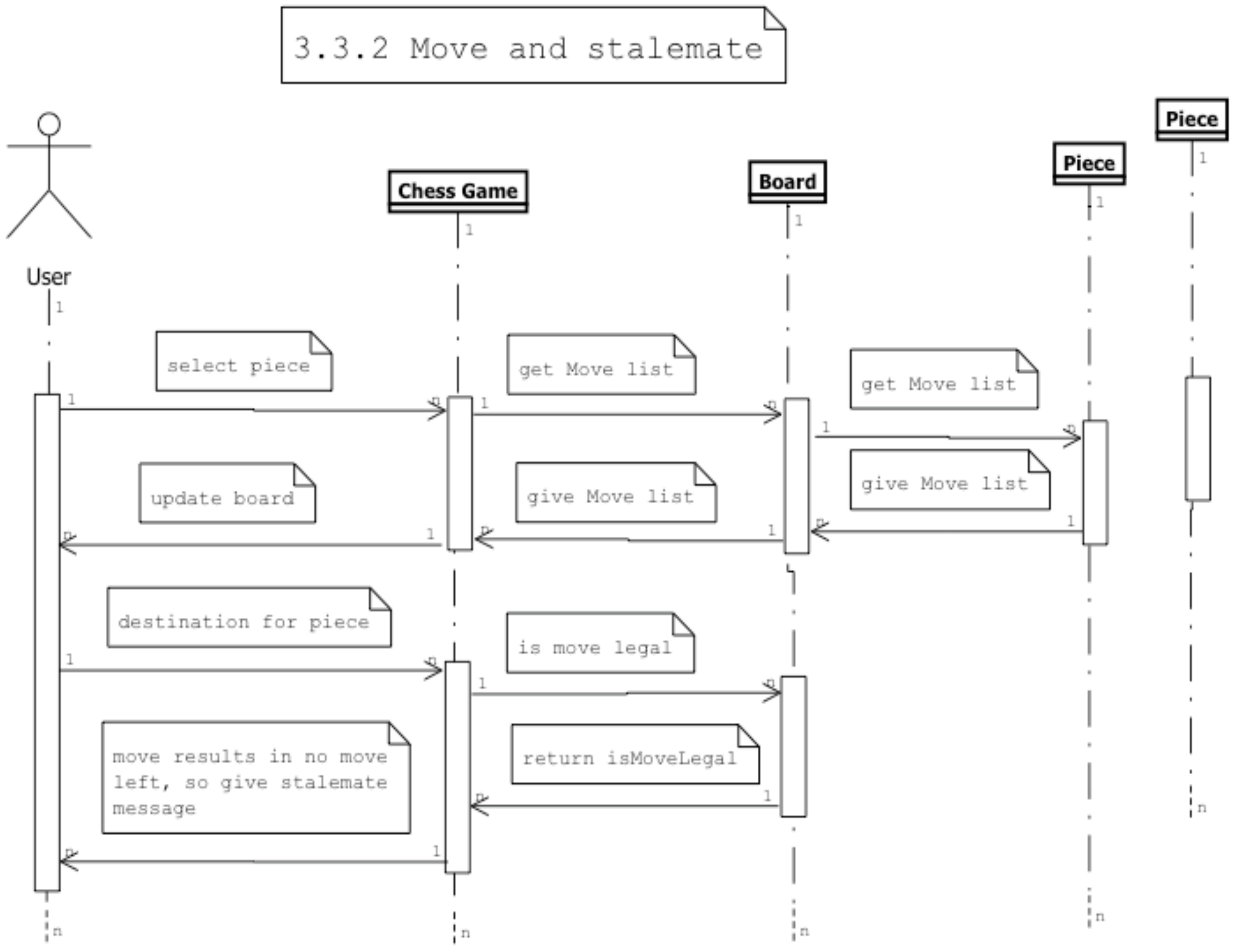


3.3. Sequence diagram for Move

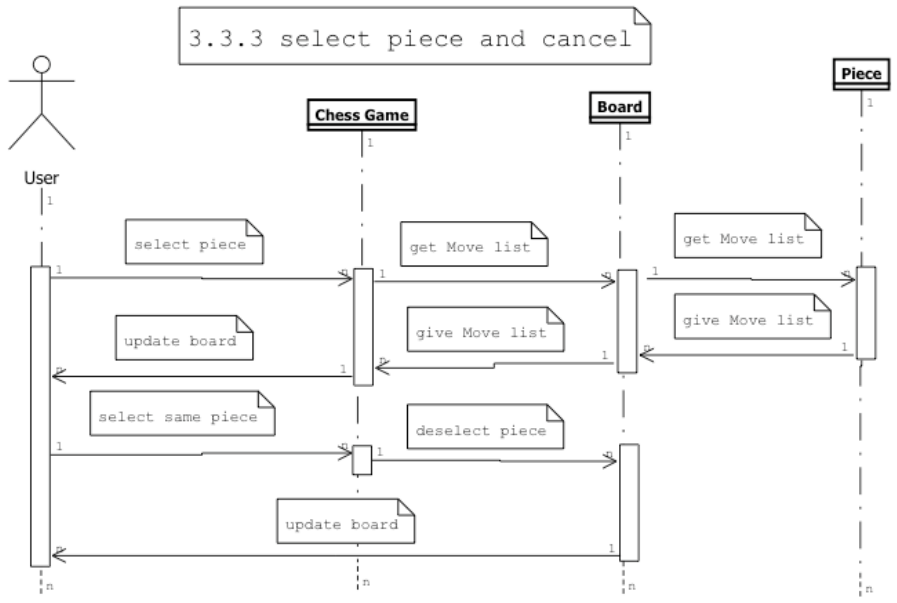
3.3.1. Sequence diagram to Move



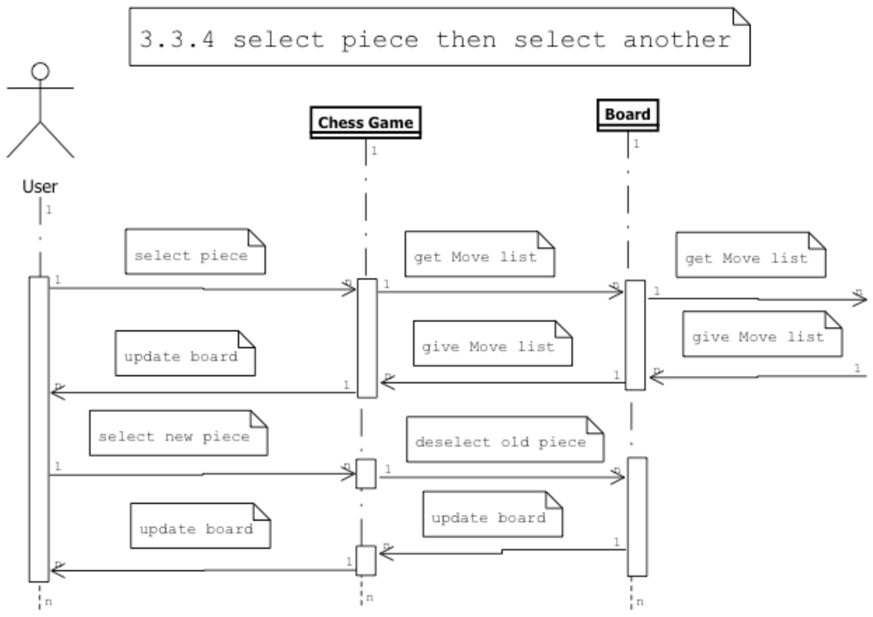
3.3.2. Sequence diagram for Move and Stalemate



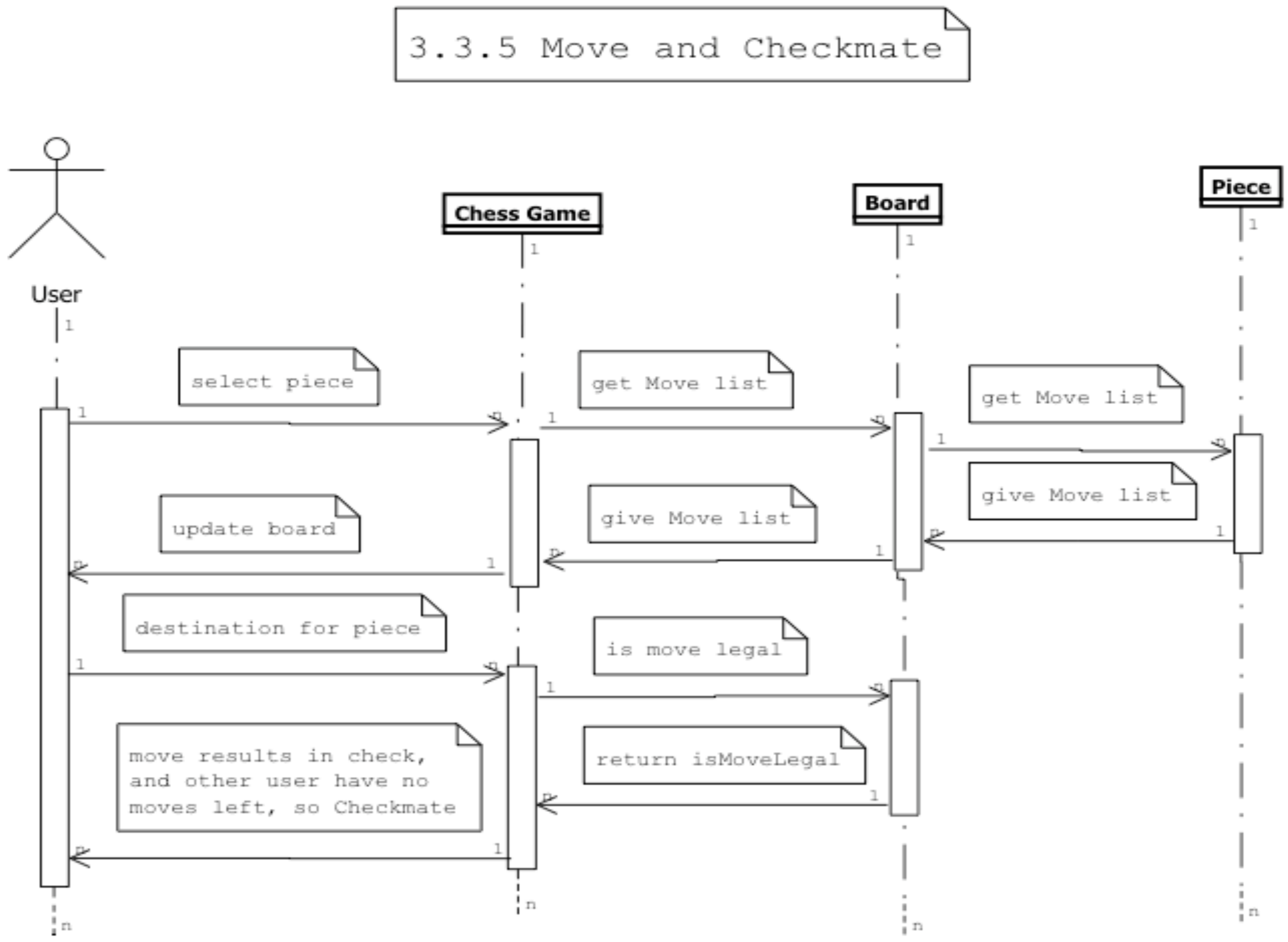
3.3.3. Sequence diagram for Select a piece and cancel



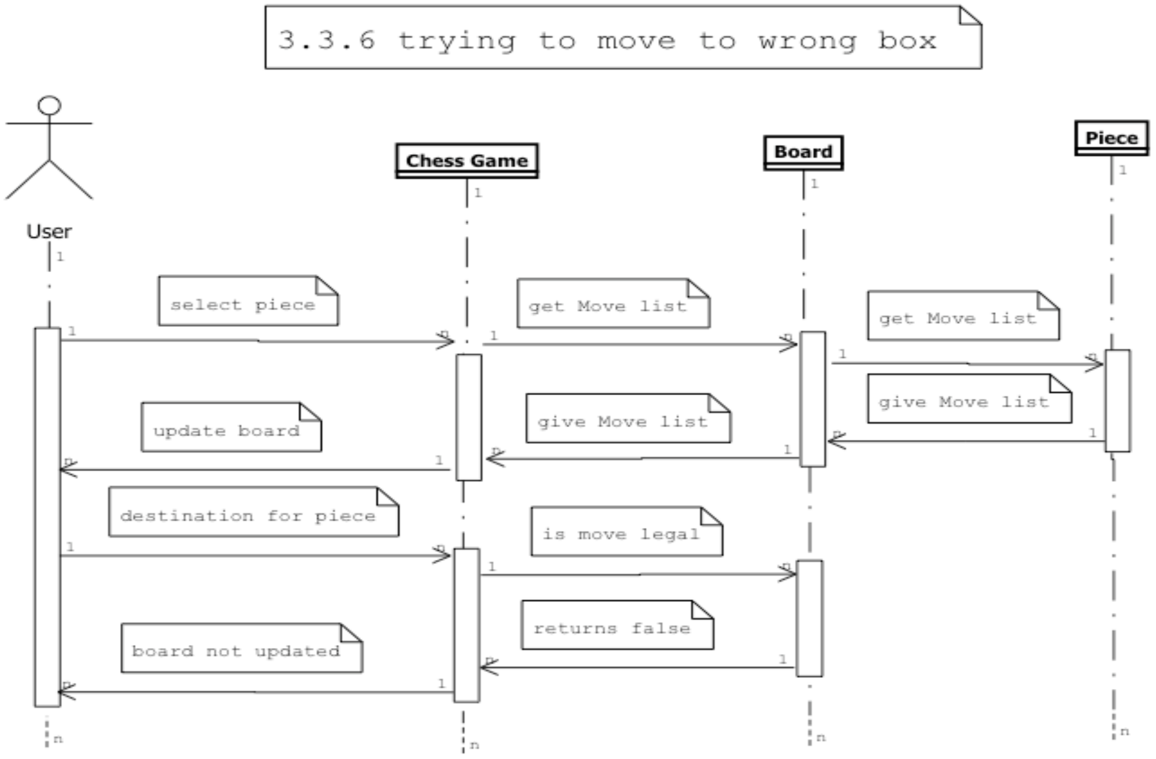
3.3.4. Sequence diagram for Select a piece then select another

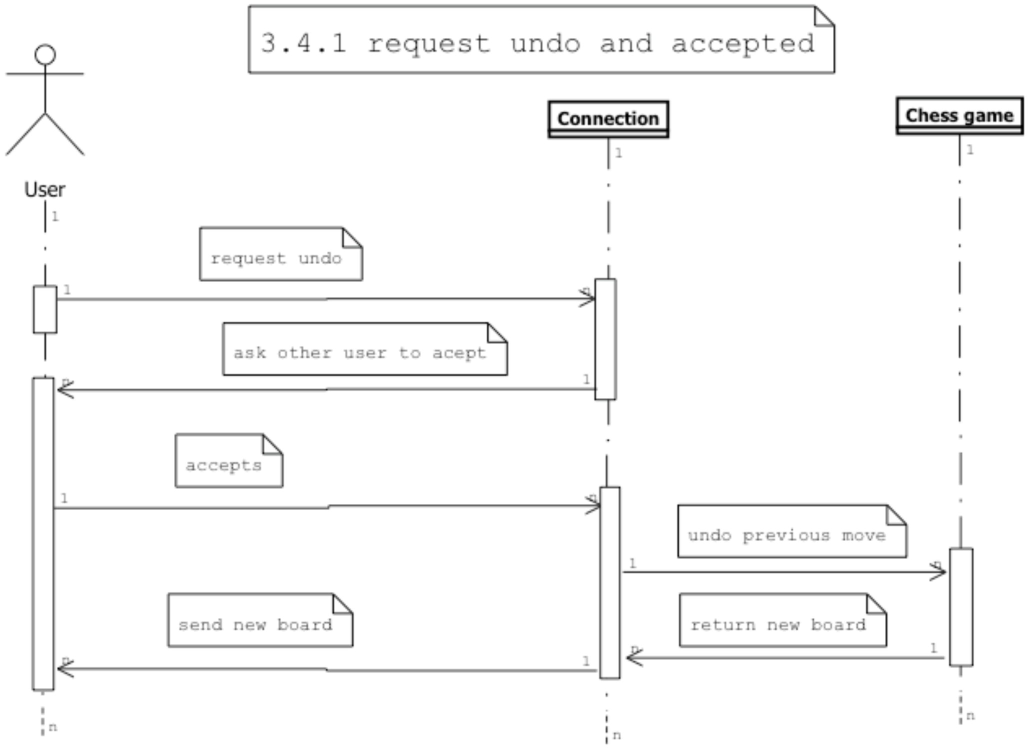


3.3.5. Sequence diagram for Select a piece then select another



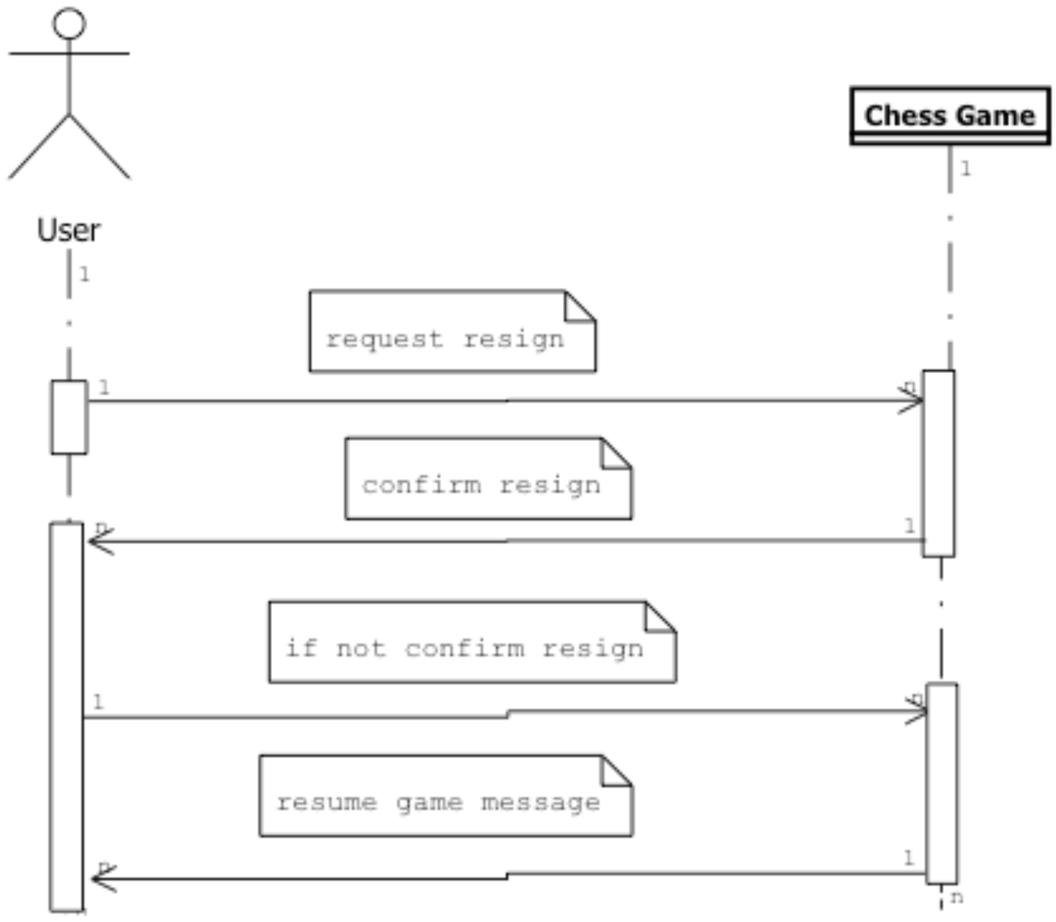
3.3.6. Sequence diagram for Select a piece to move wrong box



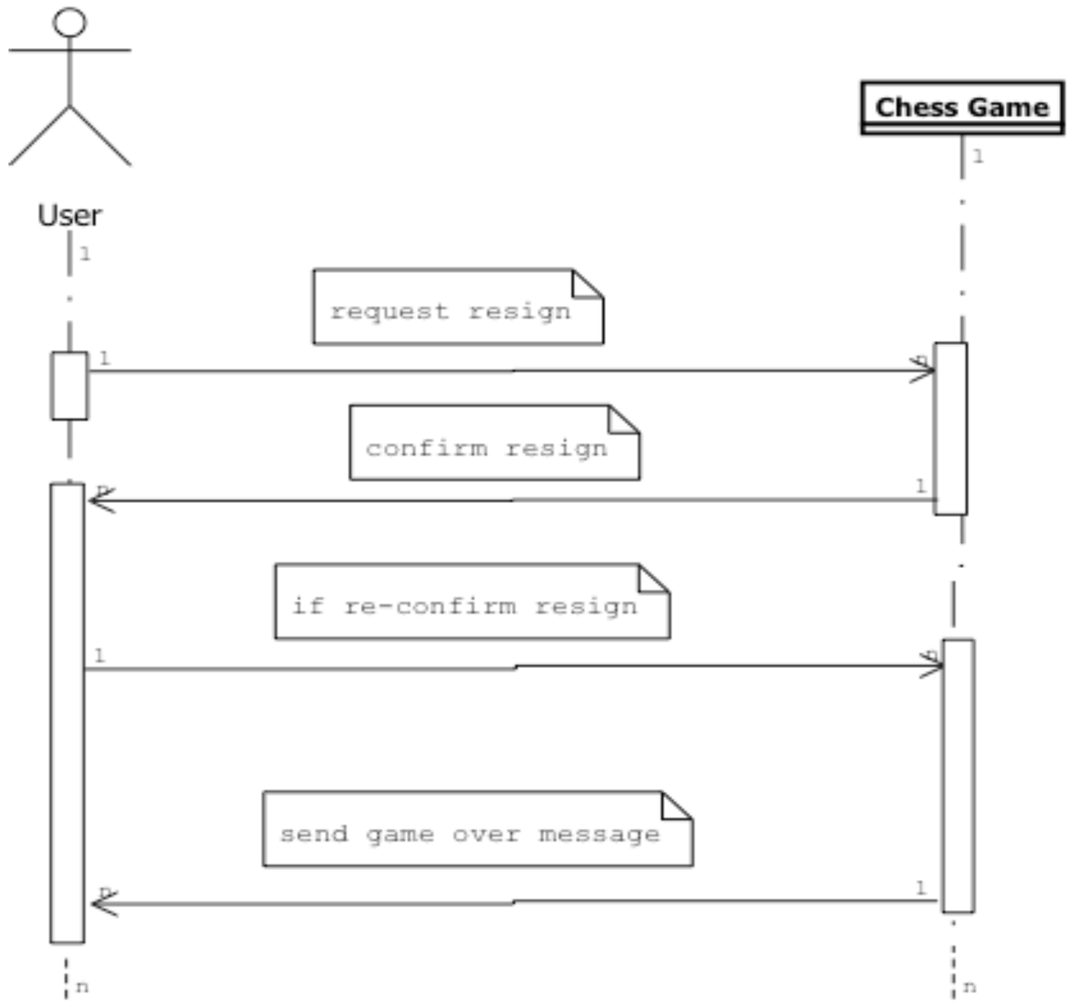


3.4. Sequence diagram for Resign

3.4.1. Sequence diagram for Resign and not confirm



3.4.2. Sequence diagram for Resign and confirm



4. Requirement Traceability Table

4.1. Traceability by Requirement Numbers

Req #	Design Component
0010	3.1.1
0020	3.1.1
0110	3.1.1,3.2.1
0120	3.2.1
0130	3.2.1
0140	3.2.1
0150	3.2.1
0210	3.1.1,3.3.1
0220	3.3.1
0230	3.3.1
0240	3.3.1
0310	3.3.1,3.4.1
0320	3.4.1,3.4.2
0330	3.4.1,3.4.2
0340	3.4.1,3.4.2
1001	3.1.1
1002	3.2.1
1003	3.2.1
1004	3.3.6
1005	3.3.3
1006	3.3.6
1007	3.3.3,3.3.4
1008	3.3.2
1009	3.3.2
1010	3.3.5
1011	3.4.1,3.4.2
1012	3.4.2

5. References

[1] http://www.cs.drexel.edu/~yfcai/CS451/artifacts/sdd.pdf
[2] http://www.thefreedictionary.com/

6. Use Cases Specifications (Optional)

6.1. Flow of Events for the Play Game Use Case

6.1.1. Preconditions

Remote player and local player shall have started the programs and local player knows the IP address of the other.

6.1.2. Main Flow

The case beings when the local player requests a play game (0010, 0110). Player shall play game with remote player (0210). The case finishes when remote or local player quit game (0020, 1001).

6.1.3. Subflows

0110 Start Game
The player starts a new game using the Start Game Case.
0210 Play Game
Player shall alternate moves using the Move Game Case (0310).
0020 Quit Game
Player shall terminate game. If there is no connection active the program will terminate.

6.1.4. Alternative Flows

1001 Quit While Connection is Active. The system shall prompt the player's confirmation.

6.2. Flow of Events for the Start New Game Use Case

6.2.1. Preconditions

Play Game was initialized.

6.2.2. Main Flow

The case beings when the user requests a new game (0110). The user enters the IP address of remote player (0120, 0130, 0140, 0150, 1002, 1003). The case finishes when system created a new game (0150).

6.2.3. Subflows

0110 Game Request

The users shall request a new game.

0120 Access

The system prompts the user for the IP address to connect. The user shall enter the IP address. The system shall send a game request to the IP address (0130).

0130 Send game request

The system shall send a game request to the IP address.

0140 Accept game request

The player shall accept the game request. The system shall begin a new game (0150).

0150 Begin new game

The system should initialize and begin the game.

6.2.4. Alternative Flows

1002 Remote Client Not Found. Invalid Address or Connection. System shall notify error.

1003 Decline Game Request. Remote Player declines game request. System shall notify error.

6.3. Flow of Events for the Move Use Case

6.3.1. Preconditions

Play Game was initialized and Start New Game was created.

6.3.2. Main Flow

The case beings when is the local player's turn to move (0210). The player shall select a piece from the board (0220, 1004, 1005). The player shall select the destination (0230, 1006, 1007). The case finishes when the system shall moves the piece (0240, 1008, 1009, 1010, 1011).

6.3.3. Subflows

0220 Select Piece

The player shall click a piece to select the piece from the board (1004). The system shall mark the piece and validate possible destinations for the piece (1005).

0230 Select Destination

The player shall click a destination (1006). The system shall move the piece (0240).

0240 Move Piece

The system shall move the piece to the destination and alternate the turn to the remote player. The remote player shall move piece using Move Use Case (0210).

6.3.4. Alternative Flows

1004 Invalid Piece Selection. The player selected an invalid piece. The player shall select a valid piece (0240).

1005 No Destination Available. The piece not have valid possible destination. The system shall unmark the piece. The player shall select a valid piece (0240).

1006 Invalid destination. The player selected an invalid destination. The system shall select a valid piece (0240).

1007 Different Piece Selection. The piece select as the destination another of his/hers own pieces. The system shall mark the piece and validate possible destinations for the piece (1005).

1008 Capture

The opponent player captures the piece. The system shall move the captured piece to "captures section". The player shall select a valid piece (0240).

1009 Check

remote player's piece in check. The system shall notify check status to the players.

1010 Checkmate

The move results in checkmate. The system shall notify checkmate status to the players. The system shall close the connection and initial a new game. The system shall allow the player to create a new game using Play Game Use case (0010).

1011 Resign

The player shall select resign using the Resign Use Case (0310). The system shall terminate the game using Terminate sub flow of Resign Use Case (0310).

6.4. Flow of Events for the Resign Use Case

6.4.1. Preconditions

Play Game was initialized and Start New Game was created.

6.4.2. Main Flow

The case start when the player accepts defeat (0310). The player shall select the "Resign Game"(0320). The program shall verify resign with the player (0330). The use case ends when the game is terminated (0340, 1012).

6.4.3. Subflows

0320 Select Resign
The local player shall select "Resign Game." The program shall confirm resign with the player (0330).
0330 Verify Resign
The program shall confirm resign to the local player. If the local player confirm, the program shall Terminate Game (0340)
0340 Terminate Game
The program shall notify the player, who won the game. The program shall close the connection, and restart game. The program shall allow the player to create a new game per the Start Game use case (0110).

6.4.4. Alternative Flows

1012 Resignation Canceled
The player selects to cancel resignation. The system shall continue using Move Piece use case (0210).

6.5. Non-Functional Requirements

6.5.1. Performance Requirements

6.5.1.1. Computer Requirements

2001CHESS can be installed in any operating system

6.5.1.2. Program Stability

2002 CHESS will not crash when it is being used.

6.5.2. Security Requirements

6.5.2.1. Computer Security

2003 CHESS will not open any vulnerability to user computers

6.5.3. Software Quality Attributes

6.5.3.1. Program Platform

2004 CHESS will be platform independent.

6.5.3.2. Program Modules

2005 CHESS components should be maintainable

6.5.3.3. Program Usability

2006 CHESS shall be easy to use with a simple interface

2007 CHESS shall be easy to understand a simple interface