

SOFTWARE REQUIREMENT SPECIFICATION (JATIN JAIRATH PROJECT)



GAMING (LUDO USING C LANGUAGE):-

TABLE OF CONTENT:-

1. INTRODUCTION :-
 - 1.1 PURPOSE
 - 1.2 SCOPE
 - 1.3 DEFINITION , ACRONYMS AND ABBREVIATIONS
 - 1.4 REFERENCE
 - 1.5 OVERVIEW
2. OVERALL DESCRIPTION:-
 - 2.1 PRODUCT PERSPECTIVE
 - 2.2 MAIN FEATURES
 - 2.3 USER CHARACTERISTICS
3. SPECIFIC REQUIREMENTS:-
 - 3.1 EXTERNAL INTERFACE SPECIFICATION
 - USER'S INTERFACE
 - SOFTWARE INTERFACE
 - COMMUNICATION INTERFACE
 - 3.2 FUNCTIONAL REQUIREMENT
 - 3.3 DESIGN CONSTRAINTS

Introduction:-

Ludo (from Latin *ludo* , "I play") is a simple board game, for two to four players, in which the players race their four coins from start to finish according to dice rolls. Like other cross and circle games, Ludo is derived from the Indian game Pachisi, but simpler. The game and its variants are popular in many countries and under various names

1.1 PURPOSE

The purpose of this application is to provide the indoor game service .The user can install this amazing application on his/her system (PC) and can get all feature of this application. It contain two to four player .this game is played by adults and children's as well. It is also known as board game .The benefits of board game for kid? Some are obvious kids enjoy them and board games are opportunities for families to play together. In addition social scientist has argued games teach lesson about getting along with others. Board game can teach the important social skills.

For example Games may encourage kids to

- Consider the concept of rule
- Practice of rules
- Reason about moral problem

1.2 SCOPE:-

It is use as an indoor game. In contemporary Indian scenario it is popular game in most of the Indians families. It helps us to connect the children to their native game. It may be developed as smart phone game. So that it can be easily accessible to the user.

Scope of our application will be limited to the functional the design of the original Ludo, along with our own selection of additions and modifications to the same. It will support saving and loading the states of games in progress, and also the ability to save or abandon a game in progress.

1.3 PROJECT OVERVIEW:-

Two, three, or four may play. At the beginning of the game, each player's tokens are out of play and staged in one of the large corner areas of the board in the player's color (called the player's yard). When able to, the players will enter their tokens one per time on their respective starting squares, and proceed to race them clockwise around the board along the game track (the path of squares not part of any player's home column). When reaching the square below his home column, a player continues by racing tokens up the column to the finishing square. The rolls of a cube die control the swiftness of the tokens, and entry to the finishing square requires a precise roll from the player. The first to bring all their tokens to the finish wins the game. The others often continue play to determine second-, third-, and fourth-place finisher.

2. OVERALL DESCRIPTIONS:-

2.1 PROJECT PERSPECTIVE:-

According to our specifications, a successful implementation of Ludo will allow its user to do all of the following:

- Start a game
- choose between different options
- turn off/on sound
- Roll a die
- Battle with a foe (combined with a mathematical problem)
- move a piece, surrender
- show help menu
- Set number of players
- Exit
- Show time

In summary, our overall criteria are to have a fully functioning Ludo game in terms of core functionality, extended by our own features.

2.2 MAIN FEATURES:-

A player shall be able to:

- Be able to set options
- Turn game music on or off
- Turn game sound effects on or off
- Select how many human players to play against (4 in all),
- If multiplayer is selected. Select color and name.
- Do a turn. During his/her turn, a player may:
 - Roll a dice. Move his/her pieces

2.3 GENERAL CONSTRAINTS:-

Hardware Requirements

The game is lights enough so it can be played on all PC operating system supporting platforms.

Software Requirement's

i. Performance

Game running on system should have maximum throughput and maximum -utilization.

ii. Scalability

As this game is for personal computers so it is scalable with personal computer OS platform.

iii. Reliability

As this game is for PC so it is reliable with PC OS platform.

3.SPECIFIC REQUIREMENT:-

3.1 EXTERNAL INTERFACE SPECIFICATION

3.1.1 USER INTERFACE

3.1.2 HARDWARE INTERFACE:-

It does not require any specific hardware for execution, it does not have any direct hardware interface.

3.1.3 SOFTWARE INTERFACE:-

It requires C compiler and its text editor for execution because it is built on C programming language as a stand-alone application.

3.1.4 COMMUNICATION INTERFACE

The communication between the different parts of the system is important since they depend on each other. However, in what way the communication is achieved is important for the system and is therefore handled by the underlying operating system for personal computers.

3.2 FUNCTIONAL REQUIREMENT:-

These are the functional requirements of our application.

A player shall be able to:

- Be able to set options
- turn game music on or off
- turn game sound effects on or off
- select how many players to play (4 in all),
- If number of players is selected. Select color and name.
- Do a turn. During his/her turn, a player may:
 - roll a dice. Select a piece to move in accordance the numbers come on the dice.
 - depending on the rules do battle with another player.
- Load/save a game in progress:

If a game is saved, make a copy of the current State of the game. If a game is loaded, set it's state to the target loaded State.

3.3 DESIGN CONSTRAINTS:-

3.2.1 USE CASE DIAGRAM:-

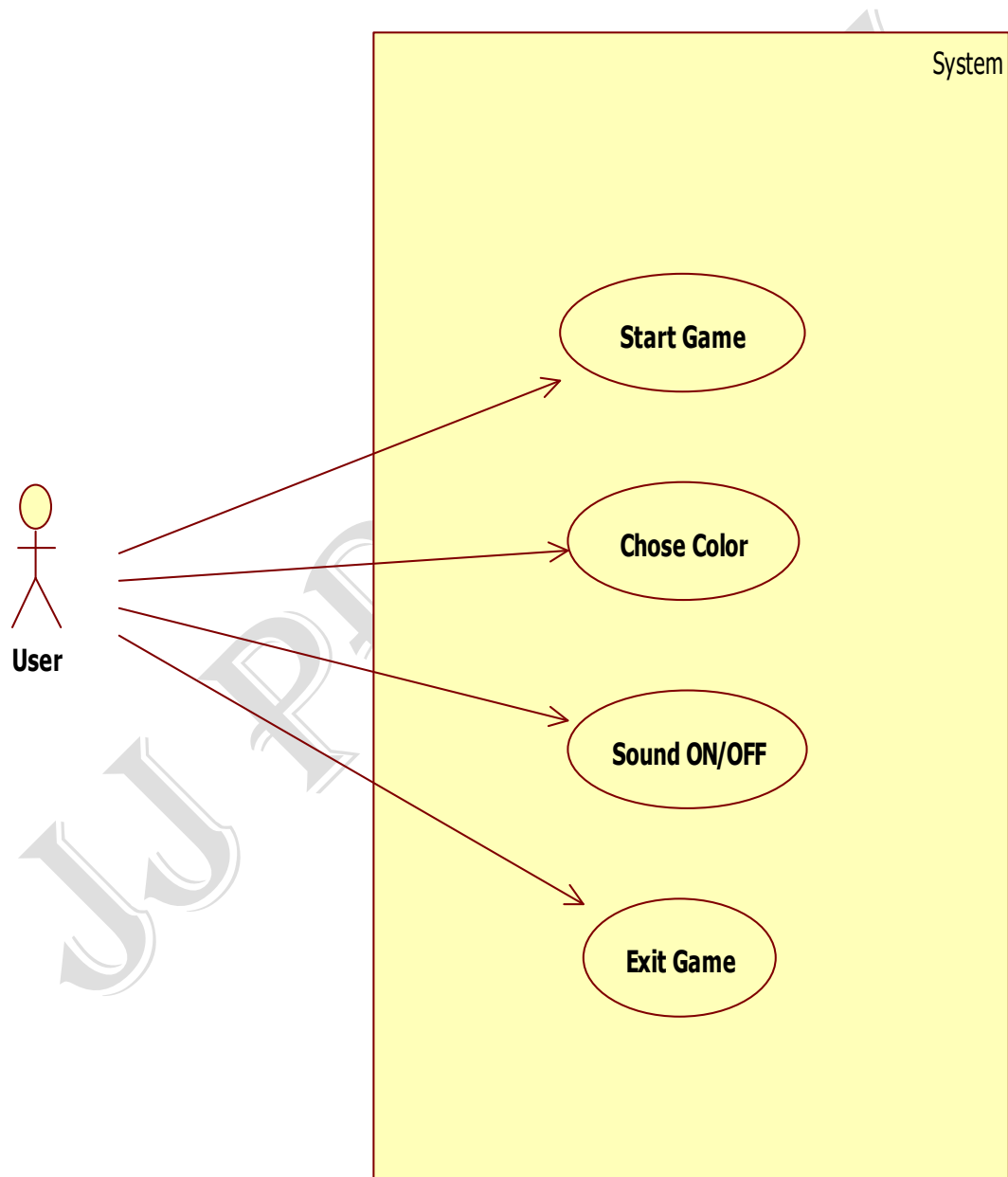


Fig: Use Case DIAGRAM

ACTIVITY DIAGRAM:-

Start game

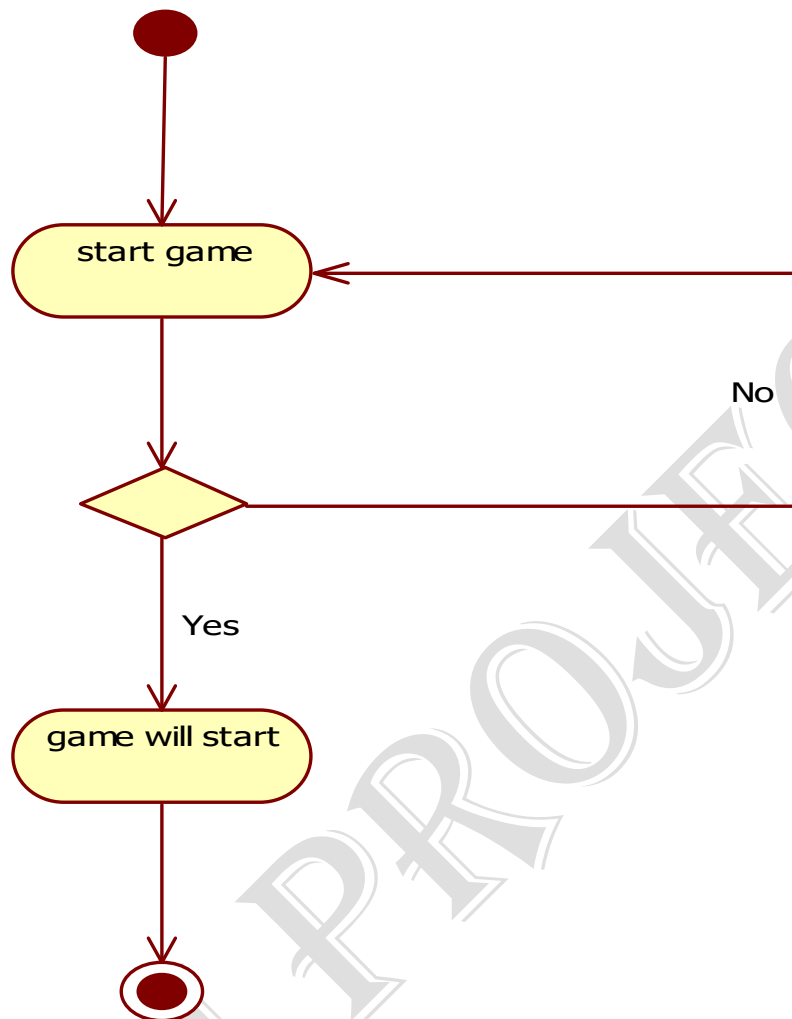


Fig: Start Game

ACTIVITY DIAGRAM:-

Choose Color

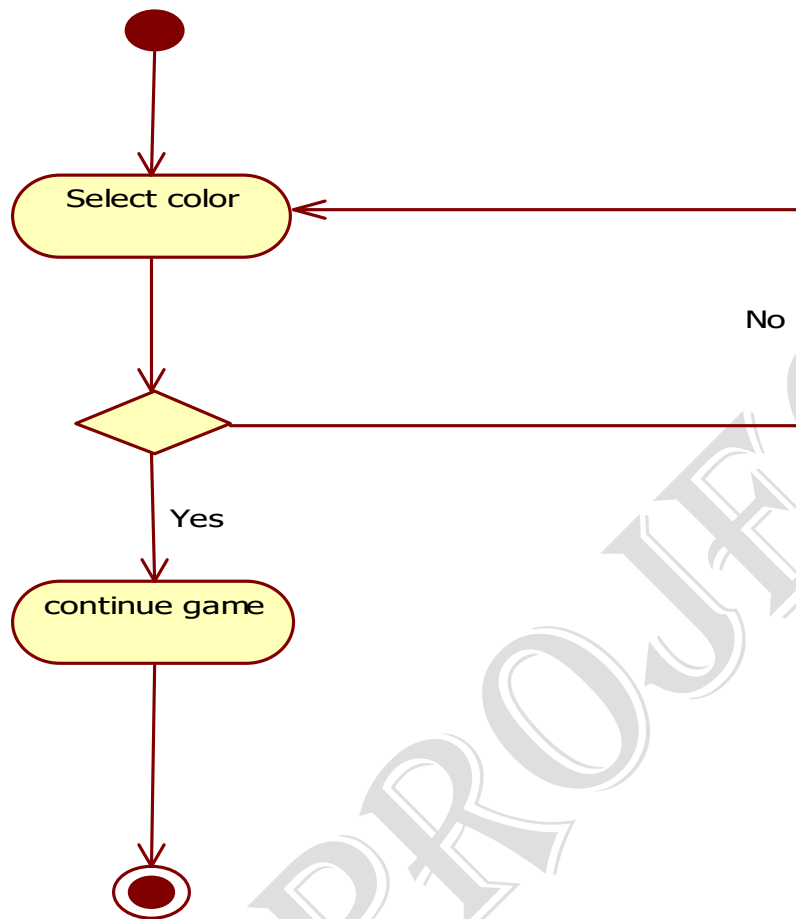


Fig: Choose Color

ACTIVITY DIAGRAM:-

Sound ON/OFF

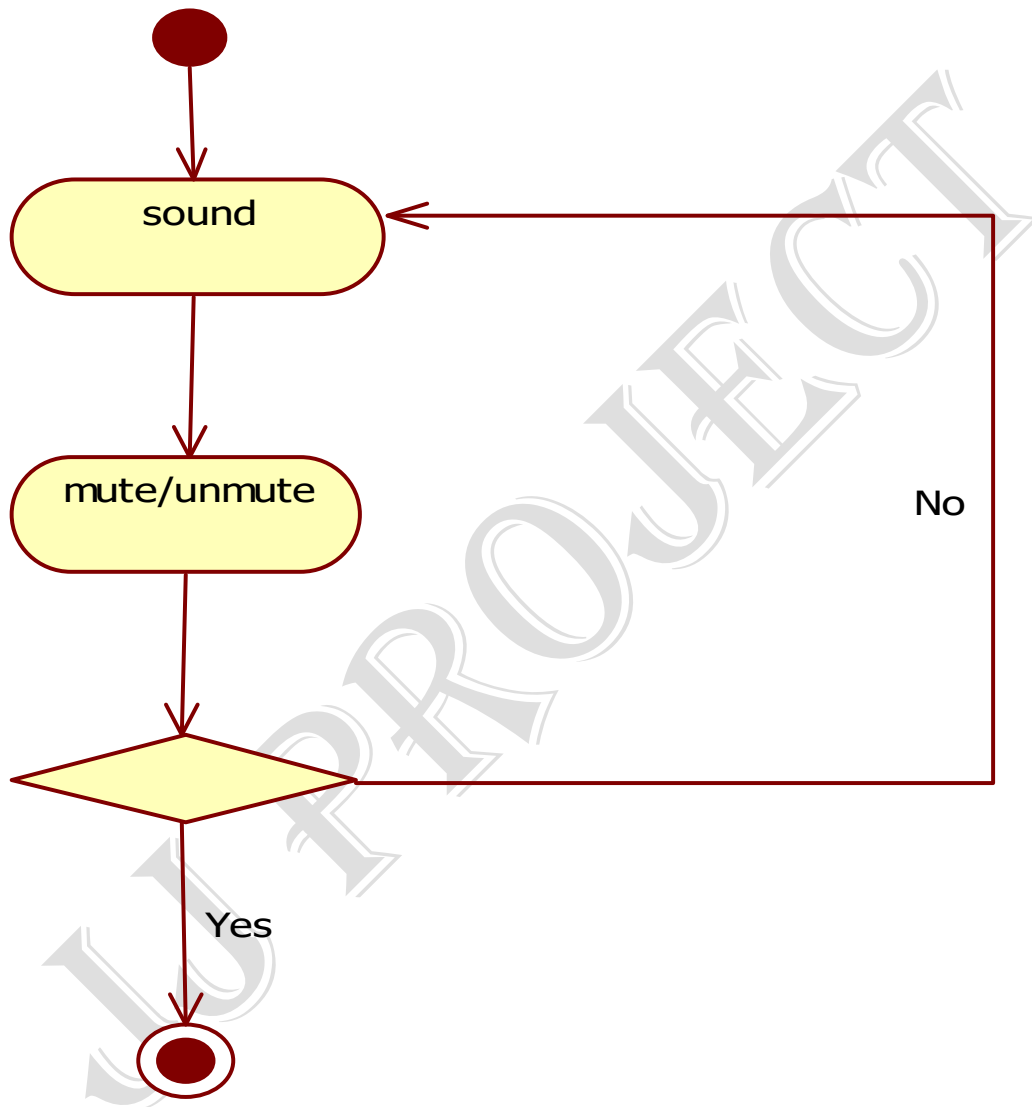


Fig 3.4 Sound ON/OFF

Activity Diagram

Exit Game

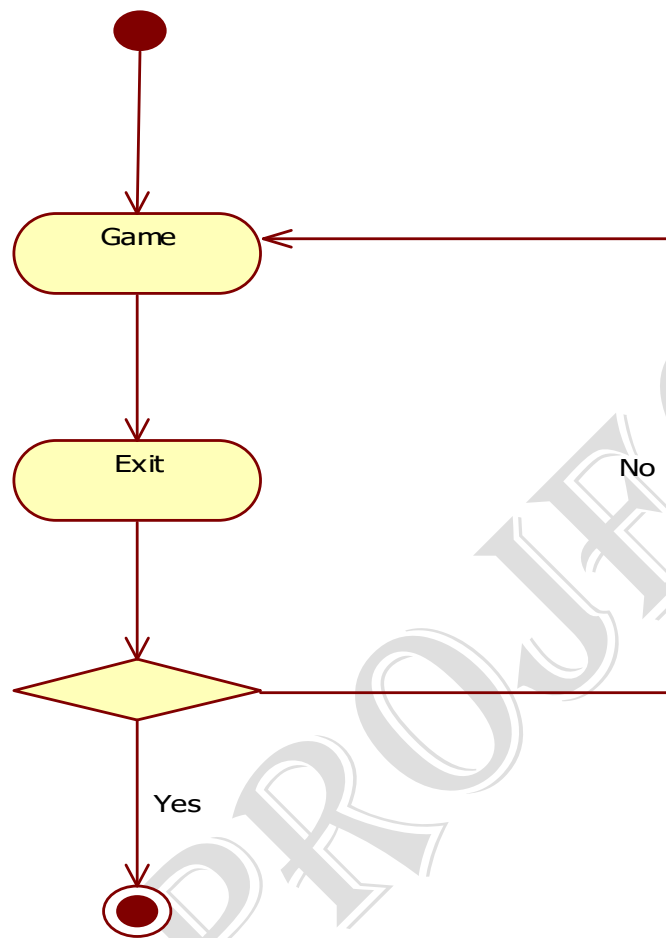


Fig: - Exit Game

3.3 Sequence Diagram

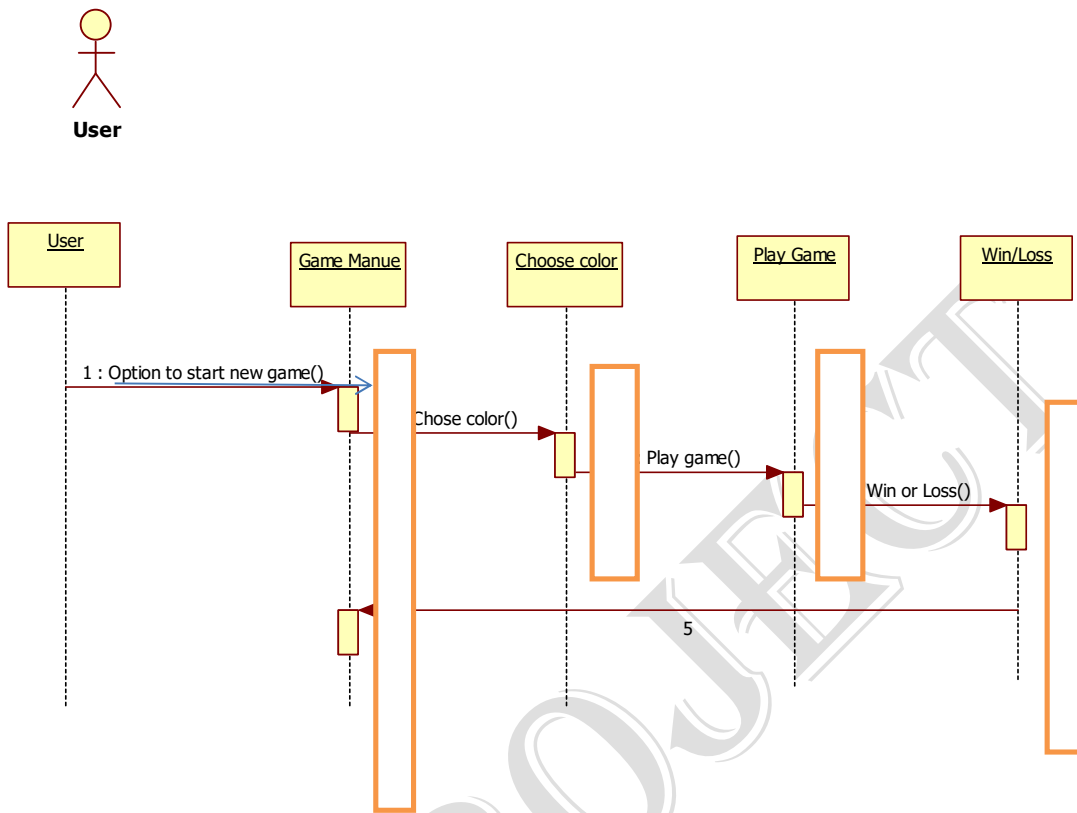


Fig 3.5 Sequence Diagram

3.4 Collaboration Diagram

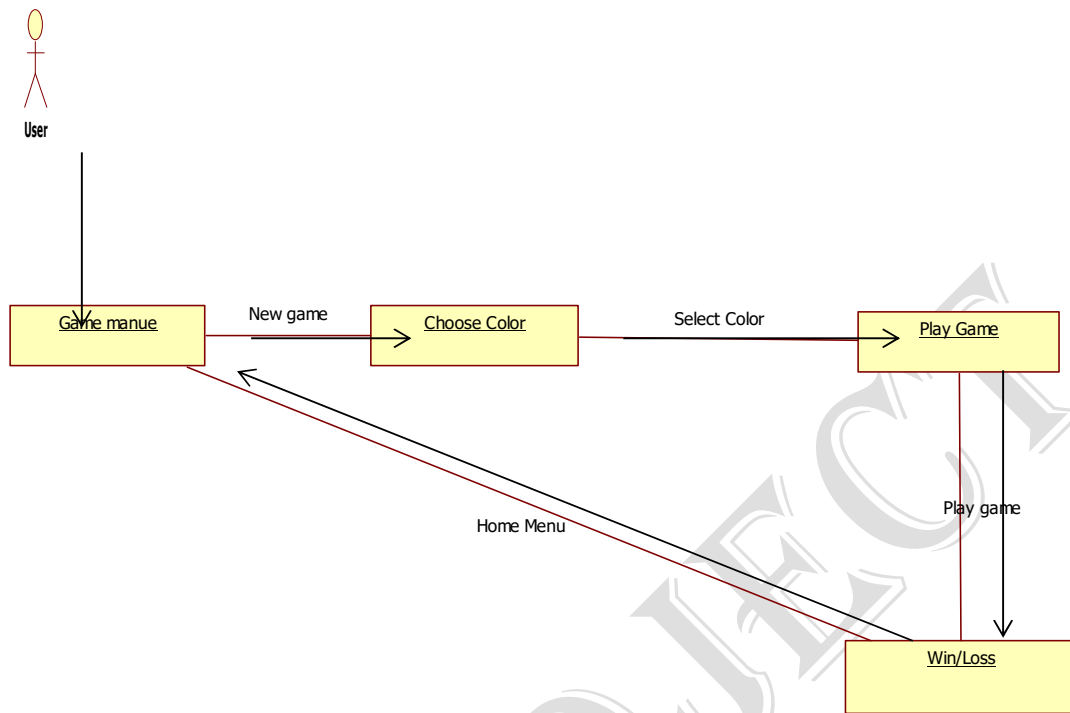


Fig 3.6 Collaboration Diagram

3.5 Deployment Diagram

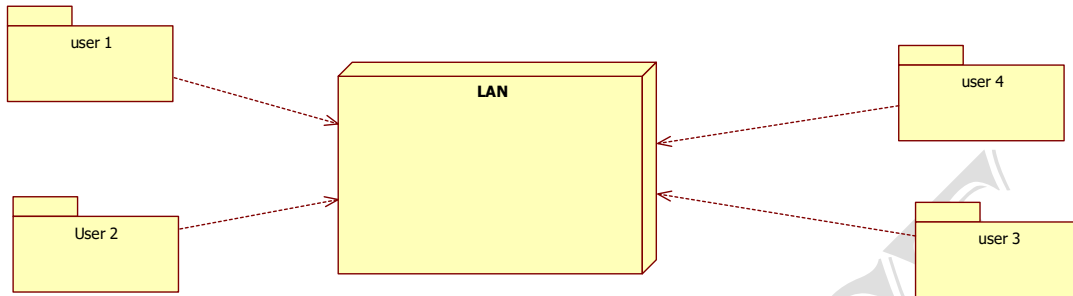


Fig 3.7 Deployment Diagram