The User Requirements Specification document contains requirements which the application will meet and the features it will possess.

User Requirements Specification

Fight The Landlord

Mengchuan Liu

Jiefan Lin

Jianfei Feng

Jiaqi Ni

Wen Zhang

Kalina Petrova

Contents

[1. Introduction 2](#_Toc462254972)

[2. Game Introduction 2](#_Toc462254973)

[1. Players, Cards, and Deal 2](#_Toc462254974)

[2. Valid cards types (combinations) 2](#_Toc462254975)

[3. Auction 3](#_Toc462254976)

[4. Play rules 3](#_Toc462254977)

[5. Example of expected cooperation 3](#_Toc462254978)

[6. Scoring 4](#_Toc462254979)

[3. Functional Requirements (M: Mandatory HD: highly desired) 5](#_Toc462254980)

[General Features 5](#_Toc462254981)

[4. System Requirements 6](#_Toc462254982)

[Hardware 6](#_Toc462254983)

[Software 6](#_Toc462254984)

[5. Non-functional Requirements 7](#_Toc462254985)

# Introduction

The User Requirements Specification document contains requirements which the application will meet and the features it will possess. Every software application which is to be developed needs to meet three kinds of functional requirements, system requirements and non- functional requirements.

Non-functional requirements of an application have to deal with its quality aspects such as how difficult it is to use it, how efficient, is it reliable and how maintainable is it.

# Game Introduction

Fight the Landlord (Dou Di Zhu) is a climbing game primarily for three players. In the game, one player, the “landlord”, plays alone and the others form a team. The landlord’s aim is to be the first to play out all his cards in valid combinations, and the ‘Farmer’ team wins if any one of them manages to play all their cards before the landlord. The game is very popular all over China and is also extensively played online.

## Players, Cards, and Deal

This game uses a 54-card pack including two jokers, red and black. 4 suits (Spade, Heart, Club, Diamond) with 13 cards in each.

The cards rank from high to low:

**-Red joker, black joker, 2, A, K, Q, J, 10, 9, 8, 7, 6, 5, 4, 3.**

Suits are irrelevant (no ranking between suits).

Cards are shuffled randomly, distributed to players counter-clockwise around the table until each player has 17 cards. Each player can only see his own cards. The last three cards are visible to everyone and belong to the landlord. So landlord has a total of 20 cards.

## Valid cards types (combinations)

In this game, there are thirteen types of combination that can be played:

1. Single card - ranking from three (low) up to red joker (high) as explained above

2. Pair - two cards of the same rank, from three (low) up to two (high)

3. Triplet - three cards of the same rank

4. Triplet with an attached card - a triplet with any single card added, for example 6-6-6-8. These rank according to the rank of the triplet - so for example 9-9-9-3 beats 8-8-8-A

5. Triplet with an attached pair - a triplet with a pair added, like a full house in poker, the ranking being determined by the rank of the triplet - for example Q-Q-Q-6-6 beats 10-10-10-K-K.

6. Sequence - at least five cards of consecutive rank, from 3 up to ace - for example 8-9-10-J-Q. Twos and jokers cannot be used.

7. Sequence of pairs - at least three pairs of consecutive ranks, from 3 up to ace. Twos and jokers cannot be used. For example 10-10-J-J-Q-Q-K-K.

8. Sequence of triplets - at least two triplets of consecutive ranks from three up to ace. For example 4-4-4-5-5-5.

9. Sequence of triplets with attached cards - an extra card is added to each triplet. For example 7-7-7-8-8-8-3-6. The attached cards must be different from all the triplets and from each other. Although triplets of twos cannot be included, a two or a joker or one of each can be attached, but not both jokers.

10. Sequence of triplets with attached pairs - an extra pair is attached to each triplet. Only the triplets have to be in sequence - for example 8-8-8-9-9-9-4-4-J-J. The pairs must be different in rank from each other and from all the triplets. Although triplets of twos cannot be included, twos can be attached. Note that attached single cards and attached pairs cannot be mixed - for example 3-3-3-4-4-4-6-7-7 is not valid.

11. Bomb - four cards of the same rank. A bomb can beat everything except a rocket, and a higher ranked bomb can beat a lower ranked one.

12. Rocket - a pair of jokers. It is the highest combination and beats everything else, including bombs.

13. Quadplex set - there are two types: a quad with two single cards of different ranks attached, such as 6-6-6-6-8-9, or a quad with two pairs of different ranks attached, such as J-J-J-J-9-9-Q-Q. Twos and jokers can be attached, but you cannot use both jokers in one quadplex set. Quadplex sets are ranked according to the rank of the quad. Note that a quadplex set can only beat a lower quadplex set of the same type, and cannot beat any other type of combination. Also a quadplex set can be beaten by a bomb made of lower ranked cards.

## Auction

There is an auction to determine which player will be the landlord, and play alone against the other two. The possible bids are 1, 2 and 3. The player who is the first to bid is randomly picked. Each player, in turn, may either pass or bid higher than the highest bid so far. If everyone passes the hand is thrown in and there is a new deal. If there is a bid, the bidding continues counter-clockwise, each player passing or bidding higher than the previous bidder until there are two consecutive players pass or someone bids 3, which ends the auction since it is the highest possible bid. The final and highest bidder is the landlord. This player now picks up the three face-down cards from the middle, for a total of 20 cards.

## Play rules

The landlord plays first, and may play any single card or any legal combination. The played cards can be seen by everyone. Each subsequent player in counter-clockwise order must either pass (play no card) or beat the previous play by playing a higher-ranking combination of the **same number of cards** and **same type**. (No ranking between different types or different number of cards).

Example: 2 aces cannot beat 3 tens.

Note: If the combination type is a triplet, QQQ can beat 999, QQQ-77 can beat 999-55, QQQ-7 can beat 999-5. But QQQ cannot beat 999-55 or 999-5, the number of cards and combination have to be same.

The play continues around the table for as many circuits as necessary until two consecutive players pass. The person who played the last card(s) begins again, leading any card or a legal combination.

Note that passing does not prevent you from playing on a future turn.

## Example of expected cooperation

Player A (the landlord) leads 3-3-3 to get rid of some low cards, player B passes, player C plays 5-5-5, player A plays K-K-K and player B plays A-A-A. C and A pass, so B can start again with anything. He leads a single 4.

Note B could have played his aces on his first turn, but preferred to pass to give his partner a chance to get rid of some cards. C will now play if possible, so as not to give the landlord (A) a free chance to lead again. Having beaten A’s second play, B leads a low card to give C the choice of playing another unwanted card of putting the landlord under pressure by playing a high card.

## Scoring

If the landlord runs out of cards first he has won, and each opponent pays him the amount of the bid - 1, 2 or 3 units - provided that no bomb or rocket was played. If one of the other two players runs out before the landlord, the landlord loses and must pay the amount of the bid **to each opponent**. For each occasion when any player played a bomb or rocket, the payment for the hand is doubled. So for example in a hand in which two bombs and a rocket were played, a player who bid 3 will win 24 points from each opponent for going out first, or pay 24 to each opponent if another player goes out first.

# Functional Requirements (M: Mandatory HD: highly desired)

## General Features

|  |  |  |
| --- | --- | --- |
|  | Requirement | Preference |
| Account system | user can register an account | M |
| registered user is able to login | M |
| Lobby | a user is able to create room | HD |
| a user is able to select an existing room and join | HD |
| the user who creates the room can start the game when the other two users are ready | HD |
| Gameplay | the player can reselect cards he wants to play | M |
| randomly give the chance which player can bid the first | M |
| determine landlord | M |
| shuffle cards | M |
| ranking all the cards player is current holding (High - Low) | M |
| when a player gets rid of cards, check the combination is valid and show a proper warning message when the combination is not beatable | M |
| a player is able to pass this turn and keep current cards | M |
| a player is able to let the system to suggest him get which card(s) out | M |
| When the player selects a card, the card will move a bit up | M |
| Players can disable/enable the background music. | M |
| Players can disable/enable the sound effect. | M |
| Scoring System | the system will record total score of every user | HD |
| every day a user can have 100 free scores | HD |
| if the score is 0 or below, the user cannot continue to play this day. | HD |

# System Requirements

## Hardware

|  |  |
| --- | --- |
| Requirement | Preference |
| The following are the recommended Server hardware requirements,the system can be run in a virtual environment and on a physical server. In both cases these are the recommended requirements. It is recommended to run 2 servers, whereby the service and database run on a different system.  **CPU:** At least a dual core CPU  **RAM:** On Windows Server 2003/Windows XP at least 2GB and Windows Server 2008/Windows 7 and later user at least 4GB  **HDD:** At least 10GB of free hard drive space  **GPU:** The server has no requirements for the GPU since the server will not use it for computing. | M |
| The following requirements are recommended for the client application  **CPU:** At least a dual core CPU  **RAM:** At least 2GB of RAM  **HDD:** At least 1GB of free hard drive space  **GPU:** A system with Integrated graphics should work fine | M |

## Software

|  |  |
| --- | --- |
| Requirement | Preference |
| The software requirements for the server follows. It is recommended to run a version of Windows Server to run the server. The minimum requirement for Operating system is Windows Server 2003. But it is recommend to run at least Windows Server 2008R2. On the server it need to have .NET 4.5 installed. | M |
| The recommended software requirements for the clients is any Windows Operating System with .NET4.5 installed | M |

# Non-functional Requirements

Of course when using an application the things that can bother us or make us happy are not always related to the product’s functionality. What about Usability, Reliability, Performance and Maintainability?

* Usability – The application will not be intrusive and will have an easy to understand GUI. The application’s interaction will be intuitive.

Usability will be tested using actual user interaction. We will present a group of people that represent our user base with the demo products and monitor how they navigate through the application. Using these observations we will determine which parts are alright as they are and which need modification. We will conduct these user tests during our testing period of each iteration.

Since we are creating a logical card game application, it would be beneficial to present the rules of the game and the different functionalities to the user using a small tutorial to get them used to the application.

* Performance – This is an important aspect in any Simulation. The application will not take up unnecessary computer memory or computing time. It will match with efficiency standards allowing the simulations to run smoothly.

We will try to test the application on various machines and devices to ensure that the performance is not constrained by any specific requirement. (CPU, GPU, RAM)

* Reliability – All the simulations from the software will be accurate and precise. They will provide with an ideal situation of traffic control that is not influenced by forces not stated by the client. Any outside tempering with the system will be prevented and prohibited.

During our user testing we will also make sure that the users get the desired results and that there are no exceptions to the rules.

We will also try to cover the code with unit testing as much as possible to make sure no bugs occur due to unexpected input.

* Maintainability – The software will be provided with error protection protocols. The software will be written in proper standards, making it easy to be accessed from third parties.
* Extensibility – We will make sure that the code is as clean as possible and structured in a nice way so that there won’t be any difficulty when adding additional features.

To ensure that our architecture and code are extensible and maintainable, we will try to research on all the possible functionalities that could be implemented into our application and take those into consideration when designing the system.