**REQUIREMENT SPEFICATION DOCUMENT FOR OOP PROJECT**

Alperen BAYAR 150115031

Büşra YAĞCI 150115057

Oğuzhan YİĞİT 150115052

**VISION**

We designed a monopoly game with Java Language which is object oriented programming language. Java Language allowed us creating working methods and variables, then re-use all or part of them without compromising security. Monopoly is a board game played by two to four players.  It is played on a board with spaces. Players roll two six-sided dice to move around the game board, buying and trading properties, and developing them with houses and hotels. Players collect from their opponents, with the goal being to drive them into bankruptcy.

**PROBLEM STATEMENT AND SCOPE**

To start the game the program takes the total number of players and names from the user. The game is moving forward with the dice. When it comes to a player, two dice are thrown then the player move around the board according to the throw of a pair of dice.

In the game board there are 40 blocks. Each side of the square board is divided into 10 small rectangles representing specific properties, lucky card, jail, tax administration and various other places. At the start of the game, each player is given a fixed amount of play money. All users have 1500$.

Any player who lands on an unowned property may buy it, but, if he or she lands on a property owned by another player, rent must be paid to that player. If the player doesn’t have enough money and if the player have another lands the player have to sell their lands till his money be enough to can buy current location or pay the current location rent. If the player's money is still not enough despite the sale, the player will go bankrupt.

A player who lands on a goToJail property get into jail. In the other round he is out of jail and he is sent to the other corner of the board.

A player who lands on a Tax Administration property have to pay tax. If the player doesn’t have enough money and if the player have another lands the player have to sell their lands till his money be enough to pay tax. If the player's money is still not enough despite the sale, the player will go bankrupt.

A player who lands on a Lucky Card property can face many different situations. His money can be increased or decreased. He can be sent to Jail, tax administration or other cities. The game controls the player not having a lucky card property twice in a row.

A player continues to travel around the board until he or she is bankrupt. Bankruptcy results in elimination from the game. The last player remaining on the board is the winner.

**SYSTEM REQUIREMENTS**

We use Intellij because;

* Easy commit and push
* Creating UML diagrams easily
* Autocompletion Github for group works.

The version we use Java 8.

**STAKEHOLDERS**

* Alperen BAYAR
* Büşra YAĞCI
* Oğuzhan YİĞİT
* Murat Can GANİZ(TEACHER)
* Berna ALTINEL

**GLOSSORR OF TERMS**

* Block: Blocks which pawn travels.
* Dice: Determine how far the proceed.
* LocationCity: Blocks which have city.
* LocationJail: Blocks holding jail entrances and exits.
* LocationLuckyCard: If you are in this block you must use lucky card.
* LocationTaxAdmin: Tax administration property. If you are in this block you must pay tax.