REQUIREMENT ANALYSIS

VISION

This project is a monopoly game simulation. It takes json file as input. The file includes game parameters and, the game is built according to these parameters. In the beginning of the game each player rolls dices and they are sorted in descending order according to the sum of face values of dices they rolled. They play the game according to this order. When a player takes turn, the player will roll the dices, and move on the board based on the face values of dices. If player stops on the tax square, the player pays tax in amount tax square payment. The player who lost all money that have, leaves the game. Finally, the winner is announced when single player stays in the game.

SCOPE

The game has 40 squares on the board. Tax squares are placed randomly according to taken tax square count. In the staring of the program, each user balance has staring balance. Each user takes money if he/she passes from the go square. The player who is not bankrupt when all other user is bankrupted is the winner of the game.

SYSTEM CONSTRAINTS

The program starts to run with reading a json file which contains a tax square count, starting balance, players name, tax payment, salary of go square, and to arrange the speed of the simulation it takes times for sleep after turn and sleep after piece move in seconds. After reading json file the game starts. If simulation cannot start, then simulation terminates.

First of all, to include the libraries the steps given below is needed to follow:

File -> Project Structure -> Modules -> Dependencies -> Export -> JARs or directories -> Lib -> OK

PROJECT MEMBERS

- Anıl ALTAY
- Ayten BİNAY
- Merve YAYIN

TERMINOLOGY

- **Go Square:** Go square is the first square on the board when players passed this square, they get salary amount of go salary.
- **Go Salary:** Go salary is that how much money a player gains when the player passes from the go square.
- **Tax Square:** Tax square is the square that when the players passed pay tax amount of tax square payment.
- **Tax Square Payment:** Tax square payment is that how much money a player pays when the player passes from the tax square.
- **Piece:** Pieces represent player of the game. Every player is represented by a unique piece and movement on the board occurs with the changing on the current location of the piece.
- **Current Location:** Current location is that index of the square that piece is on.
- **Dice Value:** Dice value is a number in between 1 and 6 which is the face value of the dice after the dice is rolled.
- **Balance of Player:** Balance of player is the amount of the money that a player has.
- **Tax Square Count:** Tax square count is that how many tax squares are on the board.

- **Starting Balance:** Starting balance is that how much money a player has when the game starting.
- **Sleep After Turn:** The time period in seconds after every single player takes turn.
- **Sleep After Piece Move:** The time period in seconds after every single piece move.
- **Bankrupt:** If a player loses all the money that he/she have then the player is bankrupt.

USE CASE

- When the Monopoly simulation runs it takes some parameters from the json file.
- Game starts with rolling dices of every player to takes turn; they are ordered by the face values of the dices in descending order.
- According to this order every player takes its turn.
- The player who takes turn rolls the two dices and according to the sum of the face values of dices, the player moves.
- When a player lands on a tax square, the player pays tax.
- When a player both lands and passes on a go square, gets the salary specified.
- When a player lost all the money, he/she has, the player is bankrupt, and this player leaves the game.
- When all the players expect one of them is bankrupt, the game is finish and the player who is left is the winner.