**BMS COLLEGE OF ENGINEERING**

**(Autonomous College under VTU)**

**Bull Temple Road, Basavanagudi, Bangalore – 560019**



A project report on

***“*MONOPOLY*”***

Submitted in partial fulfillment of the requirements for the award of degree

**BACHELOR OF ENGINEERING**

**IN**

**INFORMATION SCIENCE AND ENGINEERING**

**By**

Sunag P, 1BM19IS162

Sumith Hegde, 1BM19IS161

Shreyam Pandey , 1BM19IS153

Under the guidance of

**Sindhu K**

**Department of Information Science and Engineering**

**2020-21**

**BMS COLLEGE OF ENGINEERING**

**(Autonomous College under VTU)**

**Bull Temple Road, Basavanagudi, Bangalore – 560019**



**Department of Information Science and Engineering**

CERTIFICATE

This is to certify that the project entitled “MONOPOLY” is a bona-fide work carried out by **Sunag P(1BM19IS162),Sumith Hegde(1BM19IS161),Shreyam Pandey(1BM19IS153)**in partial fulfillment for the award of degree of Bachelor of Engineering in **Information Science and Engineering**  from **Visvesvaraya Technological University, Belgaum** during the year **2020-2021**. It is certified that all corrections/suggestions indicated for Internal Assessments have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the Bachelor of Engineering Degree.

**Signature of the Faculty Signature of the HOD**

**Name and Designation Name and Designation**

Table of Contents

Abstract

1. **Introduction ……………………………………………………………………………….**
   1. Purpose ………………………………………………………………………………
   2. Scope …………………………………………………………………………………

**Software Requirement Specification …………………………………………………….**

* 1. Software Requirements ……………………………………………………………….
  2. Hardware Requirements ………………………………………………………………
  3. Functionality……………………………………………………………………………

**ER Diagram ……………………..………………………………………………………..**

**Relational Schema ……………………………………………….……**

**Implementation……………………………………………………………………………**

* 1. SQL queries ………………………………………………………………………….
  2. Connecting back-end to front-end ……………………………………………………….

**Conclusion**………………………………………………………………………………..

* 1. Summary………………………………………………………………………………..
  2. Limitations……………………………………………………………………………..
  3. Further enhancements…………………………………………………………………..

**References**……………………………………………………………………………..

**Appendix: Snapshots** …………………………………………………………………

**ABSTRACT :**

Location based games have seen the translation of popular board games into mixed reality settings through the integration of latest technologies.

This report explores modifying the game of Monopoly from a board game to a PC based game utilizing the latest technologies to engage players.

The objective of the game is simple : Make as much money as you can, and drive your opponents into bankruptcy. Roll the dice, buy a property, pay rent, pass go, and collect money. Repeat.

**Introduction :**

**1.1 Purpose :**

* Games certainly serve some important purpose or they would not be as popular as they are. The obvious purpose they serve is as entertainment.
* Game-based learning is one of the modern pedagogical approaches beginning to gain attention in education.
* The field of Medicine highlighted the usefulness of specifying problem scenarios

which are placed within a game framework.

* Monopoly is derived from The Landlord's Game created in the United States in 1903 as a way to demonstrate that an economy that rewards wealth creation is better than one where monopolists work under few constraints, and to promote the economic theories of Henry George—in particular his ideas about taxation, this game was invented.

**1.2 Scope :**

* Over 480 million players have already played monopoly. Moreover monopoly, it is not just an entertaining game for time abuse but also a teaching tool, it helps people to develop logic and thinking.
* Currently physical boards are available for this game and have been in use for a long time.
* Since people in recent times prefer technological tools rather than physical boards it is necessary to modernize it through digital media/platforms to keep this exciting game alive.

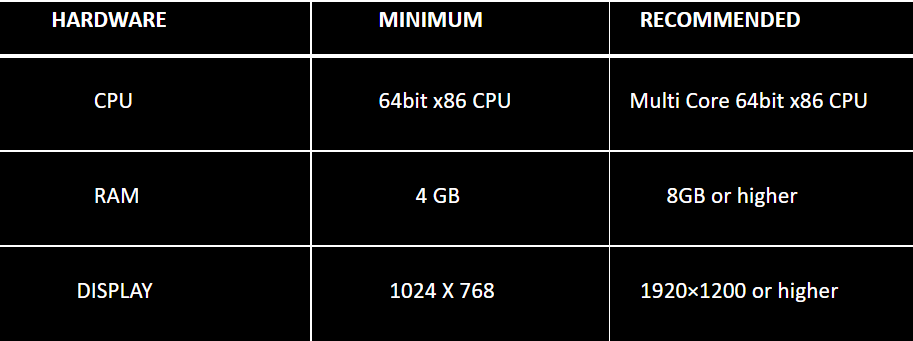
Hence we chose this as our project.

**Software Requirement Specification :**

**1.3**  **Software requirements :-**

* XAMPP
* MySQL
* JAVA Development Kit
* Visual Studio

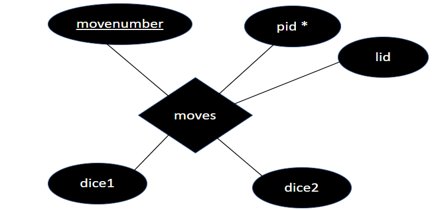
**1.4 Hardware requirements :-**

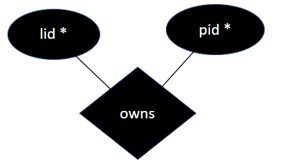
****

**Functionality :-**

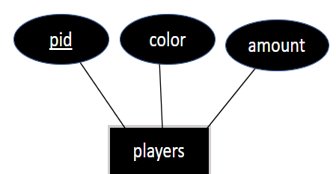
* Dynamically fetches data from the database and displays onto the screen.
* Dynamic insertions into the tables throughout the game.
* Each move of players is captured and dynamically updated to tables.
* Dynamic updations of all money transactions within the game.

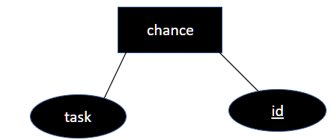
**Relationships and Attributes :**

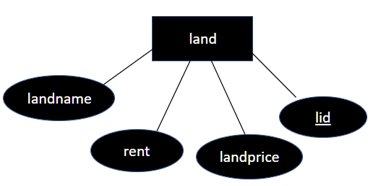


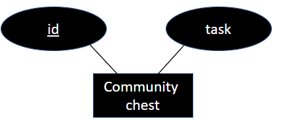


**Entities and Attributes :**

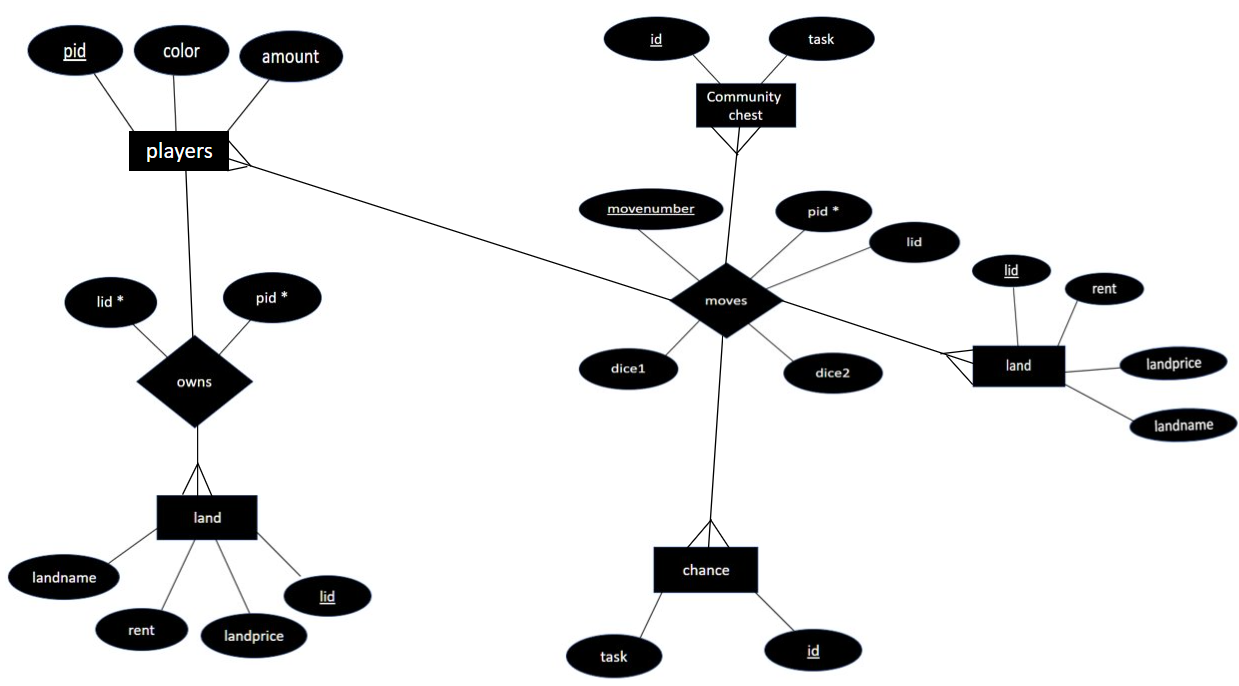




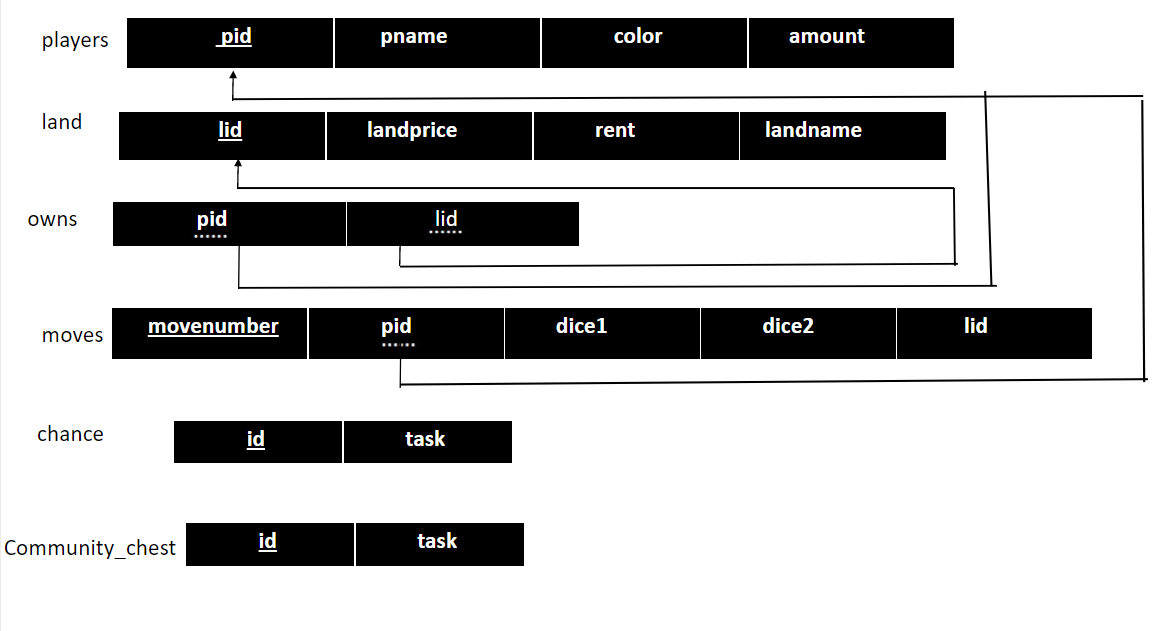




**ER Diagram:-**

****

**Schema :-**



**Implementation :-**

**1.6 Connecting Backend to Frontend** :-

try{

Class.forName("com.mysql.jdbc.Driver");

Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/monopoly","root","");

String query="query to be executed ";

con.prepareStatement(query).execute();

con.close();

}

catch(Exception ex){ System.out.println(ex);}

**1.7 SQL Queries :-**

**1.7.1 Table creation Queries:**

* CREATE TABLE PLAYERS(PID INT PRIMARY KEY,PNAME VARCHAR(20),COLOR VARCHAR(20) UNIQUE,AMOUNT INT);
* CREATE TABLE LAND(LID INT PRIMARY KEY , LANDPRICE INT , RENT INT , LANDNAME VARCHAR(20));
* CREATE TABLE OWNS(PID INT FOREIGN KEY , LID INT FOREIGN KEY);
* CREATE TABLE MOVES(MOVENUMBER INT PRIMARY KEY PID INT FOREIGN KEY , DICE1 INT DICE2 INT,LID INT);
* CREATE TABLE COMMUNITYCHEST(CCID INT PRIMARY KEY,TASK VARCHAR(100));
* CREATE TABLE CHANCE(CID INT PRIMARY KEY,TASK VARCHAR(100));

**1.7.2 Inserting into players table once the game begins:**

try{

Class.forName("com.mysql.jdbc.Driver");

Connection con;

con=DriverManager.getConnection("jdbc:mysql://localhost:3306/monopoly","root","");

String query = " insert into players (pid, pname, color ,amount)" + " values (?, ?, ?,?)";

String query3 = "delete from moves";

con.prepareStatement(query3).execute();

String query2 = "delete from owns";

con.prepareStatement(query2).execute();

String query1 = "delete from players";

con.prepareStatement(query1).execute();

PreparedStatement x2 = con.prepareStatement(query);

PreparedStatement x1 = con.prepareStatement(query);

x1.setInt(1, 0);

x1.setString(2,x);

x1.setString(3,"green");

x1.setInt(4, 10000);

x2.setInt(1, 1);

x2.setString(2,y);

x2.setString(3,"red");

x2.setInt(4, 10000);

x1.execute();

x2.execute();

con.close();

}catch(Exception \_e){ System.out.println(\_e);}

**1.7.3 Updating players table when a player buys a land:**

try{

Class.forName("com.mysql.jdbc.Driver");

Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/monopoly","root","");

Statement stmt=con.createStatement();

String query = "select landprice from land where lid="+p[turn];

String query1 = "insert into owns values("+turn+","+p[turn]+");";

String query2 = "select pid from owns where lid="+p[turn];

// con.prepareStatement(query).execute();

ResultSet rs=stmt.executeQuery(query);

while(rs.next())

{

cost=rs.getInt(1);

}

ResultSet rs1=stmt.executeQuery(query2);

while(rs1.next())

{

play=rs1.getInt(1);

}

if(play>-1)

{

System.out.println("already sold");

}

else

{

con.prepareStatement(query1).execute();

}

con.close();

}catch(Exception ex){ System.out.println(ex);}

int amt=0;

if(play>-1)

{

amt=m[turn];

}

else

{

m[turn]=m[turn]-cost;

amt=m[turn];

currentStatus.setText("<html><h2>You bought "+c.C[p[turn]]+"</h2>");

switch(turn)

{

case 0 : owns[p[turn]].setBackground(Color.GREEN);

owns[p[turn]].setOpaque(true);

break;

case 1 : owns[p[turn]].setBackground(Color.red);

owns[p[turn]].setOpaque(true);

break;

}

}

if(turn==1){

players[turn].setText(y+"("+m[turn]+")");

try{

Class.forName("com.mysql.jdbc.Driver");

Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/monopoly","root","");

String query = "update players set amount ="+amt+" where pid=1";

con.prepareStatement(query).execute();

con.close();

}catch(Exception ex){ System.out.println(ex);}

}

if(turn==0){

players[turn].setText(x+"("+m[turn]+")");

try{

Class.forName("com.mysql.jdbc.Driver");

Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/monopoly","root","");

String query = "update players set amount ="+amt+" where pid=0";

con.prepareStatement(query).execute();

con.close();

}catch(Exception ex){ System.out.println(ex);}

}

**1.7.4 Updating players table when a player gives rent to the owner of the land:**

if(owner[p[turn]]>-1&&owner[p[turn]]!=turn)

{

buy.setEnabled(false);

m[turn]=m[turn]-rent;

amt=m[turn];

m[play]=m[play]+rent;

what="<br>you paid Rs"+rent+" to "+name[play]+" as rent";

// what="";

}

**1.7.5 Queries related to Chance table:**

chance(task,turn,m,players); //function call for chance

switch(task)

{

……

case 102: m[turn]=m[turn]-250;

try{

Class.forName("com.mysql.jdbc.Driver");

Connection con;

con=DriverManager.getConnection("jdbc:mysql://localhost:3306/monopoly","root","");

String query="update players set amount="+m[turn]+" where pid="+turn;

con.prepareStatement(query).execute();

con.close();

players[turn].setText(name[turn]+"("+m[turn]+")");

}catch(Exception ex){ System.out.println(ex);}

break;

…….

}

**1.7.6 Queries related to CommunityChest table:**

communityChest(task,turn,m,players); //function call for chance

switch(task)

{

……

case 103: m[turn]=m[turn]-1000;

try{

Class.forName("com.mysql.jdbc.Driver");

Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/monopoly","root","");

String query="update players set amount="+m[turn]+" where pid="+turn;

con.prepareStatement(query).execute();

con.close();

players[turn].setText(name[turn]+"("+m[turn]+")");

}catch(Exception ex){ System.out.println(ex);}

break;

…….

}

**1.7.7 Dynamic update of moves table after each players turn:**

try{

Class.forName("com.mysql.jdbc.Driver");

Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/monopoly","root","");

String query1 = "insert into moves values("+moveNumber+","+turn+","+sumDice+","+p[turn]+")";

con.prepareStatement(query1).execute();

con.close();

}catch(Exception ex){ System.out.println(ex);}

**1.7.8 Retrieving value from table :**

try{

Class.forName("com.mysql.jdbc.Driver");

Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/monopoly","root","");

Statement stmt=con.createStatement();

String query = "select pid,rent from owns,land where owns.lid="+p[turn]+" and owns.lid=land.lid";

ResultSet rs=stmt.executeQuery(query);

while(rs.next())

{

rent = rs.getInt(2);

owner=rs.getInt(1);

}

con.close();

}catch(Exception ex){ System.out.println(ex);}

**Conclusion :**

**1.8 Summary:**

* Due to latest methods in software development,

existing games may not be as competitive as they are now.

* Therefore, it is very important to maintain the competitiveness of the game in order to achieve desired,tangible or intangible goals and there is already a well known approach called game updating.
* Almost every game is being updated in a defined time period.
* As mentioned previously, modern computer technology

is moving towards more innovative ideas .

* Therefore we came up with the idea of building this game with modern tools and providing an interactive interface.

**1.9 Limitations :**

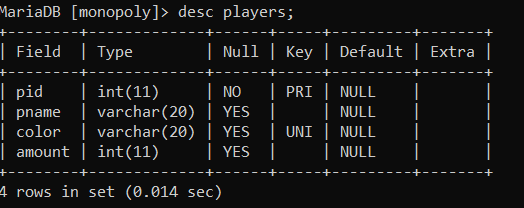
* The Game we have designed is restricted to a maximum of 4 players and a minimum of 2 players.
* A single player cannot play this against Computer (Bot)

**1.10 Further Enhancement :**

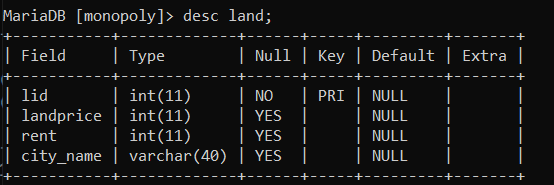
* Since Monopoly is a board game for two to eight players ,The Game we have designed is restricted to a maximum of 4 players this can be further increased to a maximum of 8 players.
* We can also enhance this by adding a feature where a single player can play against Bot

**Snapshots :**

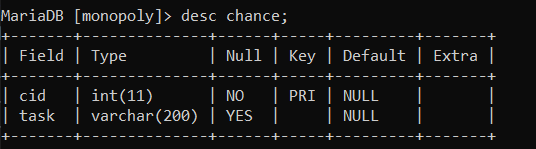
**Description of tables :**

****

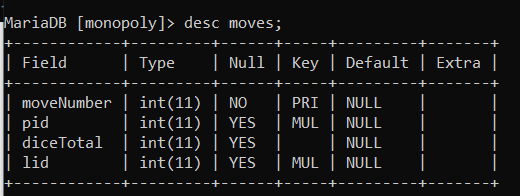
**Fig.1** Description of players table

****

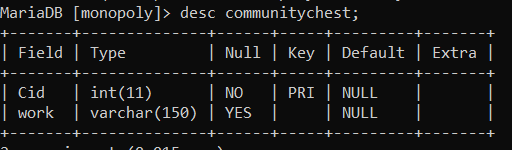
**Fig.2** Description of land table

****

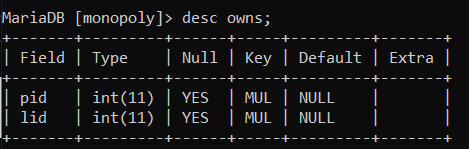
**Fig.3** Description of chance table

****

**Fig.4** Description of moves table

****

**Fig.5** Description of CommunityChest table

****

**Fig.6** Description of owns table

**Front end images :**

****

**Fig.7** When a user clicks on any city, information corresponding to that city will be fetched from database and will be displayed in info section

****

**Fig.8** When a player lands on chance dynamic updation from database onto the board

****

**Fig.9** When a player lands on some city owned by another player , rent will be debited from that player and will be credited to the owner