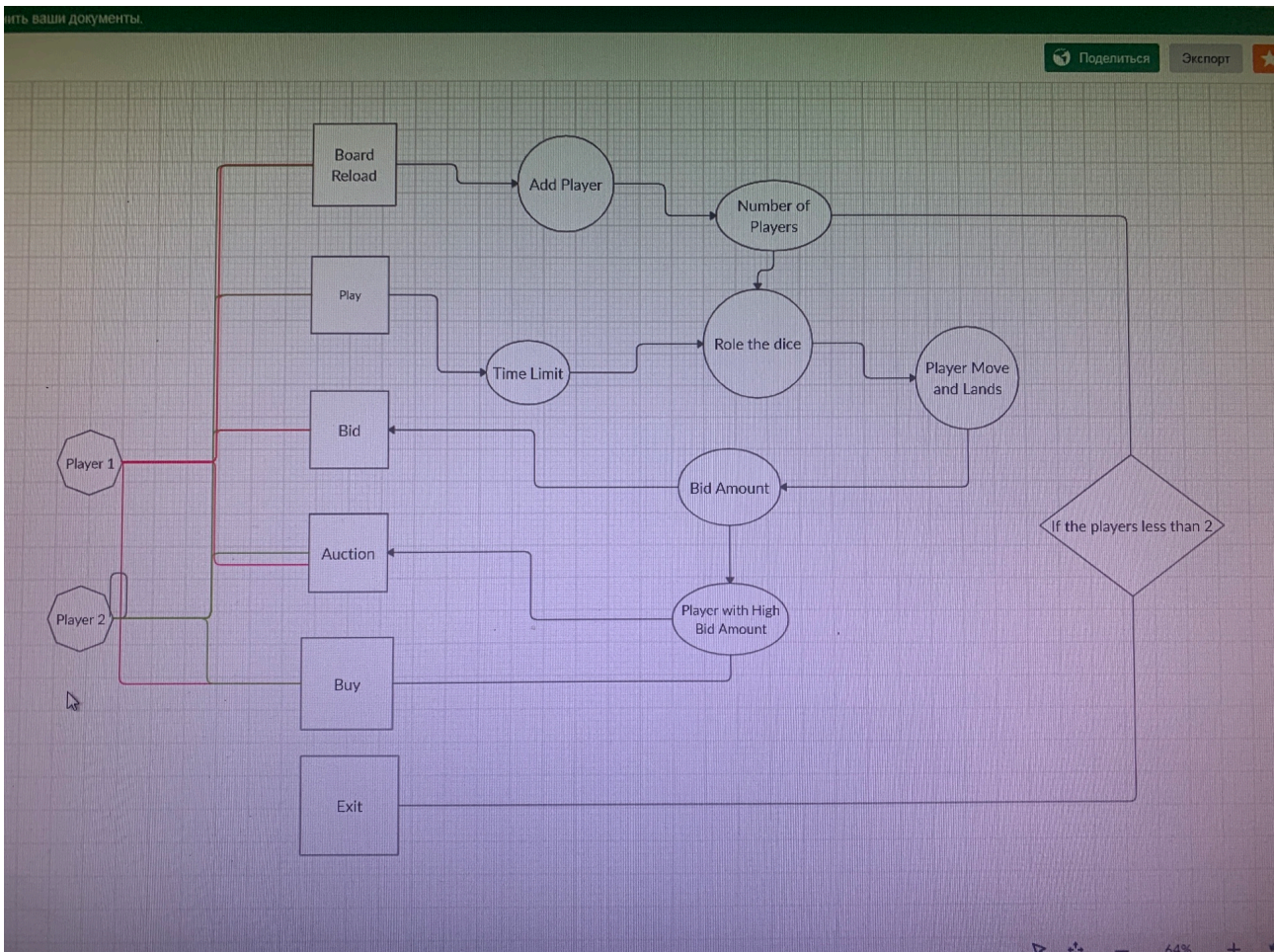


Design Documentation

1. High Level Designs :

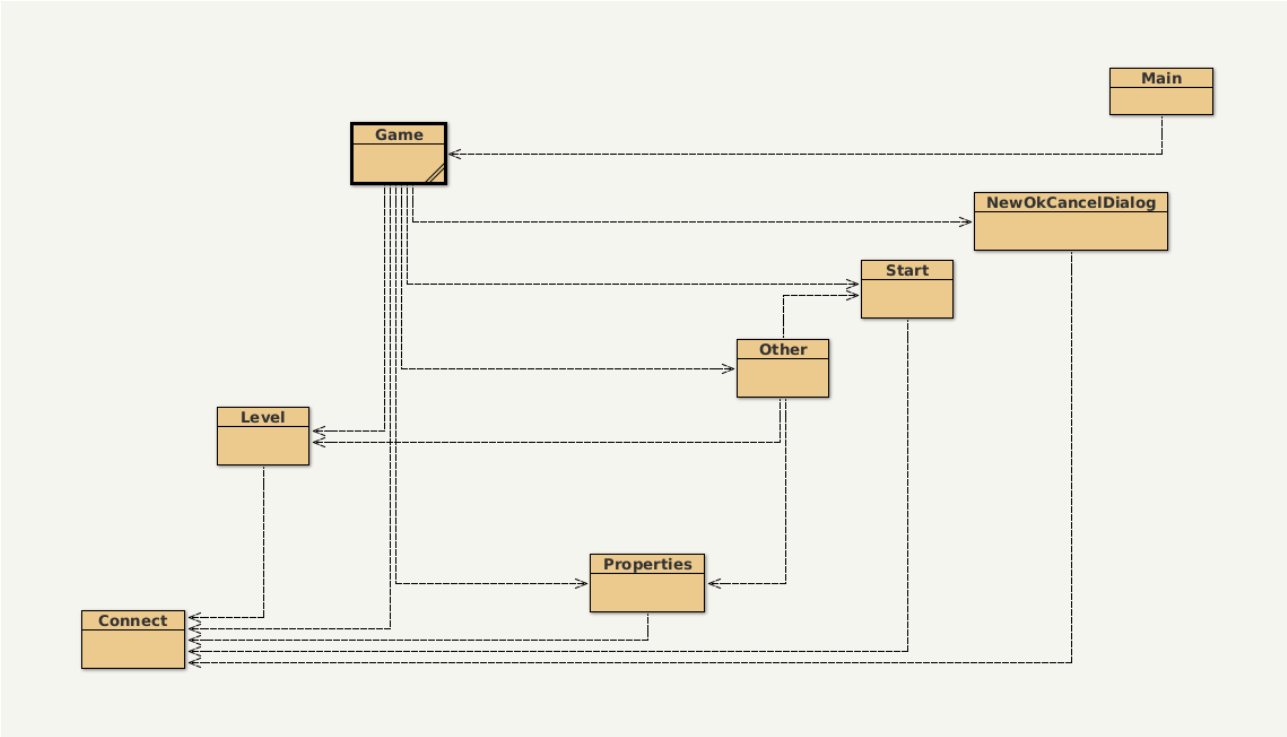
1.1 Overview



1.1 The use diagram clearly show us how the User will interface with program and also gives an idea step by step order that program will execute in order.

2. Low Level Designs :

2.1 Overview



2.2 Game:

Game dialog = new Game(new javax.swing.JFrame(), true);

2.3 Level:

<pre>static int playerLevel(int player) {</pre>	
	<pre>String cars = null;</pre>
	<pre>int level = 0;</pre>
	<pre>String sql = "SELECT * FROM level WHERE player = "+ player;</pre>

2.4 Other:

<code>public class</code>	
<code>Other {</code>	
	<code>/**</code>
	<code>* Opportunity cards in the Property Tycoon</code>
	<code>* @param value the value of card</code>
	<code>* @param player the player</code>
	<code>* @return Messages of Opportunity cards</code>
	<code>*/</code>
	<code>static String opportunity(int value, int player) {</code>
	<code>String message = "";</code>
	<code>double amount = Start.amount(player);</code>
	<code>amount = amount - 200;</code>
	<code>Properties.update(player, amount);</code>
	<code>switch (value){</code>
	<code>case 1:</code>
	<code>amount = amount + 50;</code>
	<code>Properties.update(player, amount);</code>
	<code>message += "Description : Bank pays you divided of</code>
	<code>£50\n";</code>
	<code>message += "Action : Bank pays player £50\n";</code>
	<code>break;</code>

2.5 Start:

<code>class Start {</code>	
	<code>public Start(){</code>
	<code>}</code>
	<code>@SuppressWarnings("empty-statement")</code>
	<code>/**</code>
	<code>* Returns all players participating in the game</code>
	<code>* @return all players participating in the game</code>
	<code>*/</code>
	<code>public static String Users(){</code>
	<code>String item = "Player List \t";</code>

	String sql = "SELECT * FROM players";
--	---------------------------------------

2.6 Connect

public class	
Connect {	
	/**
	* Connect to a sample database
	* @return
	* @throws SQLException
	*/
	static Connection connect() throws
	SQLException {
	String url = "jdbc:sqlite:database.db";
	Connection conn =
	DriverManager.getConnection(url);
	return conn;
	}

2.7 Properties:

public class	
Properties {	
	/**
	* Init the database
	* @return true
	*/
	public static boolean init(){
	// SQLite connection string
	String url = "jdbc:sqlite:database.db";
	// SQL statement for creating a new table
	// String sql1 = "DROP TABLE IF EXISTS
	properties";
	String sql4 = "DROP TABLE IF EXISTS
	rent_property";

	String sql5 = "DROP TABLE IF EXISTS owner";
--	---