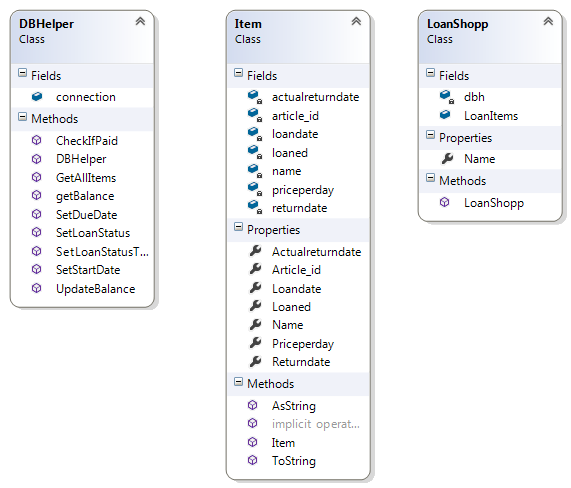
# Loaning Items App

Overview:

The loaning items application is used for loaning certain items which the event offers for a certain period of time. When the app is starded you can click on the load button which will load all the available items from the database and display them in a listbox, after that the employee can press the scan button which will activate the RFID Reader and find the person that wants to loan items’s detail via his bracelet string. The next step is to choose an item which he wants to loan and upon clicking the loan button it will interact with the database and set the user\_id to the corresponding one and the loanstatus of the item to 1 (true). After choosing a date from the datetimepicker and clicking the button loan the price will be calculated based on the days which the item will be loaned for times the price per day of the item and it will be substracted from the balance, if the balance of the visitor is insufficient a corresponding messagebox will appear. When the user wants to return an item he can give it back and the employee can choose the item which will be returned based on the article\_id and it when clicking the button return that will use the method SetLoanStatusToFalse from the DBHelper.

This is the UML-diagram for the application



As visible above the DBHelper class has a method (GetAllItems) for retrieving all the items from the database and inserting them in a list of items with their respective properties. The other methods are mostly self-explanatory, the most important methods are the SetLoanStatus() and SetLoanStatusToFalse(), the first of which updates the loanstatus to 1 and sets the user\_id to the corresponding one via the visitors table using the bracelet\_id and the latter one is just nullifying all the values of the user\_id and the loanstatus, when the method is used, which is when returning back an item. The properties of the Item class are all based on the database and are assigned values from it.

public void SetLoanStatus(int articleid, string bracelet\_id)

{

string sql = "UPDATE loanitems SET LoanStatus = 1, USER\_ID = (SELECT visitors.USER\_ID FROM visitors WHERE BRACELET\_ID = ?bracelet\_ID LIMIT 1) WHERE ARTICLE\_ID = ?Article\_id;";

MySqlCommand command = new MySqlCommand(sql, connection);

command.CommandText = sql;

command.Parameters.AddWithValue("?Article\_id", articleid);

command.Parameters.AddWithValue("?bracelet\_ID", bracelet\_id);

try

{

connection.Open();

MySqlDataReader reader = command.ExecuteReader();

}

catch (Exception exc)

{

MessageBox.Show(exc.ToString());

}

finally

{

connection.Close();

}

}

public void SetLoanStatusToFalse(int articleid)

{

string sql = "UPDATE loanitems SET LoanStatus = 0, USER\_ID = 0 ,RETURNDATE = NOW() WHERE ARTICLE\_ID = ?Article\_id;";

MySqlCommand command = new MySqlCommand(sql, connection);

command.CommandText = sql;

command.Parameters.AddWithValue("?Article\_id", articleid);

try

{

connection.Open();

MySqlDataReader reader = command.ExecuteReader();

}

catch (Exception exc)

{

MessageBox.Show(exc.ToString());

}

finally

{

connection.Close();

}

}