## Приложение 1

public partial class MainForm : Form

{

private Connector Connector;

int ShortIdOfSelectedTestCase;

string user\_login;

string run\_id;

public MainForm()

{

InitializeComponent();

Connector = new Connector();

using (var loginForm = new LoginForm())

{

var result = loginForm.ShowDialog();

if (result == DialogResult.OK)

{

this.user\_login = loginForm.login;

this.run\_id = loginForm.runId; RefreshBasicInfo();

}

}

}

private void RefreshBasicInfo()

{

if (user\_login == "admin" && run\_id == "root")

{ btnBulkEdit.Enabled = true; }

else { btnBulkEdit.Enabled = false; }

}

private void buttonRun\_Click(object sender, EventArgs e)

{

int testcase\_short\_id = this.ShortIdOfSelectedTestCase;

try

{

TestCase tcase = Connector.GetTestCase(testcase\_short\_id);

TestSteps tsteps = Connector.GetStepsForTestCase(testcase\_short\_id);

using (var form = new TestCaseForm(tcase, tsteps))

{

var result = form.ShowDialog();

if (result == DialogResult.OK)

{

string passed = form.result;

Connector.InsertRunResult(this.user\_login, this.run\_id, tcase.ShortId, "2023-13-13", "2023-13-13", passed);

}

}

}

private void buttonLogOut\_Click(object sender, EventArgs e)

{

using (var loginForm = new LoginForm())

{

var result = loginForm.ShowDialog();

if (result == DialogResult.OK)

{

this.user\_login = loginForm.login;

this.run\_id = loginForm.runId;

}

}

}

}

## Приложение 2

public class Connector

{

private SQLiteConnection \_connection;

public Connector()

{

var project\_folder = Directory.GetParent(Directory.GetCurrentDirectory()).Parent.FullName;

var pathToSQLiteDatabase = $@"URI=file:{project\_folder}\Database\cases.sqlite";

\_connection = new SQLiteConnection(pathToSQLiteDatabase);

\_connection.Open();

}

public List<TestCase> GetAllTestCases()

{

List<TestCase> cases = new List<TestCase>();

string statement = $"select \* from view\_testcases";

var cmd = new SQLiteCommand(statement, \_connection);

SQLiteDataReader reader = cmd.ExecuteReader();

while (reader.Read())

{

TestCase testcase;

testcase.ShortId = reader.GetInt32(0);

testcase.Title = reader.GetString(1);

testcase.Idea = reader.GetString(2);

testcase.ExpectedResult = reader.GetString(3);

testcase.QtyOfSteps = reader.GetInt32(4);

cases.Add(testcase);

}

return cases;

}

public List<TestSuite> GetAllTestSuites()

{

List<TestSuite> suites = new List<TestSuite>();

string statement = $"select \* from view\_testsuites";

var cmd = new SQLiteCommand(statement, \_connection);

SQLiteDataReader reader = cmd.ExecuteReader();

while (reader.Read())

{

TestSuite suite;

suite.ShortId = reader.GetInt32(0);

suite.Title = reader.GetString(1);

suite.QtyOfTestCases = reader.GetInt32(2);

suites.Add(suite);

}

return suites;

}

public TestSteps GetStepsForTestCase(int testcase\_short\_id)

{

string statement = $"select step, action from testcase\_has\_steps where testcase\_short\_id={testcase\_short\_id} order by step asc";

var cmd = new SQLiteCommand(statement, \_connection);

SQLiteDataReader reader = cmd.ExecuteReader();

TestSteps steps = new TestSteps();

while (reader.Read())

{

int step = reader.GetInt32(0);

string action = reader.GetString(1);

steps.Add(step, action);

}

return steps;

}

}

## Приложение 3

public partial class ReportForm : Form

{

private Connector connector;

private string user\_login; private string run\_id;

private void comboBox1\_SelectedValueChanged(object sender, EventArgs e)

{ switch (comboBox1.SelectedIndex) {

case 0: FormReportOne(); break;

case 1: FormReportTwo(); break;

case 2: FormReportThree(); break;

case 3: FormReportFour(); break;

case 4: FormReportFive(); break; } }

private void FormReportOne()

{

string report = "Report:\tTestcases passed in current run\n";

report += $"Login:\t{this.user\_login}\n";

report += $"Run id:\t{this.run\_id}\n";

report += $"\n";

Dictionary<TestCase, string> passed\_cases = connector.GetResultsOfTestCases(user\_login, run\_id);

foreach (TestCase tcase in passed\_cases.Keys)

{ report += $"{passed\_cases[tcase]}\t{tcase.Representation}\n"; }

report += $"\nTotal testcases passed: {passed\_cases.Count} testcases";

richTextBoxReport.Text = report; }

private void FormReportTwo()

{

Dictionary<string, int> passed = connector.GetPairOfLoginAndExpectedResult("Passed");

Dictionary<string, int> blocked = connector.GetPairOfLoginAndExpectedResult("Blocked");

Dictionary<string, int> total = new Dictionary<string, int>();

foreach (var pair in passed)

{ total[pair.Key] = pair.Value;

if (!blocked.ContainsKey(pair.Key)) { [pair.Key] = 0; }

}

foreach (var pair in blocked)

{

if (!passed.ContainsKey(pair.Key)) { passed[pair.Key] = 0; }

if (total.ContainsKey(pair.Key))

{ total[pair.Key] += pair.Value;

}

else { total[pair.Key] = 0; } }

string report = "Report:\tAnalyze percentage of passed testcases\n";

report += $"Login:\t{this.user\_login}\n";

report += $"Run id:\t{this.run\_id}\n";

report += $"\n";

report += $"Login: passed + blocked = total (passed / total \* 100%)\n";

foreach (var pair in total)

{

string login = pair.Key;

int one = passed[login];

int two = blocked[login];

double percentage = (double)one / (one + two);

report += $"- {login}: {one} + {two} = {one+two} ({String.Format("{0:P2}.", percentage)})\n";

}

richTextBoxReport.Text = report;

}

## Приложение 4

private void FormReportThree()

{

List<(string, int)> finished = connector.GetQtyOfCasesFinished();

Dictionary<string, int> total = new Dictionary<string, int>();

Dictionary<string, int> qty\_of\_runs = new Dictionary<string, int>();

foreach (var pair in finished)

{

if (total.ContainsKey(pair.Item1)) {total[pair.Item1] += pair.Item2;}

else { total[pair.Item1] = pair.Item2; }

if (qty\_of\_runs.ContainsKey(pair.Item1)){qty\_of\_runs[pair.Item1]+=1;}

else { qty\_of\_runs[pair.Item1] = 1; }

}

Dictionary<string, float> average = new Dictionary<string, float>();

foreach (var item in total)

{ average[item.Key] = (float)qty\_of\_runs[item.Key] / total[item.Key]; }

string report = "Report:\tAverage testcases finished by user\n";

report += $"Login:\t{this.user\_login}\n";

report += $"Run id:\t{this.run\_id}\n";

report += $"\n";

report += $"Login: total / qty of runs = average\n";

foreach (var pair in total)

{ string login = pair.Key;

report+=$"- {login}: {total[login]}/{qty\_of\_runs[login]} = {average[login]}\n";

}

richTextBoxReport.Text = report;

}

private void FormReportFour()

{

List<TestCase> testcases = connector.GetAllTestCases();

string report = "Report:\tTest steps for all testcases\n";

report += $"Login:\t{this.user\_login}\n";

report += $"Run id:\t{this.run\_id}\n";

report += $"\n";

foreach (TestCase tcase in testcases)

{

report += $"{tcase.Representation}\n";

TestSteps steps = connector.GetStepsForTestCase(tcase.ShortId);

for (int i = 1; i<=steps.Count; i++)

{ report += $"{i}. {steps[i]}\n"; }

report += "\n";

}

richTextBoxReport.Text = report;

}

private void FormReportFive()

{

List<TestSuite> suites = connector.GetAllTestSuites();

string report = "Report:\tTestcases in test suites\n";

report += $"Login:\t{this.user\_login}\n";

report += $"Run id:\t{this.run\_id}\n";

report += $"\n";

foreach (TestSuite suite in suites)

{

report += $"{suite.Representation}\n";

List<TestCase> cases = connector.GetTestCasesInTestSuite(suite.ShortId);

int count = 1;

foreach (TestCase tcase in cases)

{ report += $"{count}. {tcase.Representation}\n";

count++; }

report += "\n"; }

richTextBoxReport.Text = report;

}

## Приложение 5

public partial class LoginForm : Form

{

public string login;

public string runId;

public LoginForm()

{

InitializeComponent();

this.login = "";

this.runId= "";

}

private void buttonLogin\_Click(object sender, EventArgs e)

{

this.login = this.textBoxLogin.Text;

this.runId = this.textBoxRunId.Text;

if (AllFieldsFilled() == true)

{

this.DialogResult = DialogResult.OK;

this.Close();

return;

}

// in case fields are not filled

using (ErrorLoginForm errorForm = new ErrorLoginForm())

{ errorForm.ShowDialog(); }

}

private void textBoxRunId\_KeyUp(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)

{ buttonLogin\_Click(sender, e); }

}

private void textBoxLogin\_KeyUp(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)

{ textBoxRunId.Focus(); }

}

private bool AllFieldsFilled()

{

if (this.login.Length > 0 && this.runId.Length > 0)

{ return true; }

return false;

}

}