# 

**Department of Computer Science**

**Spring-2017**

**CSC 565**

Final Documentation-Cell Project

**By:**

Shivakrishna Bujunoori - F00442378

Mohammed Awais Ansari – F00367304

Contents:

[Client](#_Toc437595958) Details----------------------------------------------------------------------------------------------3

Abstract-----------------------------------------------------------------------------------------------------4

Project Title------------------------------------------------------------------------------------------------4

Requirements and Specifications--------------------------------------------------------------------5

Existing System ------------------------------------------------------------------------------------------6

Problems with Existing Solution-----------------------------------------------------------6

Proposed System------------------------------------------------------------------------------------------7

Solution over Existing Constraints: -------------------------------------------------------7

Solution Strategy-----------------------------------------------------------------------------------------8

Benefits of Proposed System: ---------------------------------------------------------------8

Prototype and Design flow-----------------------------------------------------------------------------9

Implementation------------------------------------------------------------------------------------------11

Testing and Sample test Cases-----------------------------------------------------------------------15

Future Enhancements---------------------------------------------------------------------------------20

Bibliography-----------------------------------------------------------------------------------------------21

# Client Details:

**Jessica Loppnow***and* **VanHart Rochelle M**

Assistant Professors of Natural Sciences

Email: [Jessica.Loppnow@cuw.edu](mailto:Jessica.Loppnow@cuw.edu) and [Rochelle.VanHart@cuw.edu](mailto:Rochelle.VanHart@cuw.edu)

Address: 12800 N. Lake Shore Dr.

Mequon, WI 53097

# Developers’ Details:

1. Shivakrishna Bujunoori – F00442378

**Email:** Shivakrishna.bujunoori@cuw.edu

2. Mohammed Awais Ansari – F00367304

**Email:**

**Abstract of the problem**

The Science Department has a working cell project application that is used for helping students learn different processes that cells go through. The application needs some improvements and bug fixes. The science department faculty will be the clients for this project.

**Detailed Problem Definition**

Cell project application is an interactive game and learning system. Where user or students will login to get access the system and play the games and scenarios and submit it to their professor. Student will be given with a random scenario in which they will be asked to pick the parameters and proceed to the next step. Based on the decisions of the steps in the scenario will lead to the end of the scenario. The completion of the scenario will be the end of the game for that session.

Professor will be evaluating the decisions of the students and give them grade. Current system has some problems and needs enhancements from the administration, student and application perspectives.

**Requirements were categorized in 3 parts:**

1) Requirements from the perspective of administrator (admin users for application)-Catagory1

2) Requirements from the perspective of Students (one set of users for application)-Catagory2

3) Captured Application Issues - such as User Interface requirements-Catagory3

**Category 1 Requirements:-**

**For Admin activities:**

**R1:** New feature for Admin to be able to create dynamic scenarios with priority of ranking.

- **Ranking:** While application provides a random scenario to student, Instructors wants to give the priority to some of the scenarios which means a level of priority so that most of the students will get a chance to act to the scenario.

**R2:** For the existing scenarios, giving the feature of “Edit” on the content of entire scenario and updating it to the database.

-Confirmed and client cancelled this requirement due to business reasons.

**R3:** Missing entire section of activity which is in the intermediate level of scenario3. There are existing 4 scenarios available in the existing application. In the Third scenario some portion of the activity is not loading or issue with loading to the gam which means a deadlock scenario while playing that scenario3.

**Category 2 Requirements:**

**R4**:- R**andomize available scenarios:** When a student enters into the application to, in existing application student have to choose the available scenarios to play the game. Instead a randomly picked scenario would be shown to a student to play the game and finish it.

* To pick up a scenario from the bucket of available scenarios should be prioritized based on the ranking given to the scenario at the time of creating it. Giving ranking number to the scenario is listed under the administration requirement which is in category2 requirements.

**R5:-** Giving feature of adding comments at the time of taking an action in the steps of a scenario. For example, providing comment text box for selecting parameter in selection of available parameters (options) in the intermediate steps of a scenario.

**Category 3 Requirements:-**

**R6**:- From the admin and student login perspectives: User Interface changes for the login pages, home page screens, scenario backgrounds in terms of color, proper alignments of textual information, comment boxes alignments.

**System Description**

**System specifications:-**

**Front-end:**

HTML (Hypertext markup language)

CSS (Cascading styles sheets)

JavaScript

Json (JavaScript Object Notation)

Ajax (Asynchronous JavaScript and XML)

**Back-end:**

ASP.NET

**Database:**

Microsoft Sql Server

**Prototype and Design**

Individual requirements and respective designs are available as follows

**Requirement R1:** New feature for Admin to be able to create dynamic scenarios with priority of ranking.

- **Ranking:** While application provides a random scenario to student, Instructors wants to give the priority to some of the scenarios which means a level of priority (ranking number) to the scenario while creating reflects that in the process of picking scenario randomly gets high chance of probability to that scenario. Resulting to that most of the students will get a chance to act to the scenario.

**Flow chart:**

Authenticate

Admin Login

The username or password is incorrect.

Scenario – View, Edit & Delete

Home

Edit First name, Last name & Email Address

My Corner

Available scenarios

**+Add scenario**

Scenario

Content – View, Edit, Delete, Add image and Links to contents

Content

Name of the Student, Level, Comment and Assign Grade.

Grade

Displays Student Name, Allow Edit option and Delete existing Student

Manage Student

Old Password, New Password & Confirm Password

Change Password

Logout

**On click +Addscenario:-** admin will be prompted to enter scenario details ,selects the root content checkbox and at the end based on the priority of the scenario , admin will select either of the values listed below.

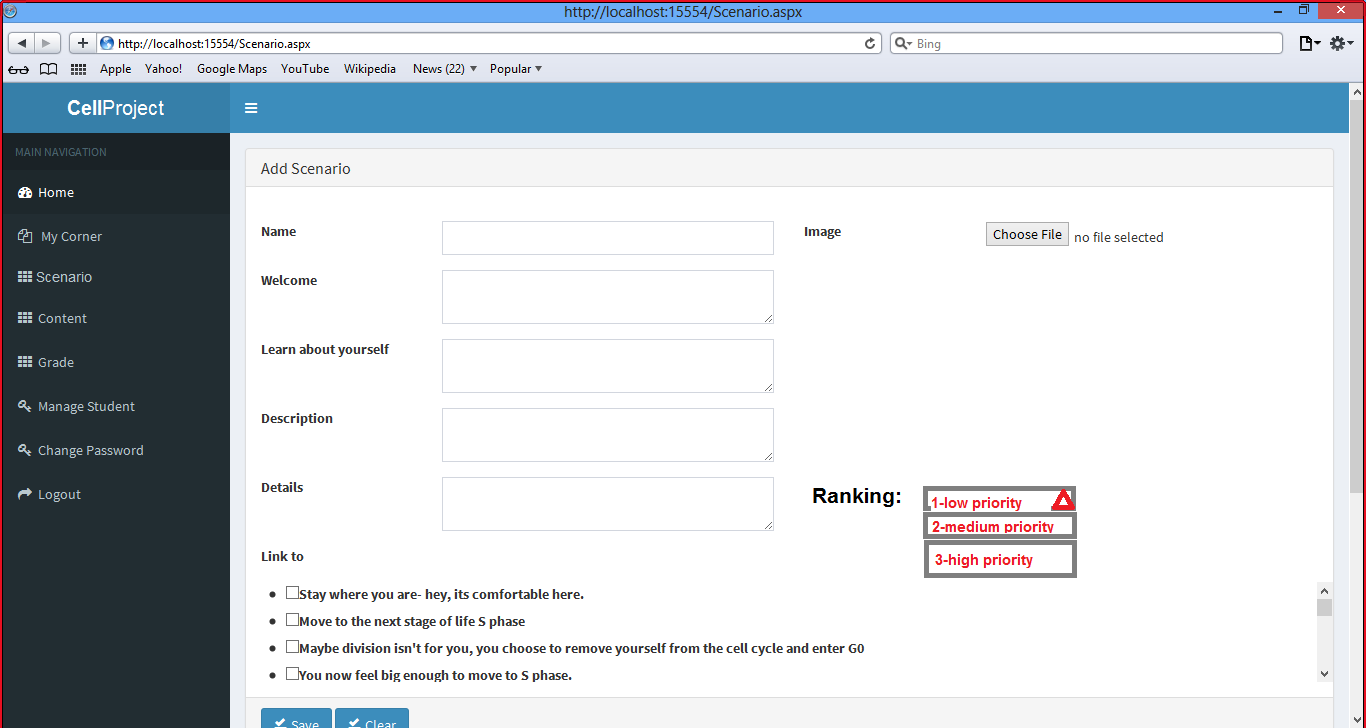
Ranking: 1-low priority

2: medium priority

3. High Priority

Please have a look at the below screenshot as a prototype and the use case diagram.

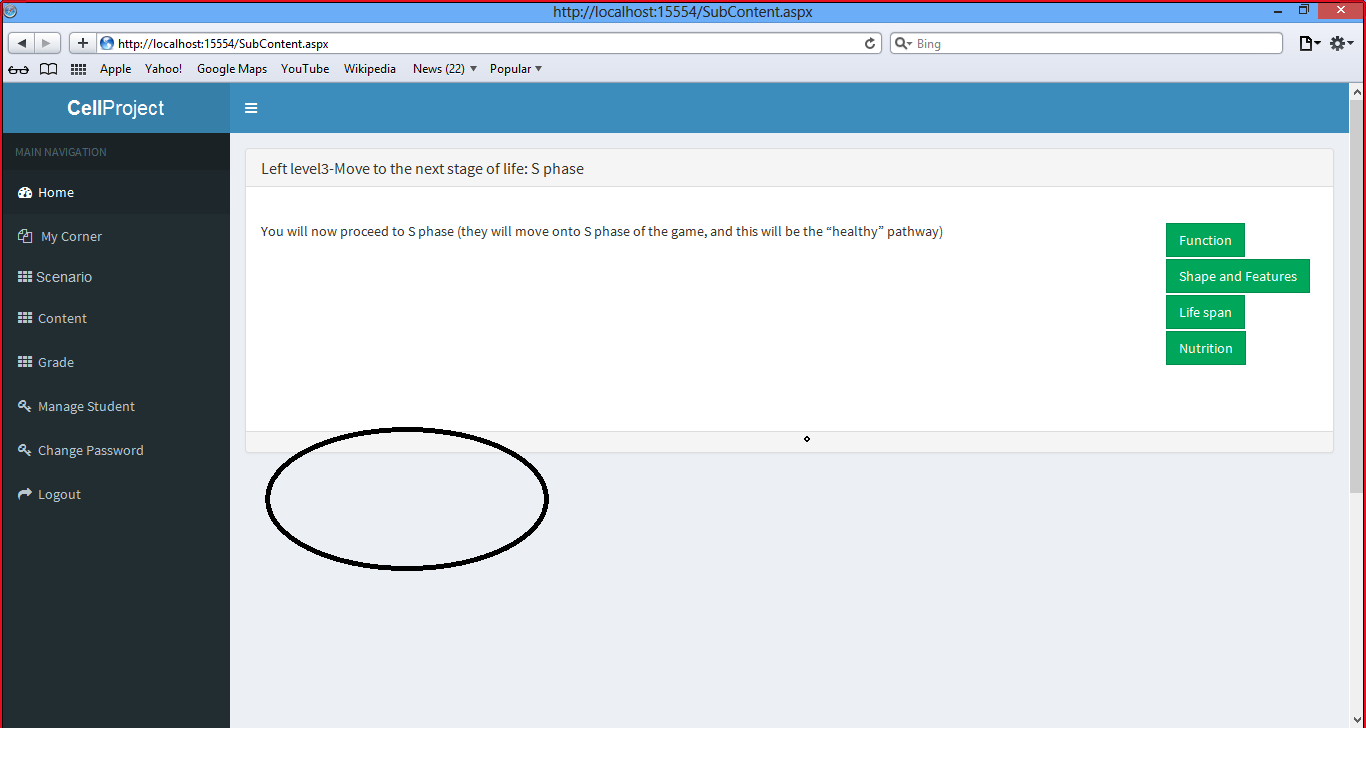
**Expected Screen:**

****

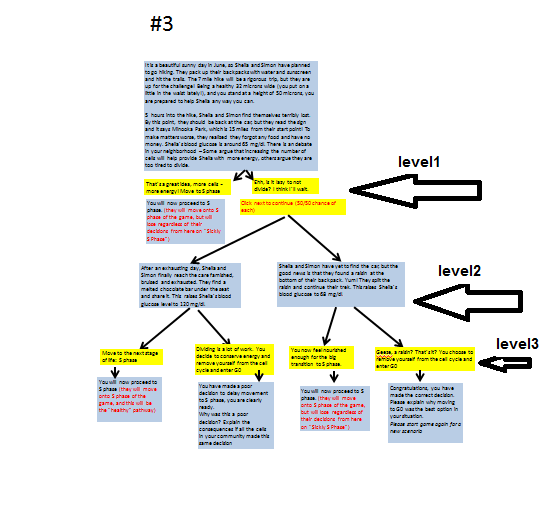
**Requirement R3:** Missing entire section of activity which is in the intermediate level of scenario3. There are existing 4 scenarios available in the existing application. In the Third scenario some portion of the activity is not loading or issue with loading to the gam which means a deadlock scenario while playing that scenario3.

**Flow chart:** this issue with third link propagation of contents. It will come under functional bugs or Functionality issue.

**Existing system Bug:**

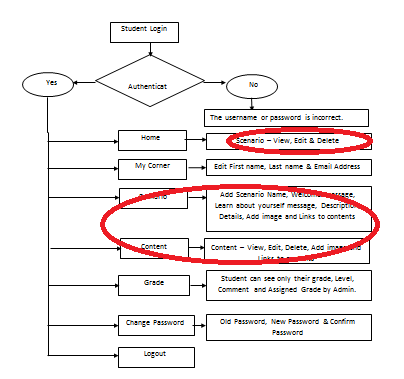


**Given Scenario:**

****

**R4**:- **Randomize available scenarios:** When a student enters into the application to, in existing application student have to choose the available scenarios to play the game. Instead a randomly picked scenario would be shown to a student to play the game and finish it.

* To pick up a scenario from the bucket of available scenarios should be prioritized based on the ranking given to the scenario at the time of creating it. Giving ranking number to the scenario is listed under the administration requirement which is in category2 requirements.



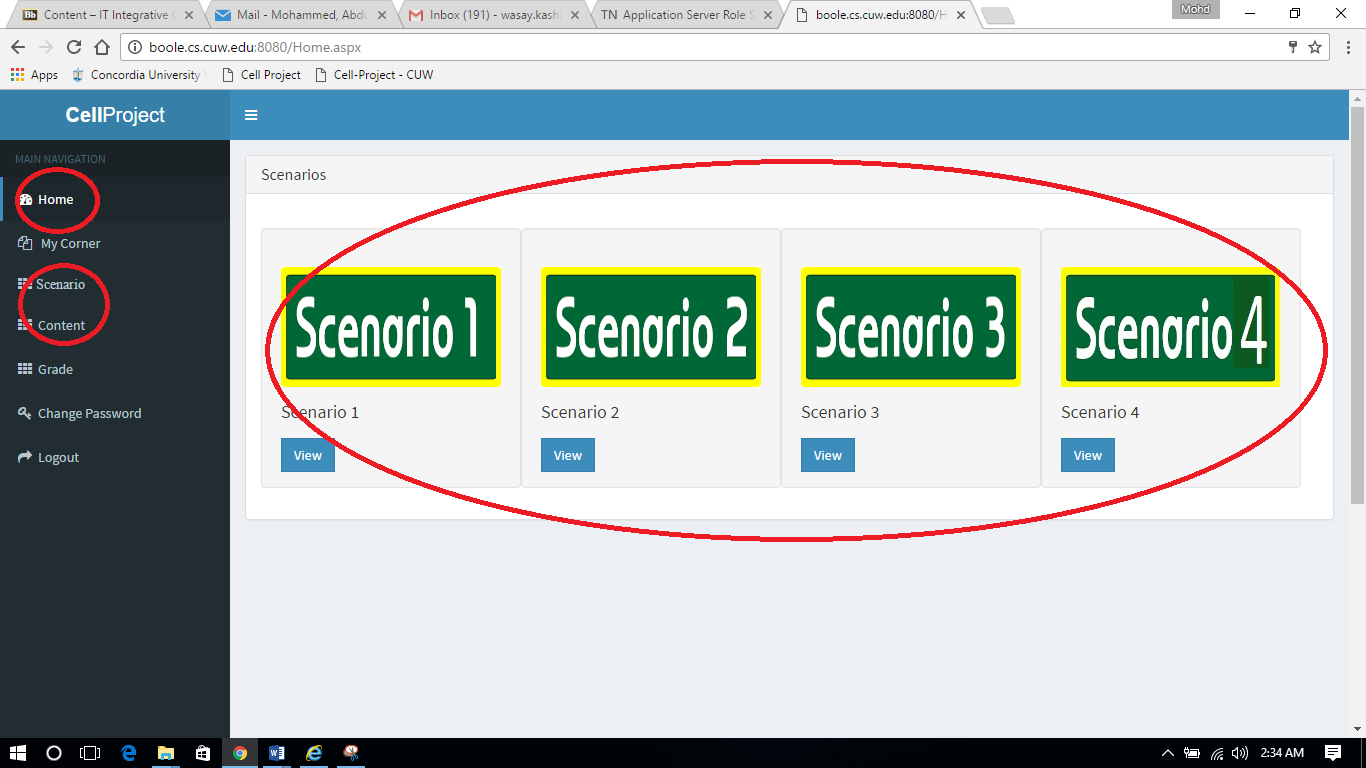
**Solution Strategy for this Requirement:-**

-Student will be omitted to pick up the scenario, edit the scenario or edit the content

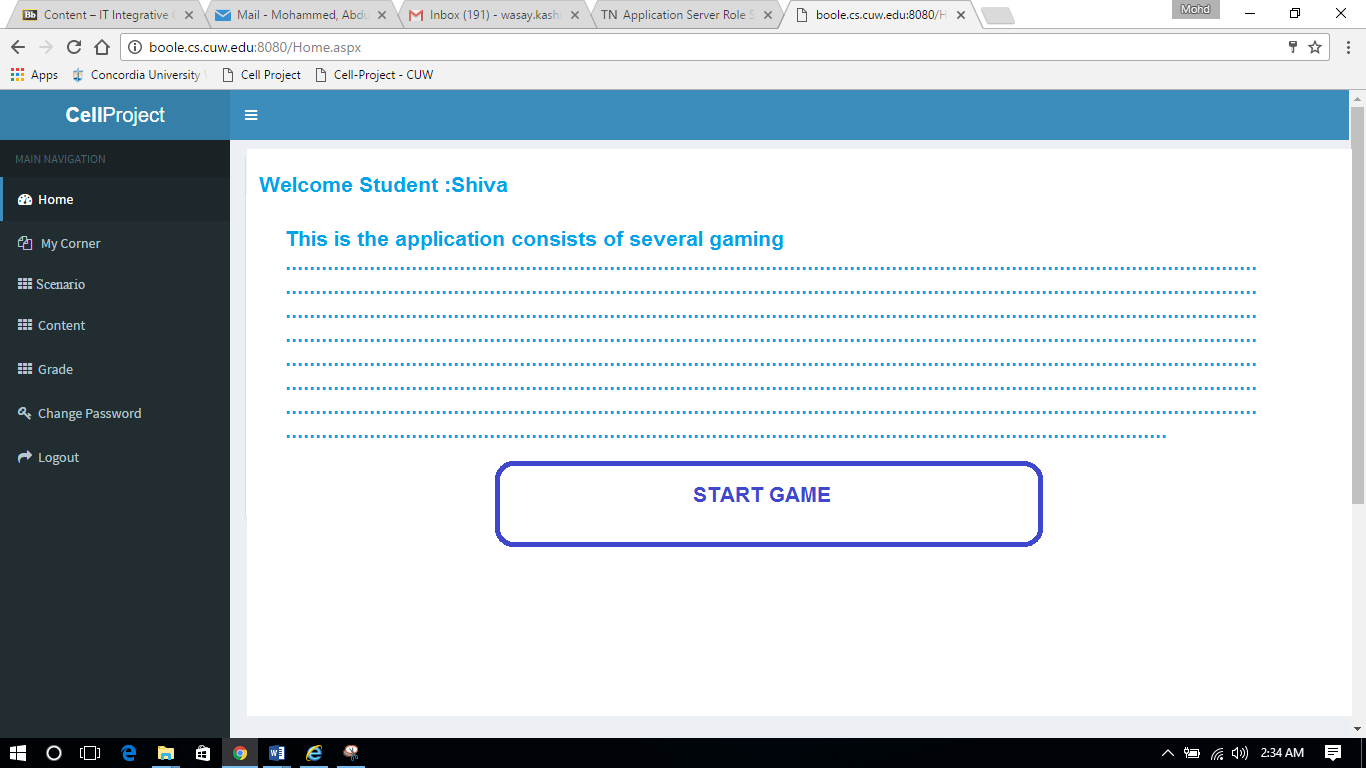
-In the home page of login success- It will contain the introduction of the application and the studies of the game and a button to states “Start Game” – On click of it, a randomly picked scenario will be started.

**Sample prototype and Use case diagram as follows:-**

Issues in Existing system as shown below:



Proposed idea: **PROTOTYPE**

****

By clicking “ START GAME” , student will be given a randomly picked scenario to play the game.

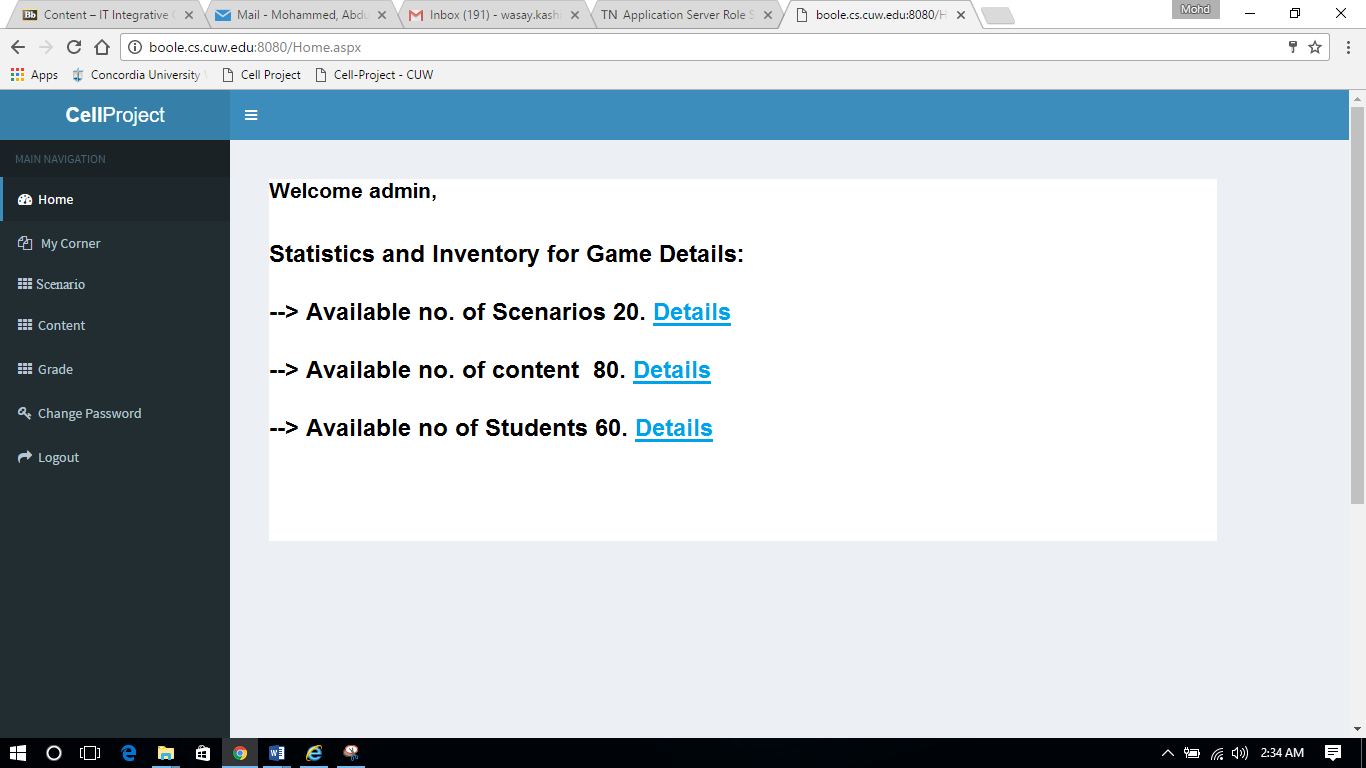
**Category 3 Requirements:-**

**R6**:- From the admin and student login perspectives: User Interface changes for the login pages, home page screens, scenario backgrounds in terms of color, proper alignments of textual information, comment boxes alignments.

**Following are the several issues and needs to be solved under the Application flow bugs and management of the application.**

1. Home screen of admin should be showing welcome message, Statistics or Inventory details.

**Solution Prototype: -**



On clicking Details after the item shown above in the figure respective page will be loaded. **For example:** -

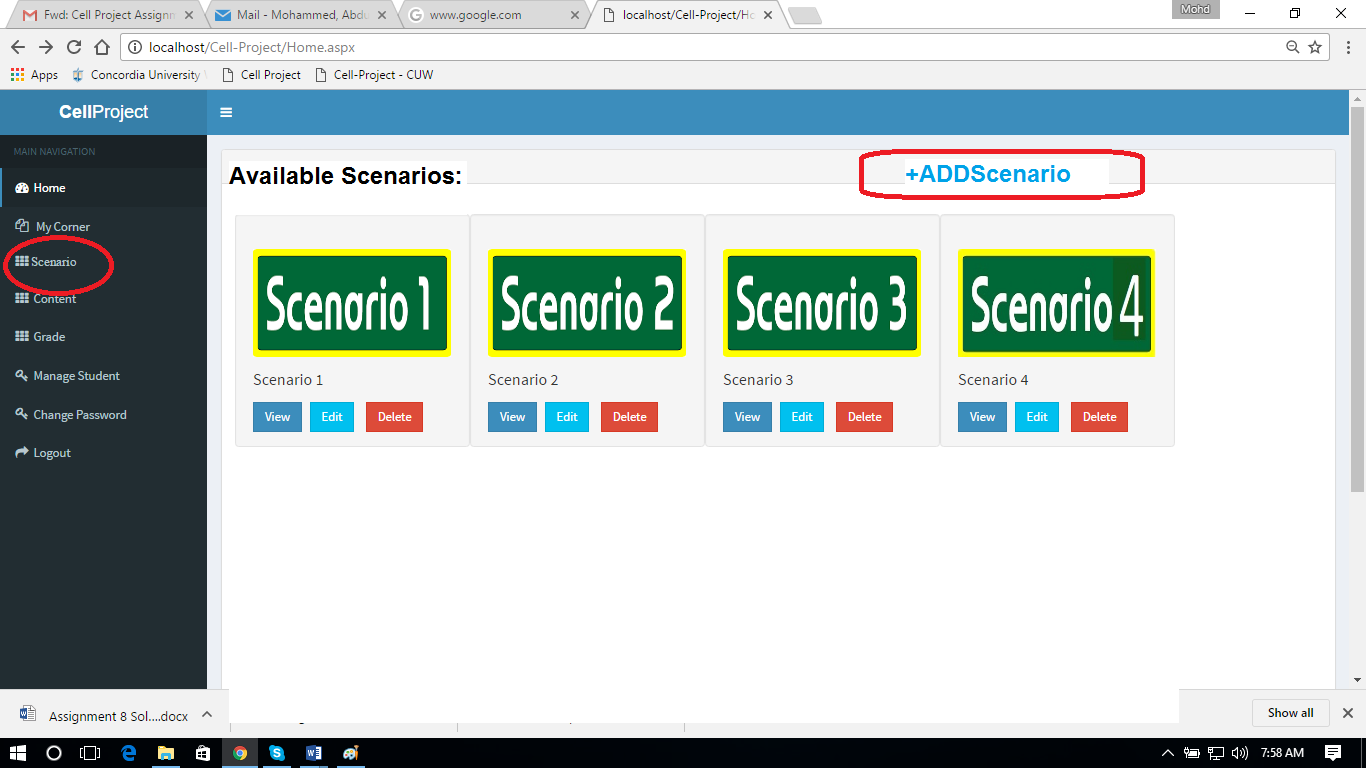
1. Details after no. of scenarios resulting to loading of scenario page by click on it.

2. Details after no. of content resulting to loading of scenario page by click on it.

3. Details after no. of students resulting to loading of students list into current page by click on it.

**2)** Scenario screen of admin should be showing welcome message, Available scenarios and +ADDScenario.

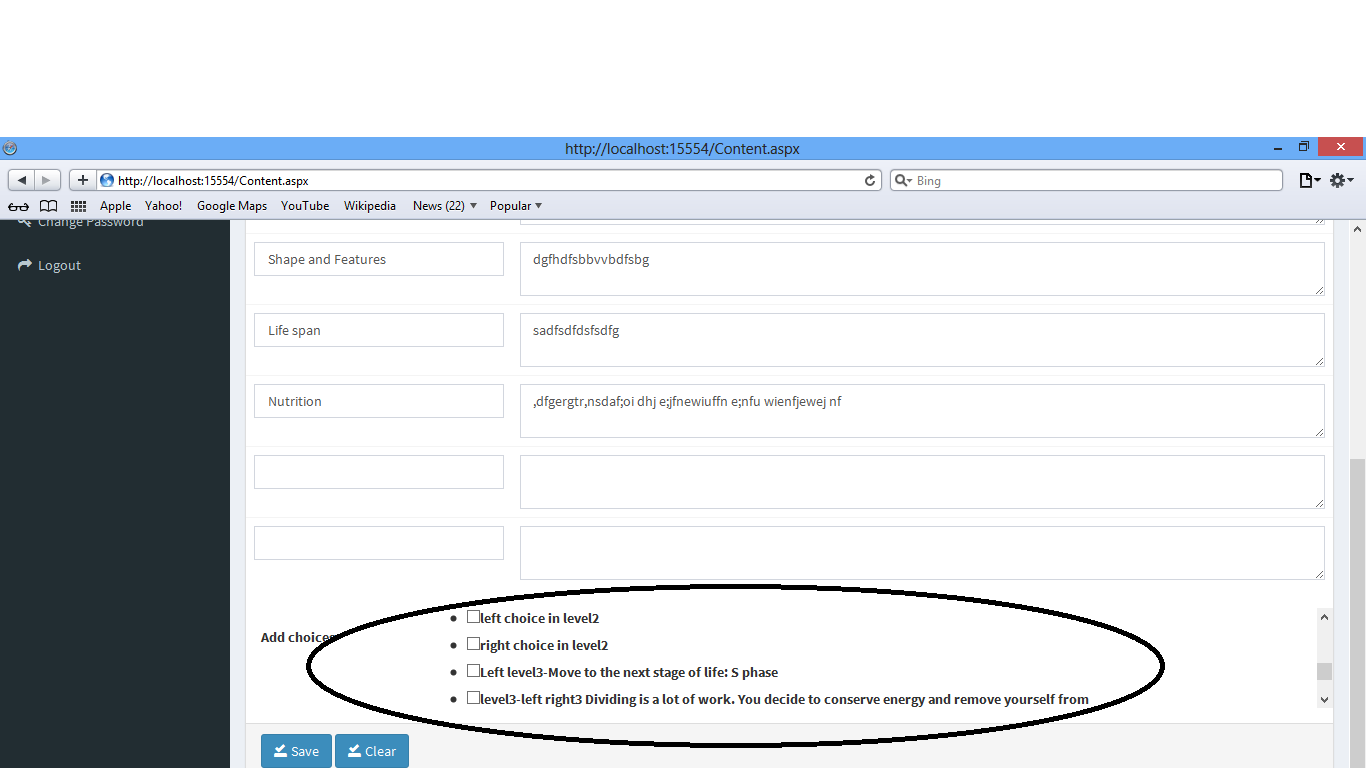
**Sample Prototype:**



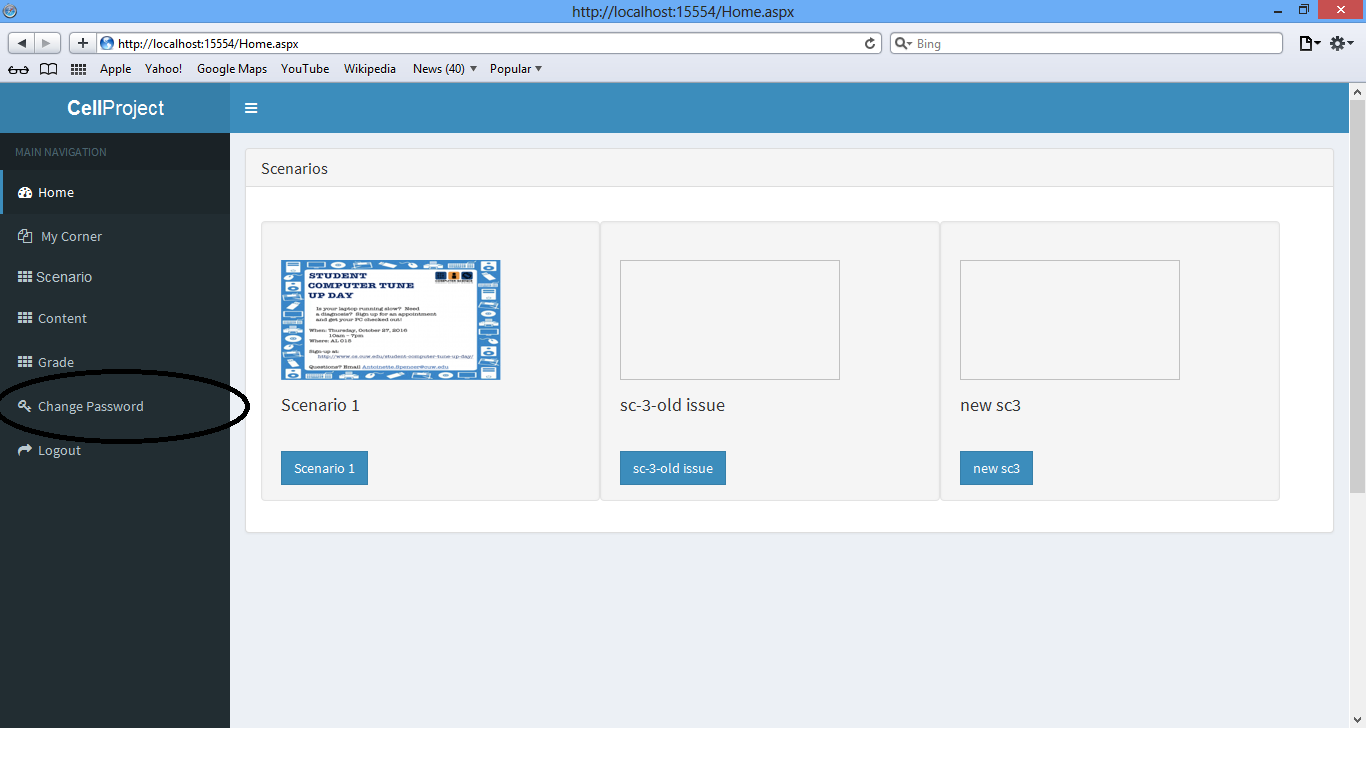
**3)** On admin login, while editing content – linked contents are not highlited resulting to loss the connectivity of the underlying contents on editing the current content.

**Solution:** Need to modify the content EDIT button call by populating the edit screen with already linked content headers under the link to section of the edit window.

**Existing issue sample screen:**



4) On student login change password should be displayed as account setting. **Label change issue.**



**Implementation**

This stage of process consists of the requirements, files changed and content of the respective code file changed to accomplish the current requirement.

**Category 1 Requirements:**

**For Admin activities:**

**\* R1:** New feature for Admin to be able to create dynamic scenarios with priority of ranking.

- **Ranking:** While application provides a random scenario to student, Instructors wants to give the priority to some of the scenarios which means a level of priority so that most of the students will get a chance to act to the scenario.

File modified was **scenario.aspx.cs**  with add scenario with ranking case. In the database end, two additional columns were added to satisfy the constraints of this requirement.

**Scenario.aspx.cs:**

protected void btnSave\_Click(object sender, EventArgs e)

{

using (SqlConnection con = new SqlConnection())

{

try

{

#region Declaration

int isStudent = Convert.ToInt32(Session["IsStudent"]);

string userID = Session["UserID"].ToString();

string image = null;

if (fileImage.HasFile)

{

image = DateTime.Now.ToString("yyyyMMddHHmmssfff").ToString() + ".jpg";

string filePath = "ScenarioImages/" + image;

fileImage.SaveAs(Server.MapPath(filePath));

}

int scenarioID = -1;

if (Session["ScenarioID"] != null)

{

scenarioID = Convert.ToInt32(Session["ScenarioID"]);

}

#endregion

#region Saving parent content

con.ConnectionString = ConfigurationManager.ConnectionStrings["CPDBCS"].ConnectionString;

SqlCommand cmd = new SqlCommand();

cmd.CommandText = "SaveScenario";

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Clear();

cmd.Parameters.Add("@ScenarioID", SqlDbType.Int).Value = scenarioID;

cmd.Parameters.Add("@Name", SqlDbType.VarChar).Value = txtName.Text;

cmd.Parameters.Add("@Welcome", SqlDbType.VarChar).Value = txtwelcome.Text;

cmd.Parameters.Add("@LearnAbt", SqlDbType.VarChar).Value = txtlearnabt.Text;

cmd.Parameters.Add("@Desc", SqlDbType.VarChar).Value = txtDesc.Text;

cmd.Parameters.Add("@Details", SqlDbType.VarChar).Value = txtDetails.Text;

if (lstRAnking.Text != null)

cmd.Parameters.Add("@Rank", SqlDbType.VarChar).Value = lstRAnking.Text;

else

cmd.Parameters.Add("@Rank", SqlDbType.VarChar).Value = "Low";

if (!string.IsNullOrEmpty(image))

{

cmd.Parameters.Add("@Image", SqlDbType.VarChar).Value = image;

}

cmd.Parameters.Add("@IsCreatedByStudent", SqlDbType.Int).Value = isStudent;

cmd.Parameters.Add("@CreatedBy", SqlDbType.Int).Value = userID;

cmd.Parameters.Add("@NewScenarioID", SqlDbType.VarChar, 50);

cmd.Parameters["@NewScenarioID"].Direction = ParameterDirection.Output;

cmd.Connection = con;

con.Open();

cmd.ExecuteNonQuery();

string newScenarioID = cmd.Parameters["@NewScenarioID"].Value.ToString();

#endregion

#region Saving child content

foreach (ListItem childContentID in chkContents.Items)

{

int contentID = Convert.ToInt32(childContentID.Value);

int isDel = 0;

if (!childContentID.Selected)

{

isDel = 1;

}

cmd.CommandText = "SaveScenarioContent";

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Clear();

cmd.Parameters.Add("@ScenarioID", SqlDbType.Int).Value = newScenarioID;

cmd.Parameters.Add("@ContentID", SqlDbType.Int).Value = contentID;

cmd.Parameters.Add("@IsDeleted", SqlDbType.Int).Value = isDel;

cmd.Connection = con;

cmd.ExecuteNonQuery();

}

#endregion

ScriptManager.RegisterClientScriptBlock(this, GetType(), "alert", "alert('Scenario saved successfully')", true);

Response.Redirect("ShowScenarios.aspx",false);

ApplicationInstance.CompleteRequest();

}

catch (Exception ex)

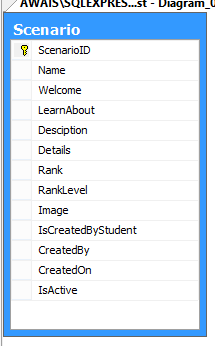
{

throw new Exception(ex.Message);

}

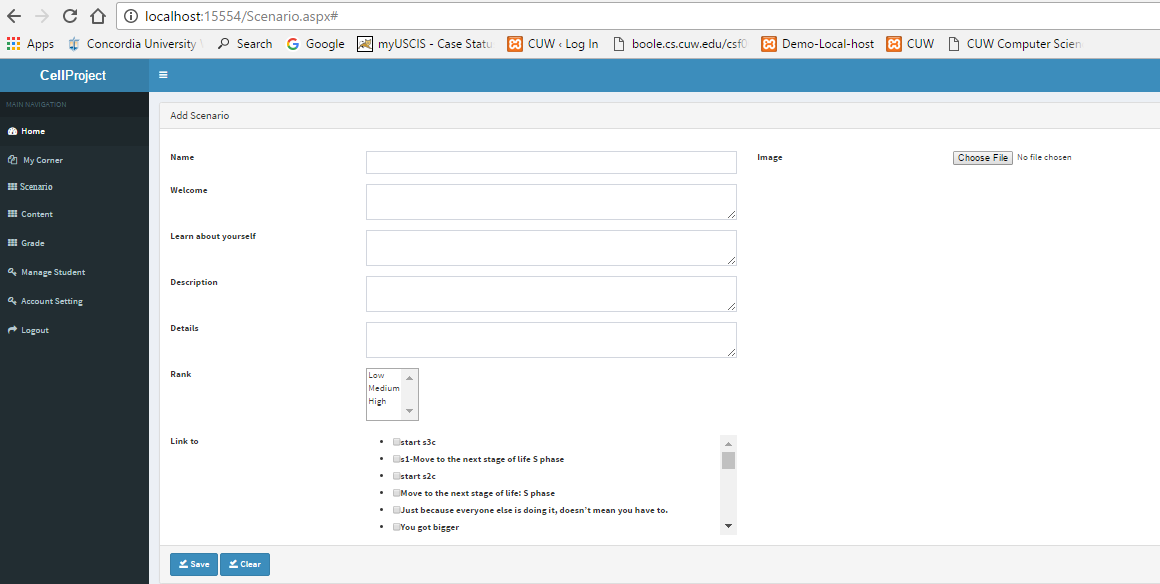
};

Database new Scenario table with columns list:



Rank and Ranklevel columns are used for this requirement and also used for randomly pulling a scenario from the student perspective.

**Screenshot of output:-**



**R3:** Missing entire section of activity which is in the intermediate level of scenario3. There are existing 4 scenarios available in the existing application. In the Third scenario some portion of the activity is not loading or issue with loading to the gam which means a deadlock scenario while playing that scenario3.

Following is the modified code of file affected in satisfying the constraints of the above requirement which is missing content edit functionality Bug while adding links/add Choices to it.

showContent.aspx.cs

using System;

using System.Web.UI;

using System.Data;

using System.Data.SqlClient;

using System.Configuration;

using System.Web.UI.WebControls;

public partial class ShowContents : Page

{

protected void Page\_Load(object sender, EventArgs e)

{

#region Validating session

if (Session["UserID"] == null || Session["IsStudent"] == null)

{

Response.Redirect("index.html", false);

ApplicationInstance.CompleteRequest();

}

#endregion

Session["HideComment"] = null;

if (!IsPostBack)

{

getContents();

}

}

public void getContents()

{

#region Bind contents to data list

using (SqlConnection con = new SqlConnection())

{

con.ConnectionString = ConfigurationManager.ConnectionStrings["CPDBCS"].ConnectionString;

SqlCommand cmd = new SqlCommand();

cmd.CommandText = "ShowContent";

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Clear();

cmd.Parameters.Add("@UserID", SqlDbType.Int).Value = Session["UserID"].ToString();

cmd.Connection = con;

con.Open();

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataSet ds = new DataSet();

da.Fill(ds, "ShowContent");

if (ds.Tables["ShowContent"].Rows.Count > 0)

{

dlContents.DataSource = ds.Tables["ShowContent"];

dlContents.DataBind();

}

//SqlDataReader dr = cmd.ExecuteReader();

//if (dr.Read())

//{

// dlContents.DataSource = dr;

// dlContents.DataBind();

//}

};

#endregion

}

protected void btnAddContent\_Click(object sender, EventArgs e)

{

Session["ContentID"] = null;

Response.Redirect("Content.aspx", false);

ApplicationInstance.CompleteRequest();

}

protected void dlContents\_ItemCommand(object source, System.Web.UI.WebControls.DataListCommandEventArgs e)

{

if (e.CommandName == "cmdEdit")

{

Session["ContentID"] = Convert.ToInt32(e.CommandArgument);

Response.Redirect("Content.aspx");

}

else if (e.CommandName == "cmdDelete")

{

Session["ContentID"] = Convert.ToInt32(e.CommandArgument);

using (SqlConnection con = new SqlConnection())

{

con.ConnectionString = ConfigurationManager.ConnectionStrings["CPDBCS"].ConnectionString;

SqlCommand cmd = new SqlCommand();

cmd.CommandText = "DeleteContent";

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Clear();

cmd.Parameters.Add("@ContentId", SqlDbType.Int).Value = Convert.ToInt32(Session["ContentID"]);

cmd.Connection = con;

con.Open();

cmd.ExecuteNonQuery();

ScriptManager.RegisterStartupScript(this, this.GetType(), "alert", "alert('Content Deleted Sucessfully..')", true);

getContents();

}

}

else if(e.CommandName == "cmdView")

{

Session["ContentID"] = Convert.ToInt32(e.CommandArgument);

Session["HideComment"] = true;

Response.Redirect("SubContent.aspx", false);

ApplicationInstance.CompleteRequest();

}

}

protected void dlContents\_ItemDataBound(object sender, System.Web.UI.WebControls.DataListItemEventArgs e)

{

if (Convert.ToInt32(Session["IsStudent"]) != 0 && Convert.ToBoolean(Session["IsAllowEdit"]) == false)

{

LinkButton lnkEdit = (LinkButton)e.Item.FindControl("lnkEdit");

lnkEdit.Visible = false;

LinkButton lnkDelete = (LinkButton)e.Item.FindControl("lnkDelete");

lnkDelete.Visible = false;

}

}

}

🡪

And also changed stored Procedure in the database:-

**Dbo.Savecubcontent**

USE [CellTest]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER PROC [dbo].[SaveSubContent]

@ParentID INT,

@ChildID INT,

@IsDeleted INT

AS

BEGIN

SET NOCOUNT ON;

--IF(@IsDeleted = 1)

--BEGIN

-- DELETE FROM

-- SubContent

-- WHERE

-- ParentID = @ParentID

-- AND

-- ChildID = @ChildID

--END

--ELSE

--BEGIN

-- IF (@ParentID != @ChildID)

-- BEGIN

INSERT INTO

SubContent(ParentID, ChildID, IsActive)

VALUES

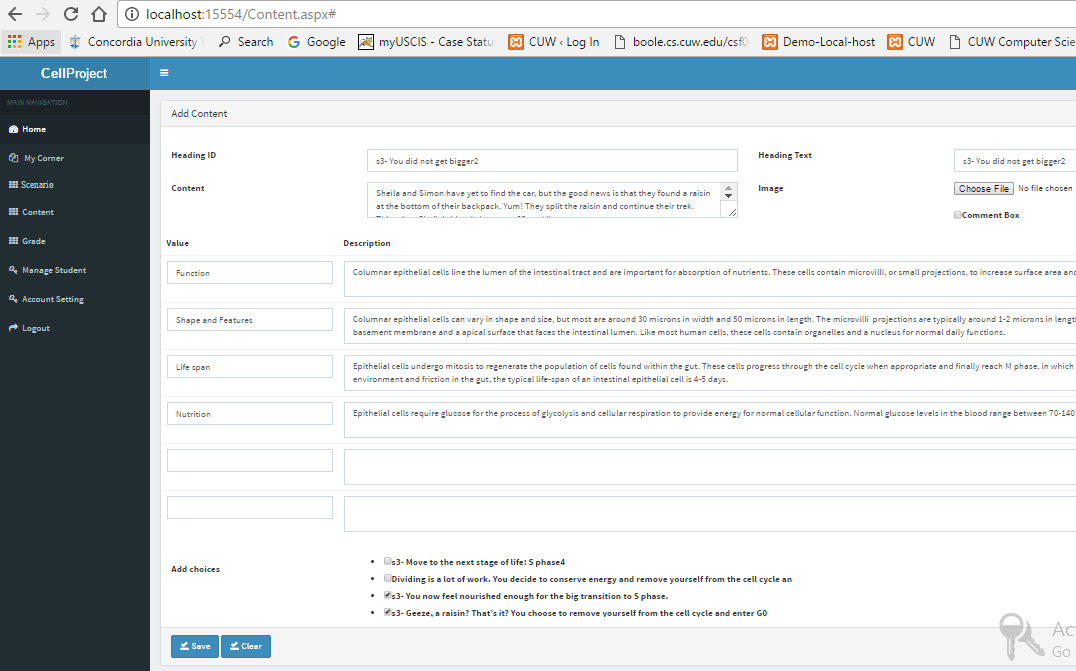
(@ParentID, @ChildID, 1)

-- END

--END

END

**Screenshot of the sample output of enabled links in edit screen of the content**:-



**Category 2 Requirements:**

**\*\*\*R4**:- randomize available scenarios: When a student enters into the application to, in existing application student have to choose the available scenarios to play the game. Instead a randomly picked scenario would be shown to a student to play the game and finish it.

* To pick up a scenario from the bucket of available scenarios should be prioritized based on the ranking given to the scenario at the time of creating it. Giving ranking number to the scenario is listed under the administration requirement which is in category2 requirements.

Algorithm used in randomly puling scenario from high priority scenarios.

**Algorithmic Steps:**

1) When a user clicks “StartGame” button in the – A scenario will be picked from the logical container consists of highly prioritized scenarios based on the below logical steps.

1.1: all the highly prioritized scenarios are given with a head value of 0 at the time of creation.

1.2: once the certain scenario is picked up, its head value will be incremented by 1 and removed from the logical container.

1.3 Another student clicks on Start Game will results picking up of scenario from the bucket of least head value list.

1.4 After every scenario has picked up from the logical container and when all the head values are equal. Those scenarios are again dumped into logical container.

1.5 step 1.2 to 1.4 repeats into infinite loop.

1.6 ends.

**ShowScenarios.aspx :** this is the file modified while adding functionality of randomly picking one scenario.

protected void Page\_Load(object sender, EventArgs e)

{

#region Validating session

if (Session["UserID"] == null || Session["IsStudent"] == null)

{

Response.Redirect("index.html", false);

ApplicationInstance.CompleteRequest();

}

if (Convert.ToInt32(Session["IsStudent"]) != 0 && Convert.ToBoolean(Session["IsAllowEdit"]) == false)

{

btnAddScenario.Visible = false;

}

#endregion

Session["ScenarioID"] = null;

Session["HideComment"] = null;

if (!IsPostBack)

{

if (Convert.ToInt32(Session["IsStudent"]) != 1)

getScenario();

else

{

using (CellEntities context = new CellEntities())

{

var objscanrio = (from rs in context.Scenarios where rs.Rank== "High" select rs).ToList().OrderBy(cond => cond.RankLevel).FirstOrDefault();

if (objscanrio != null)

{

List<ScenarioView> scenario = new List<ScenarioView>();

scenario.Add(new ScenarioView() { Name = objscanrio.Name, ScenarioID = objscanrio.ScenarioID, ScenarioImages = objscanrio.Image });

dlScenario.DataSource = scenario;

dlScenario.DataBind();

// getscanrio.RankLevel = getscanrio.RankLevel + 1;

// context.SaveChanges();

}

else

{

dlScenario.DataSource = null;

}

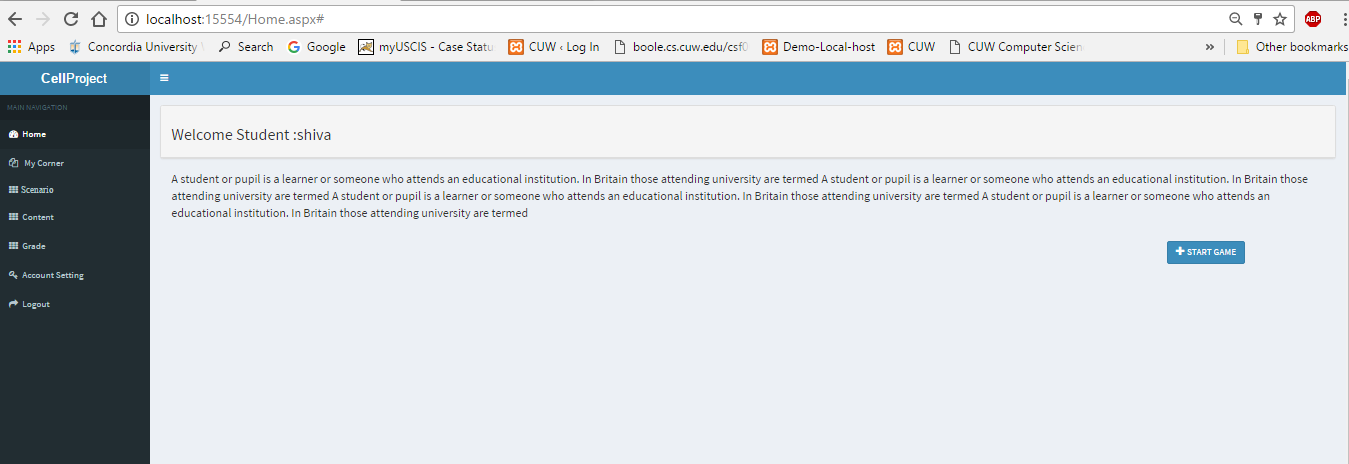
}

}

}

}

**Sample Output screen which is student home screen: where student clicks on “Start Game” to pick a random scenario**



**R5:-** Giving feature of adding comments at the time of taking an action in the steps of a scenario. For example, providing comment text box for selecting parameter in selection of available parameters (options) in the intermediate steps of a scenario. –confirmed with clients. Need to give comments for every step but not before the finish.

Files affected and stored procedures used to accomplish this requirement and their purpose as follows:-

SubContent.aspx

protected void btnSave\_Click(object sender, EventArgs e)

{

int scenarioid = Convert.ToInt32(Session["ScenarioID"]);

if (txtComment.Text == "")

{

ScriptManager.RegisterStartupScript(this, this.GetType(), "alert", "alert('Please Post Your Comment')", true);

return;

}

#region Save comment

using (SqlConnection con = new SqlConnection())

{

con.ConnectionString = ConfigurationManager.ConnectionStrings["CPDBCS"].ConnectionString;

SqlCommand cmd = new SqlCommand();

cmd.CommandText = "SaveComment";

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Clear();

cmd.Parameters.Add("@ScenarioID", SqlDbType.Int).Value = scenarioid;

if (Session["ContentID"] != null)

{

cmd.Parameters.Add("@ContentID", SqlDbType.Int).Value = Session["ContentID"].ToString();

}

cmd.Parameters.Add("@StudentID", SqlDbType.Int).Value = Session["UserID"].ToString();

cmd.Parameters.Add("@Comment", SqlDbType.VarChar).Value = txtComment.Text;

cmd.Connection = con;

con.Open();

cmd.ExecuteNonQuery();

btnSave.Enabled = false;

if (btnSave.Text == "Finish")

{

using (CellEntities context = new CellEntities())

{

var getscanrio = (from rs in context.Scenarios where rs.ScenarioID == scenarioid select rs).FirstOrDefault();

if(getscanrio!=null)

{

getscanrio.RankLevel = getscanrio.RankLevel + 1;

context.SaveChanges();

}

}

ScriptManager.RegisterStartupScript(this, GetType(), "alert", "alert('Game finish successfully');window.location ='Home.aspx';", true);

}

};

#endregion

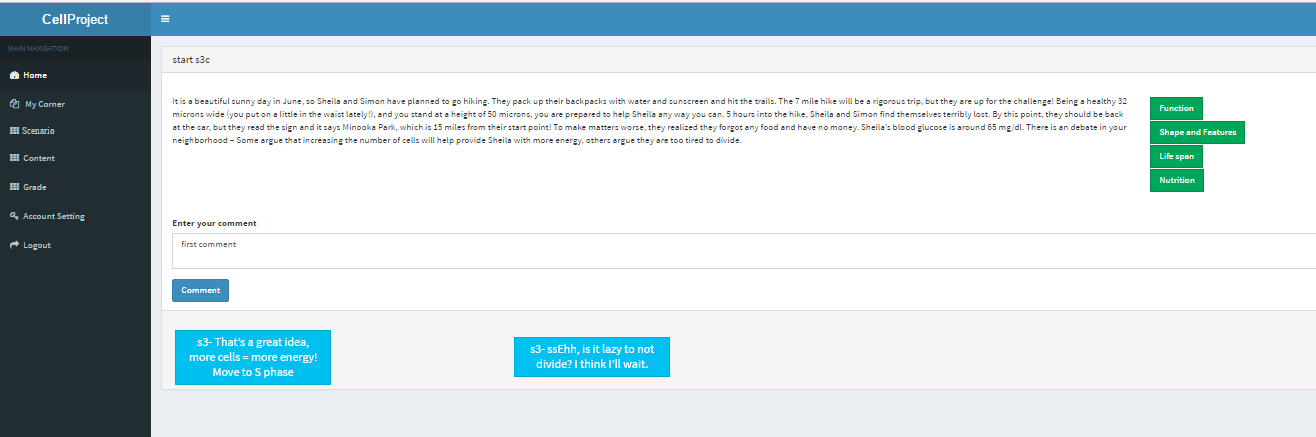
btnSave.Visible = false;

txtComment.Text = "comment Save successfully";

txtComment.Enabled = false;

}

**Sample screen showing comment box in the middle of the game:**



Above requirement lead to another feature development which is “Activity LOG”.

**R7:- Activity LOG Screen**: when a student plays a game with providing comments at each stage of game then from the admin perspective, while grading a screen after clicking “grade” button in the grade screen to grade the student, A screen appears consists of the student name, scenario and the activity of all the captured steps and the given comments by students.

**Use of this Requirement:** Activity log screen consists of Detailed analysis of the scenario played by student to be graded by Instructor.

To load activity log followed stored procedure gets run onclicking “grade” button in respective game item list in grade screen.

Stored Procedure Name: Dbo.get\_student\_for\_grade

USE [CellTest]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER PROC [dbo].[GET\_STUDENT\_FOR\_GRADE]

@studentId INT,

@scenarioID INT

AS

BEGIN

SELECT

CMT.Comments AS COMMENT,

CMT.ScenarioID AS SCID,

(ST.firstName+ST.lastName) AS STUDENTNAME,

CMT.[Grade] AS GRADE,

SC.Name AS LEVEL,

CMT.CommentID AS COMMENTID,

SC.ScenarioID as scenarioID,

cnt.HeadingID as contentId

FROM

Comment AS CMT

INNER JOIN

Scenario AS SC

ON

CMT.ScenarioID = SC.ScenarioID

INNER JOIN

Student\_Table AS ST

ON

ST.[studentId] = CMT.StudentID

inner join Content as cnt

on cmt.ContentID=cnt.ContentID

where cmt.StudentID=@studentId

and cmt.ScenarioID=@scenarioID

ORDER BY SC.ScenarioID

END

**Stored Procedure : to show students grade: -**

**dbo.show\_grade\_to\_student**

USE [CellTest]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER PROC [dbo].[SHOW\_GRADE\_TO\_STUDENT]

@studentId INT

AS

BEGIN

SELECT

SC.Name AS LEVEL,

stgrd.Grade AS GRADE,

SC.ScenarioID as scenarioID

FROM StudentGrade AS stgrd

INNER JOIN

Scenario AS SC

ON stgrd.ScenarioID = SC.ScenarioID

WHERE stgrd.StudentID = @studentId

ORDER BY SC.ScenarioID

END

Code file changed for the grading Activity LOG requirement:

Name: StudentGrade.aspx

using System;

using System.Configuration;

using System.Data.SqlClient;

using System.Data;

using System.Collections.Generic;

using System.Linq;

public partial class studentGrade : System.Web.UI.Page

{

SqlConnection con = new SqlConnection(ConfigurationManager.ConnectionStrings["CPDBCS"].ConnectionString);

SqlCommand cmd = new SqlCommand();

DataSet dsGrade = new DataSet();

SqlDataAdapter adp;

protected void Page\_Load(object sender, EventArgs e)

{

if (!Page.IsPostBack)

{

try

{

//int studentid = Convert.ToInt32(Session["StudentID"]);

//int ScenarioID = Convert.ToInt32(Session["ScenarioID"]);

//using (CellEntities context = new CellEntities())

//{

// var objscanrio = (from rs in context.StudentGrades where rs.StudentID == studentid && rs.ScenarioID == ScenarioID select rs).FirstOrDefault();

// if (objscanrio != null)

// {

// txtGrade.Text = objscanrio.Grade;

// }

//}

if (con.State == ConnectionState.Closed)

{

con.Open();

}

cmd.Connection = con;

cmd.Parameters.Clear();

cmd.Parameters.Add("@studentId", SqlDbType.Int).Value = Convert.ToInt32(Session["UserID"]);

cmd.CommandType = CommandType.StoredProcedure;

cmd.CommandText = "SHOW\_GRADE\_TO\_STUDENT";

adp = new SqlDataAdapter(cmd);

adp.Fill(dsGrade, "dsGrade");

gridGrade.DataSource = dsGrade;

gridGrade.DataBind();

}

catch (Exception ex)

{

throw ex;

}

finally

{

if (con.State == ConnectionState.Open)

{

con.Close();

}

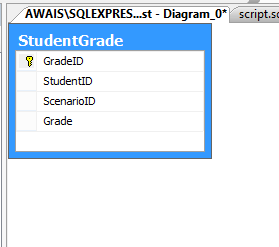
}

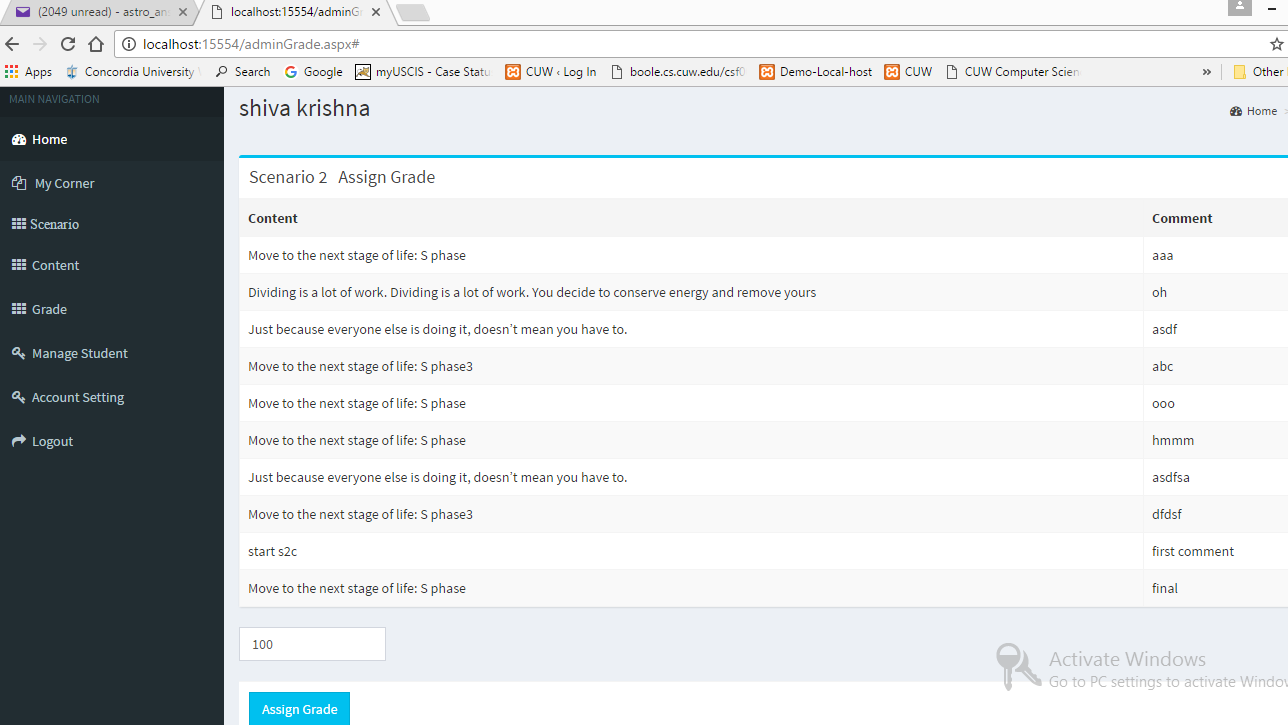
}

}

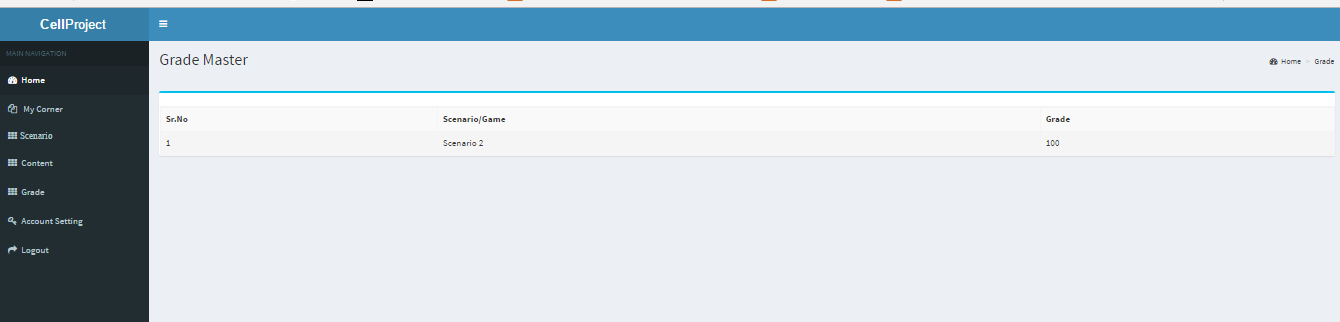
}

Grade Table in database:



Sample Grade screen: 

**From the student view Grade screen for above graded item will look like below:-**



**Testing and Issues**

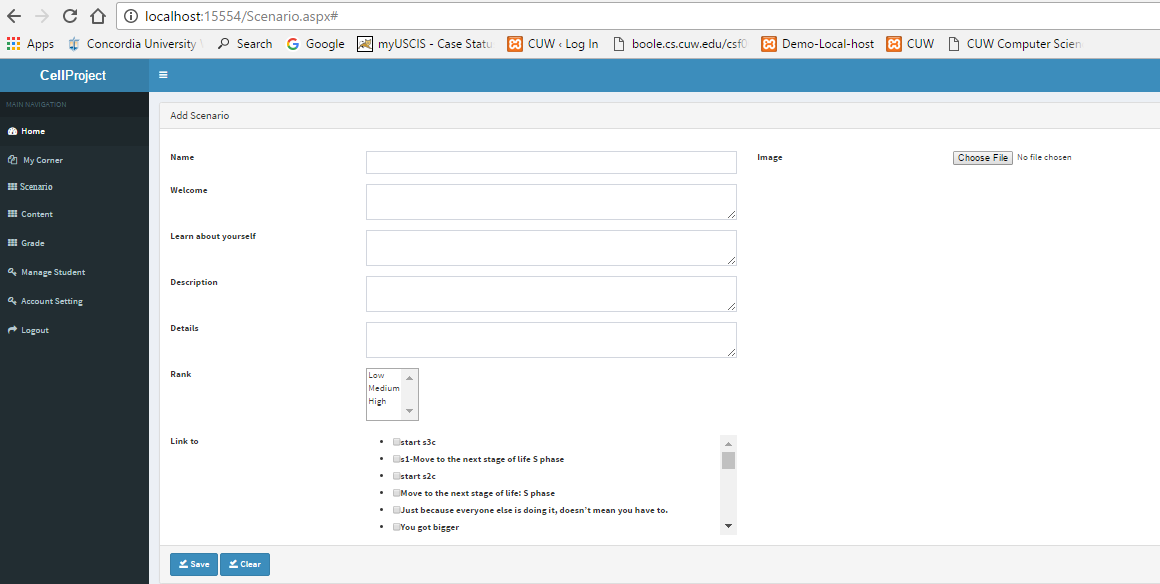
**R1:** New feature for Admin to be able to create dynamic scenarios with priority of ranking.

**R4**: Randomize available scenarios: When a student enters into the application to, in existing application student have to choose the available scenarios to play the game. Instead a randomly picked scenario would be shown to a student to play the game and finish it.

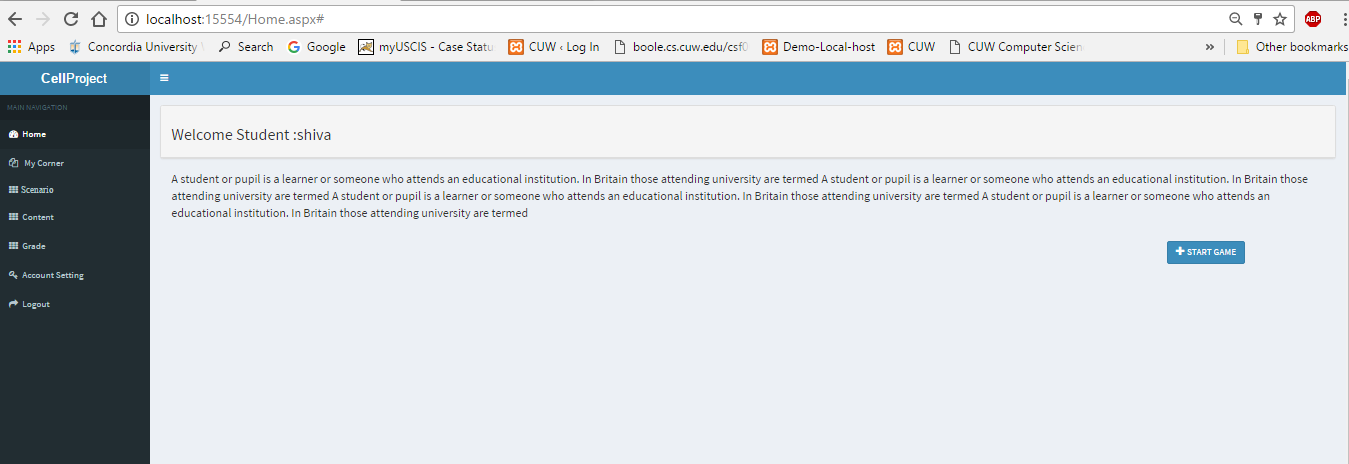
For R1 and R4 requirement, we did follow manual approach by creating several test cases of changing priority levels as shown in below screen and checking the data in scenario table in database.

Followed screen shots are the outputs of the above R1 and R4 requirements accomplishments: -

-- scenario screen shot



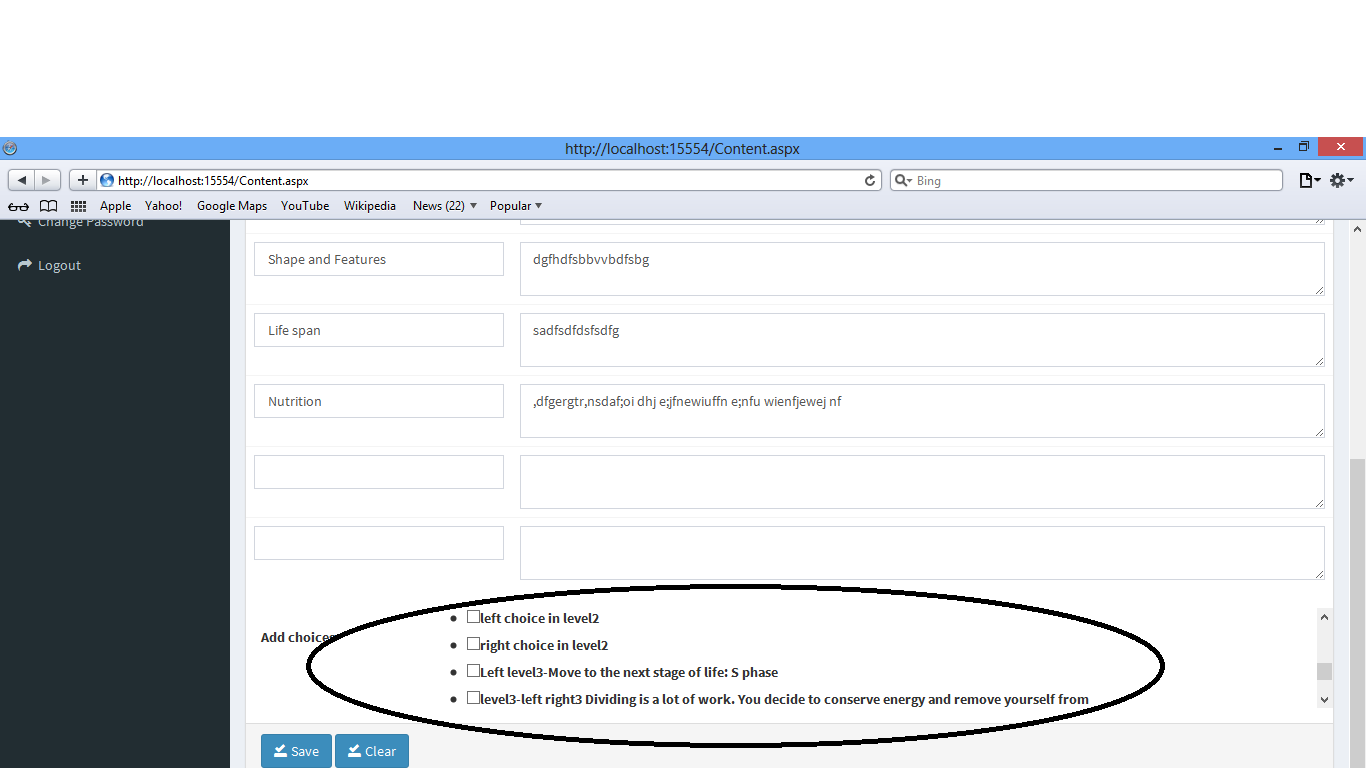
-- Start Game screenshot



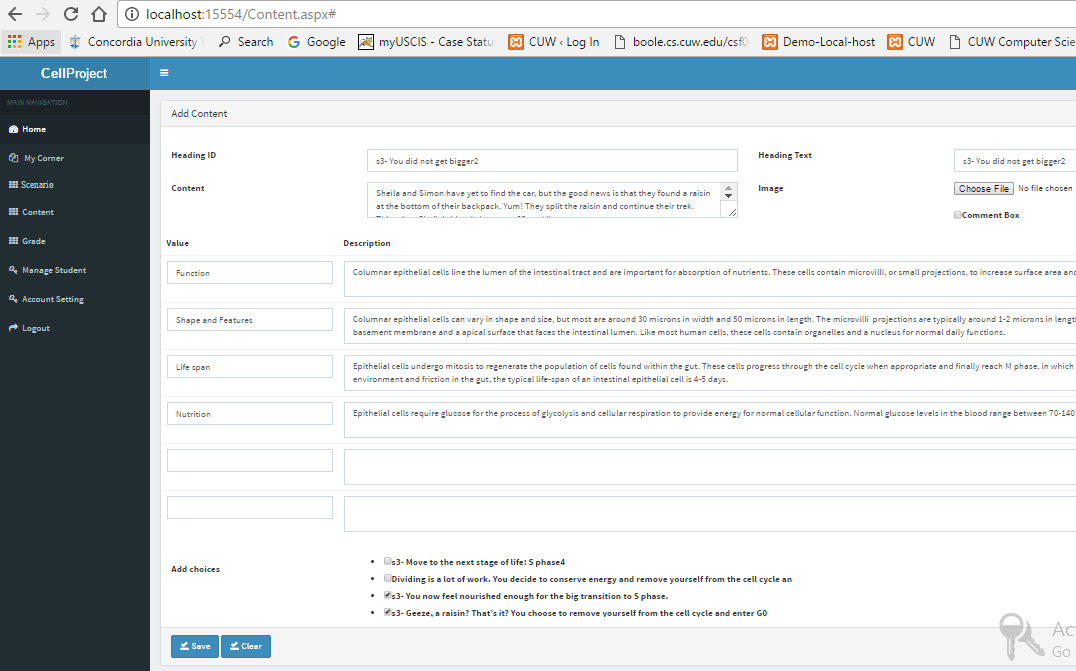
**R3:** Missing entire section of activity which is in the intermediate level of scenario3. There are existing 4 scenarios available in the existing application. In the Third scenario, some portion of the activity is not loading or issue with loading to the gam which means a deadlock scenario while playing that scenario3.

For this requirement, we followed manual testing as it requires action of editing the content window and checking the links status in SubContent, scenario and scenarioContent table.

**Sample bug existing screen for above requirement:**

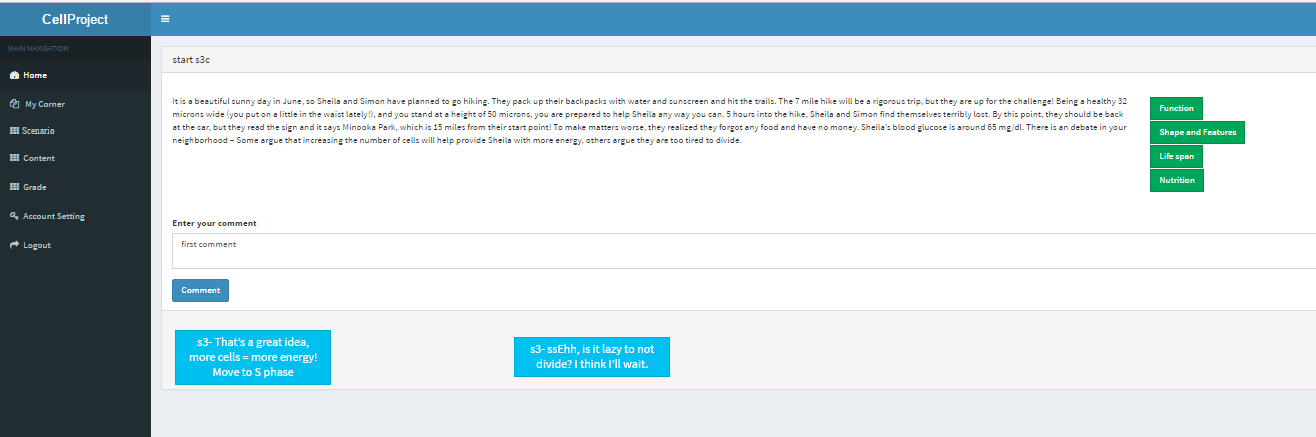


**Sample output screen:**



**R5:-** Giving feature of adding comments at the time of taking an action in the steps of a scenario. For example, providing comment text box for selecting parameter in selection of available parameters (options) in the intermediate steps of a scenario. –confirmed with clients. Need to give comments for every step but not before the finish.

Comment successfully added to the table “Comment” in the database with scenario, content and student Ids.



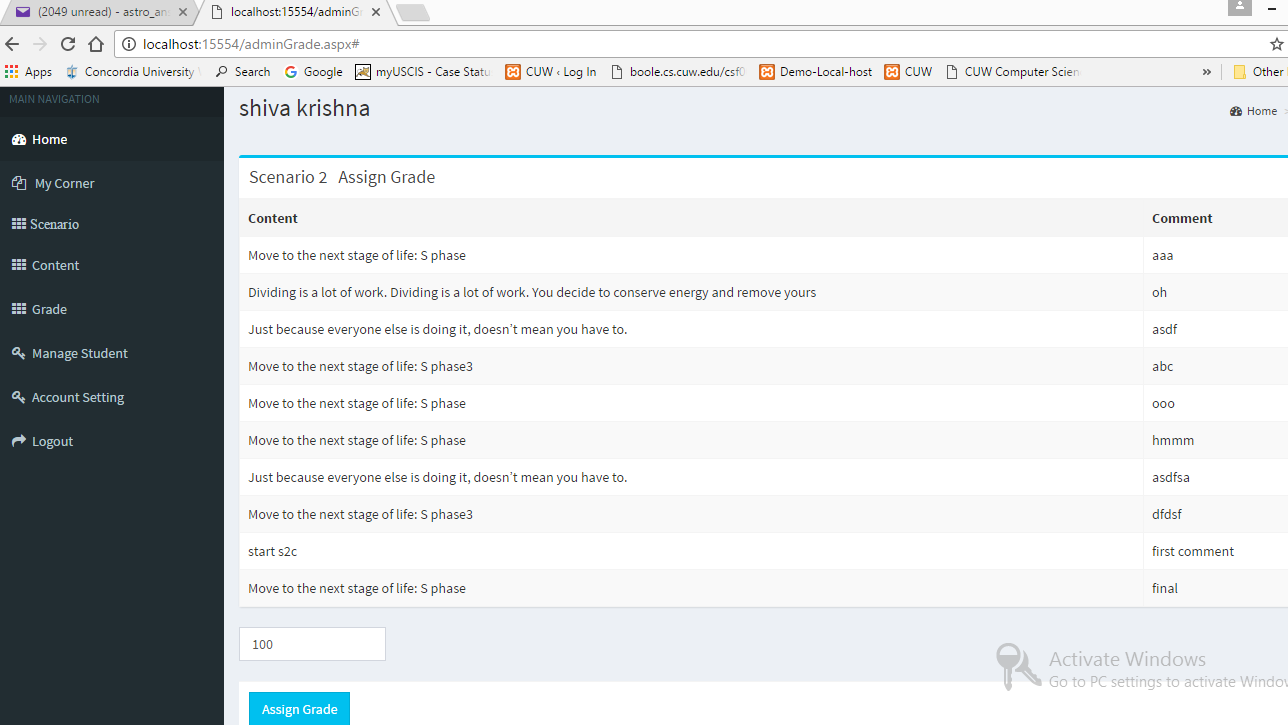
Above requirement lead to another feature development which is “Activity LOG”.

**R7:- Activity LOG Screen**: when a student plays a game with providing comments at each stage of game then from the admin perspective, while grading a screen after clicking “grade” button in the grade screen to grade the student, A screen appears consists of the student name, scenario and the activity of all the captured steps and the given comments by students.

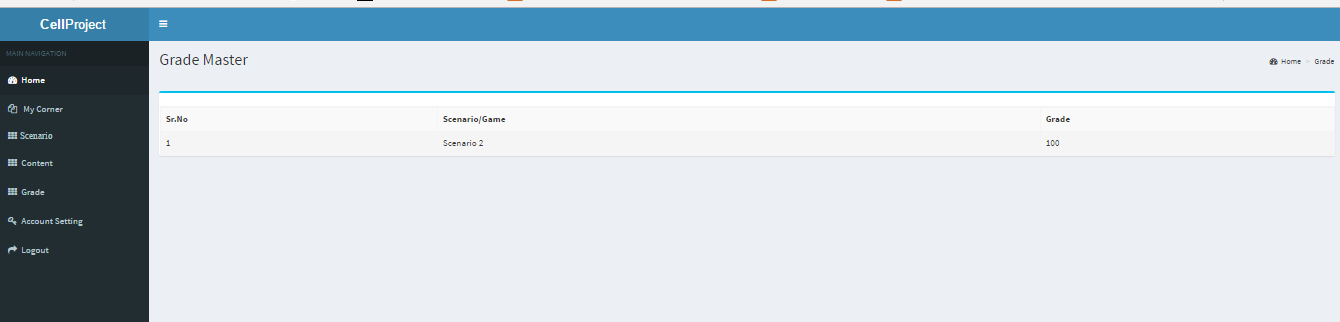
**Use of this Requirement:** Activity log screen consists of Detailed analysis of the scenario played by student to be graded by Instructor.

**Designed screen and output screen**

**1) grade Screen from admin login:**



**2) grade Screen from student login:**



**Future Enhancements**

There are wide range of opportunities can be found in terms of adding new features and making Cell-Project gaming platform more intuitive. Suggestible enhancements as follows:

**a) Point of creating scenario:** To implement uploading File to create entire scenario would be an automated version of creating individual contents and linking them to create a scenario which eventually termed as a Game.

**b) Visualization:** As video gaming and gaming platforms are emerging to the field of education for better understanding, a project like Cell Project can be modified by

* Adding few animations, graphics and also visual effects according to decision making visual effects in the intermediate levels of a scenario.
* By proper design of visual back grounds can triggers the curiosity for learning. For example: structure of human cell in the game home screen with animated texts/images for instructions to play.

**Source Code and References:**