**Unit Testing**

| **Unit Testing** | 1 |
| --- | --- |
| **Test Case Objective** | Checking duplicate qualification code |
| **Test Case Description** | This testing to test the University code if is existing, it will prompt error message “Duplicate Qualification Found”. |
| **Source Code** | if (isset($txtQualificationCode)){  $chkQualification = $cf->chkDupMaster($txtQualificationCode,"lf\_gbl\_qualification","qualification\_code");    if($chkQualification){  $strError .= "Duplicate Qualification Found";  }  }  echo "<script type='text/javascript'>alert('$strError');</script>";  echo "<script type='text/javascript'>alert('Record has been failed... Please try again...');window.location = '$redirect\_url\_fail'</script>";  public function chkDupMaster($str,$table,$col)  {  $txtReturn = '';  $cond = "where comp\_code = 'MY' and $col = '$str'";  $sqlChkDup = "SELECT COUNT(\*) AS countRow FROM $table $cond";  $stmt = $this->conn->prepare($sqlChkDup);  $stmt->execute();  if ($stmt->execute()) {  $stmt-> bind\_result($token2);  if($stmt-> fetch() )  {  $txtReturn = $token2;  }  // user existed  $stmt->close();  if($txtReturn == 1){  return true;  }else{  return false;  }  } else {  // user not existed  $stmt->close();  return false;  }  } |
| **Expected Result** | Pop up message: ‘Duplicate Qualification Found’  Pop up message: ‘Record has been failed... Please try again...’  Redirect to user main page. |
| **Outcome** |  |

**Unit Testing**

| **Unit Testing** | 2 |
| --- | --- |
| **Test Case Objective** | Setup Qualification (Edit Mode) Amend or add new Grade |
| **Test Case Description** | This testing is to test the qualification amend or add new on the grade without clear off whole line. |
| **Source Code** | function addGrade(){      var strGrade = document.getElementById("txtGrade").value.toUpperCase();      var strScore = document.getElementById("txtScore").value;      var strGradeSystem = document.getElementById("txtGradeSystem").value;      var strFlag = true;        var array = "";      var arrayCount = "";      var arryItem = "";      var tmpStr = "";      var newItem = false;      var piority = false;      var i;      if(strGrade.trim() != ""){              if(strScore.trim() != ""){       if(strGradeSystem.trim() != ""){          array = strGradeSystem.split(";");          arrayCount = array.length -1;          for(i=0;i<arrayCount;i++){              arrayItem = array[i].split("=");              if(strGrade == arrayItem[0]){                  tmpStr += strGrade + "=" + strScore + ";";                  piority = true;                  newItem = false;              }else{                  tmpStr += array[i] + ";";                  if(!piority){                      newItem = true;                  }              }          }          if(newItem){              tmpStr += strGrade + "=" + strScore + ";";          }          document.getElementById("txtGradeSystem").value = "";          document.getElementById("txtGradeSystem").value += tmpStr;          document.getElementById("txtGrade").value = "";          document.getElementById("txtScore").value = "";          strFlag=false;      }      if(strFlag){              document.getElementById("txtGradeSystem").value += strGrade + "=" + strScore + ";";              document.getElementById("txtGrade").value = "";              document.getElementById("txtScore").value = "";      }          }else{  alert("Score Cannot be Empty");  }          }else{              alert("Grade Cannot be Empty");          }  } |
| **Expected Result** | From Source : A=4.00;B=3.00;C=2.00;D=1.00;E=0.00; To Source : A=5.00;B=4.00;C=3.00;D=2.00;E=1.00;F=0.00; Popout Message : ‘Record Has been Successful’. |
| **Outcome** |  |

**Unit Testing**

| **Unit Testing** | 3 |
| --- | --- |
| **Test Case Objective** | Setup Qualification (Edit Mode) Amend or add new Grade |
| **Test Case Description** | This testing is to test the qualification amend or add new on the grade without clear off whole line. |
| **Source Code** | function addGrade(){      var strGrade = document.getElementById("txtGrade").value.toUpperCase();      var strScore = document.getElementById("txtScore").value;      var strGradeSystem = document.getElementById("txtGradeSystem").value;      var strFlag = true;        var array = "";      var arrayCount = "";      var arryItem = "";      var tmpStr = "";      var newItem = false;      var piority = false;      var i;      if(strGrade.trim() != ""){              if(strScore.trim() != ""){       if(strGradeSystem.trim() != ""){          array = strGradeSystem.split(";");          arrayCount = array.length -1;          for(i=0;i<arrayCount;i++){              arrayItem = array[i].split("=");              if(strGrade == arrayItem[0]){                  tmpStr += strGrade + "=" + strScore + ";";                  piority = true;                  newItem = false;              }else{                  tmpStr += array[i] + ";";                  if(!piority){                      newItem = true;                  }              }          }          if(newItem){              tmpStr += strGrade + "=" + strScore + ";";          }          document.getElementById("txtGradeSystem").value = "";          document.getElementById("txtGradeSystem").value += tmpStr;          document.getElementById("txtGrade").value = "";          document.getElementById("txtScore").value = "";          strFlag=false;      }      if(strFlag){              document.getElementById("txtGradeSystem").value += strGrade + "=" + strScore + ";";              document.getElementById("txtGrade").value = "";              document.getElementById("txtScore").value = "";      }          }else{  alert("Score Cannot be Empty");  }          }else{              alert("Grade Cannot be Empty");          }  } |
| **Expected Result** | From Source : A=4.00;B=3.00;C=2.00;D=1.00;E=0.00; To Source : A=5.00;B=4.00;C=3.00;D=2.00;E=1.00;F=0.00; |
| **Outcome** |  |

**Unit Testing**

| **Unit Testing** | 4 |
| --- | --- |
| **Test Case Objective** | Setup Qualification (Edit Mode) Amend or add new Subject |
| **Test Case Description** | This testing is to test the qualification amend or add new on the subject without clear off whole line. |
| **Source Code** | function addSubject(){  var strSubjectBefore = document.getElementById("drpSubjectBefore").value.toUpperCase();  var strSubjectAfter = document.getElementById("txtSubjectAfter").value.toUpperCase();  var strGradeSubject = document.getElementById("txtGradeSubject").value;  var strFlag = true;    var array = "";  var arrayCount = "";  var editMode = false;  var tmpStr = "";  var newItem = false;  var piority = false;  var i;  if(strSubjectAfter.trim() != ""){  if(strGradeSubject.trim() != ""){  array = strGradeSubject.split(";");  arrayCount = array.length -1;  for(i=0;i<arrayCount;i++){  if(strSubjectBefore != ""){  if(strSubjectBefore == array[i]){  tmpStr += strSubjectAfter + ";";  piority = true;  newItem = false;  }else{  tmpStr += array[i] + ";";  if(!piority){ newItem = true; }  }  }else{  if(strSubjectAfter == array[i]){  tmpStr += strSubjectAfter + ";";  piority = true;  newItem = false;  }else{  tmpStr += array[i] + ";";  if(!piority){ newItem = true; }  }  }  }  if(newItem){  tmpStr += strSubjectAfter + ";";  }  document.getElementById("txtGradeSubject").value = "";  document.getElementById("txtGradeSubject").value += tmpStr;  document.getElementById("drpSubjectBefore").value = "";  document.getElementById("txtSubjectAfter").value = "";  strFlag=false;  }  if(strFlag){  document.getElementById("txtGradeSubject").value += strSubjectAfter + ";";  document.getElementById("drpSubjectBefore").value = "";  document.getElementById("txtSubjectAfter").value = "";  }  }else{ alert("Subject Cannot be Empty"); }  } |
| **Expected Result** | From Source : SC;MM;ASC;AMM; To Source : SC;MATHEMATIC;ASC;AMM; |
| **Outcome** |  |

**Unit Testing**

| **Unit Testing** | 5 |
| --- | --- |
| **Test Case Objective** | Setup Qualification (Edit Mode) Subject Amend Button |
| **Test Case Description** | This testing is to test the qualification amend button validation the txtSubjectAfter is empty and prompt message to user what was error occur. |
| **Source Code** | function addSubject(){  var strSubjectBefore = document.getElementById("drpSubjectBefore").value.toUpperCase();  var strSubjectAfter = document.getElementById("txtSubjectAfter").value.toUpperCase();  var strGradeSubject = document.getElementById("txtGradeSubject").value;  var strFlag = true;    var array = "";  var arrayCount = "";  var editMode = false;  var tmpStr = "";  var newItem = false;  var piority = false;  var i;  if(strSubjectAfter.trim() != ""){  if(strGradeSubject.trim() != ""){  array = strGradeSubject.split(";");  arrayCount = array.length -1;  for(i=0;i<arrayCount;i++){  if(strSubjectBefore != ""){  if(strSubjectBefore == array[i]){  tmpStr += strSubjectAfter + ";";  piority = true;  newItem = false;  }else{  tmpStr += array[i] + ";";  if(!piority){ newItem = true; }  }  }else{  if(strSubjectAfter == array[i]){  tmpStr += strSubjectAfter + ";";  piority = true;  newItem = false;  }else{  tmpStr += array[i] + ";";  if(!piority){ newItem = true; }  }  }  }  if(newItem){  tmpStr += strSubjectAfter + ";";  }  document.getElementById("txtGradeSubject").value = "";  document.getElementById("txtGradeSubject").value += tmpStr;  document.getElementById("drpSubjectBefore").value = "";  document.getElementById("txtSubjectAfter").value = "";  strFlag=false;  }  if(strFlag){  document.getElementById("txtGradeSubject").value += strSubjectAfter + ";";  document.getElementById("drpSubjectBefore").value = "";  document.getElementById("txtSubjectAfter").value = "";  }  }else{ alert("Subject Cannot be Empty"); }  } |
| **Expected Result** | From Source : SC;MATHEMATIC;ASC;AMM;  Input : (empty string);  Popout message :’Subject Cannot Be Empty’ |
| **Outcome** |  |

**Unit Testing**

| **Unit Testing** | 6 |
| --- | --- |
| **Test Case Objective** | Setup Qualification (Edit Mode) Amend or add new Grade |
| **Test Case Description** | This testing is to test the qualification amend or add new on the grade without clear off whole line. |
| **Source Code** | function addGrade(){      var strGrade = document.getElementById("txtGrade").value.toUpperCase();      var strScore = document.getElementById("txtScore").value;      var strGradeSystem = document.getElementById("txtGradeSystem").value;      var strFlag = true;        var array = "";      var arrayCount = "";      var arryItem = "";      var tmpStr = "";      var newItem = false;      var piority = false;      var i;      if(strGrade.trim() != ""){              if(strScore.trim() != ""){       if(strGradeSystem.trim() != ""){          array = strGradeSystem.split(";");          arrayCount = array.length -1;          for(i=0;i<arrayCount;i++){              arrayItem = array[i].split("=");              if(strGrade == arrayItem[0]){                  tmpStr += strGrade + "=" + strScore + ";";                  piority = true;                  newItem = false;              }else{                  tmpStr += array[i] + ";";                  if(!piority){                      newItem = true;                  }              }          }          if(newItem){              tmpStr += strGrade + "=" + strScore + ";";          }          document.getElementById("txtGradeSystem").value = "";          document.getElementById("txtGradeSystem").value += tmpStr;          document.getElementById("txtGrade").value = "";          document.getElementById("txtScore").value = "";          strFlag=false;      }      if(strFlag){              document.getElementById("txtGradeSystem").value += strGrade + "=" + strScore + ";";              document.getElementById("txtGrade").value = "";              document.getElementById("txtScore").value = "";      }          }else{  alert("Score Cannot be Empty");  }          }else{              alert("Grade Cannot be Empty");          }  } |
| **Expected Result** | From Source : A=4.00;B=3.00;C=2.00;D=1.00;E=0.00;  Input : Grade = (empty string);  Popout message :’Grade Cannot Be Empty’  Input : Grade: ‘A’ , Score = (empty string);  Popout message :’Score Cannot Be Empty’ |
| **Outcome** |  |

**Unit Testing**

| **Unit Testing** | 7 |
| --- | --- |
| **Test Case Objective** | Setup Qualification (Edit Mode) Manipulate the Subject Drop Down List |
| **Test Case Description** | This testing is to test the qualification when existing subject, and update or add new subject will auto re-generate the subject into the drop down list |
| **Source Code** | function genDrpDwnListSubject(){  document.getElementById("drpSubjectBefore").options.length = 1;  var select = document.getElementById("drpSubjectBefore");  var strGradeSubject = document.getElementById("txtGradeSubject").value;  var i;    if(strGradeSubject.trim() != ""){  document.getElementById("drpSubjectBefore").disabled=false;  array = strGradeSubject.split(";");  arrayCount = array.length -1;  for(i = 0; i < arrayCount; i++) {  var option = document.createElement('option');  option.text = option.value = array[i];  select.add(option, 1);  }  }  } |
| **Expected Result** | From Source : SC;MATHEMATIC;ASC;AMM;  Input : Subject = ‘ENGLISH’  To Source : SC;MATHEMATIC;ASC;AMM;ENGLISH; |
| **Outcome** |  |

**Unit Testing**

| **Unit Testing** | 8 |
| --- | --- |
| **Test Case Objective** | Setup Qualification Check Field Only Numeric |
| **Test Case Description** | This testing is to test the qualification insert data, verified the certain field must be numeric value, and not for character. |
| **Source Code** | function chkNumeric(str){  if(isNaN(str.value)){  alert("(" + str.value + ") is not a number format.");  str.value = "";  }else{  str.value = str.value;  }  return str.value;  } |
| **Expected Result** | Input : Average best Of = A;  Popout message :’(A) is not a number format.’  Reset the input to empty . |
| **Outcome** |  |

**Unit Testing**

| **Unit Testing** | 9 |
| --- | --- |
| **Test Case Objective** | Checking duplicate university code |
| **Test Case Description** | This testing to test the University code if is existing, it will prompt error message “Duplicate University Code Found”. |
| **Source Code** | if (isset($txtUniversityCode)){  $chkUniversity = $cf->chkDupMaster($txtUniversityCode,"lf\_gbl\_university","uni\_code");    if($chkUniversity){  $strError .= "Duplicate University Found";  }  }  echo "<script type='text/javascript'>alert('$strError');</script>";  echo "<script type='text/javascript'>alert('Record has been failed... Please try again...');window.location = '$redirect\_url\_fail'</script>";  public function chkDupMaster($str,$table,$col)  {  $txtReturn = '';  $cond = "where comp\_code = 'MY' and $col = '$str'";  $sqlChkDup = "SELECT COUNT(\*) AS countRow FROM $table $cond";  $stmt = $this->conn->prepare($sqlChkDup);  $stmt->execute();  if ($stmt->execute()) {  $stmt-> bind\_result($token2);  if($stmt-> fetch() )  {  $txtReturn = $token2;  }  // user existed  $stmt->close();  if($txtReturn == 1){  return true;  }else{  return false;  }  } else {  // user not existed  $stmt->close();  return false;  }  } |
| **Expected Result** | Pop up message: ‘Duplicate University Found’  Pop up message: ‘Record has been failed... Please try again...’  Redirect to user main page. |
| **Outcome** |  |

**Unit Testing**

| **Unit Testing** | 10 |
| --- | --- |
| **Test Case Objective** | Setup University Check Field is Email |
| **Test Case Description** | This testing to test the Email is not a valid email, it will prompt error message “Invalid Email Address”. |
| **Source Code** | function chkValidEmail(str){  if(str.value != ""){  if(!str.value.includes("@")){  alert("("+str.value+") Invalid Email Address");  str.value = "";  }else{  str.value = str.value;  }  return str.value;  }  } |
| **Expected Result** | Input : E-mail = ‘help.edu.my’;  Popout message :’(help.edu.my) Invalid Email Address.’  Reset the input to empty . |
| **Outcome** |  |

**Unit Testing**

| **Unit Testing** | 11 |
| --- | --- |
| **Test Case Objective** | Setup University Check Field is Contact Number |
| **Test Case Description** | This testing to test the Email is not a valid email, it will prompt error message “Invalid Email Address”. |
| **Source Code** | function chkValidNumber(str){  if(str.value != ""){  if(str.value.length > 9){  if(str.value.length < 12){  if(isNaN(str.value)){  alert("("+str.value+") Invalid Phone Number");  str.value = "";  }else{  str.value = str.value.replace(".","");  str.value = str.value.replace("+","");  str.value = str.value.replace("-","");  str.value = str.value;  }  }else{  alert("("+str.value+") Invalid Phone Number Format");  str.value = "";  }  }else{  alert("("+str.value+") Invalid Phone Number Format");  str.value = "";  }  return str.value;  }  } |
| **Expected Result** | Input : Contact No = ‘012-3456789’ (below 10 digits) (no symbol)  Popout message :’(012-3456789) Invalid Phone Number Format.’  Reset the input to empty . |
| **Outcome** |  |