

**BIT302**

**Software Engineering Principles**

**DESIGN AND TESTING DOCUMENT ITERATION 2**

***MK SYSTEM***

**Student Name:** **LIM PEIR WEI Student ID: B1401800 E-mail: kevinklzai@gmail.com**

**Student Name: ZHOU YU JIE Student ID: B1402332 E-mail: 1539883891mitchie@gmail.com**

**15 Jalan Sri Semantan 1, Off Jalan Semantan, Bukit Damansara, 50490 Kuala Lumpur, Malaysia.**

**Tel : 603-2716 2000, Fax : 603 - 2095 7100**

Table of Contents

[Identification of Problems in Iteration 1 and Solutions 3](#_Toc5614348)

[Updates on the analysis/design documentation 4](#_Toc5614349)

[Update Design Class Diagram 5](#_Toc5614350)

[Update ERD 6](#_Toc5614351)

[Update Database Design 7](#_Toc5614352)

[Update Site Map 15](#_Toc5614353)

[Screenshot 16](#_Toc5614354)

[Division of Work 24](#_Toc5614355)

[Test Objectives 24](#_Toc5614356)

[Update Test Plan 25](#_Toc5614357)

[Test Cases 26](#_Toc5614358)

[Unit Testing 26](#_Toc5614359)

[Integration Testing 47](#_Toc5614360)

[System Testing 50](#_Toc5614361)

[Installation Testing 58](#_Toc5614362)

[Test Analysis Report 60](#_Toc5614363)

[Update Gantt Chart 61](#_Toc5614364)

[Conclusion 63](#_Toc5614365)

### Identification of Problems in Iteration 1 and Solutions

**Problems**

During Iteration 1, we have planned to finish the use cases: “Setup Qualifications, Register University, Record Subject, and Record University Admin”.

Firstly, we almost finished these use cases, but we got an issue when we try to “add score” during the “Setup Qualification” process, we only can “add grade” to the grade system.

Secondly, we made a confusion about the test cases which are unit testing, system testing, and integration testing and did wrongly for our iteration.

Thirdly, we haven’t make a wonderful user interface design to make it ease of use, to provide user with a good user experience.

Lastly, we process a wrong ERD data design and test analysis report during the design specification.

**Solutions**

We have solved these issues in Iteration 2. At first, we added a GUI widget function in “Setup Qualification” which is “checkbox” to permits the SAS admin to make a binary choice between “add score” and “add grade”. The checkbox is displayed as the square box that is selected when activated. The checked attribute can be set after the selection of SAS admin, with the JavaScript.

And then, we asked lecturer and other group members about the test cases how to process. We made a discussion about it aimed to make it better in Iteration 2.

Moreover, in order to improve the user interface design, we will through the following approaches. Unifying visual standards to improve recognition, enhance the visual impact of the main action point and enhance its comparability in the page. Improve the consistency of interface elements and reduce the cost of user learning, designing a more pleasant interface, improve the visual level of the page and enhance readability. Based on these ways to offer a better user experience for users.

What’s more, we reviewed how to draw the ERD diagram to make a better data design. And we asked lecturer for the test analysis report part how to write, we discussed that to make a better test analysis report.

### Updates on the analysis/design documentation

Introduction

In Iteration 2 of our design specification document, we will update the description and graphical documentation for MK System. The design class diagram, entity relationship diagram (ERD), site map and so forth will be updated in Iteration 2. Here are also offered the system design newly created which based on database design and system sequence diagram of use cases to be developed for Iteration 2.

We have added one more use case which is “Check Application Status” to help applicant check their application status for applying for the programme. This function aims to help the applicant to get the application status information as soon as possible, let them know whether they are approved by the university or not.



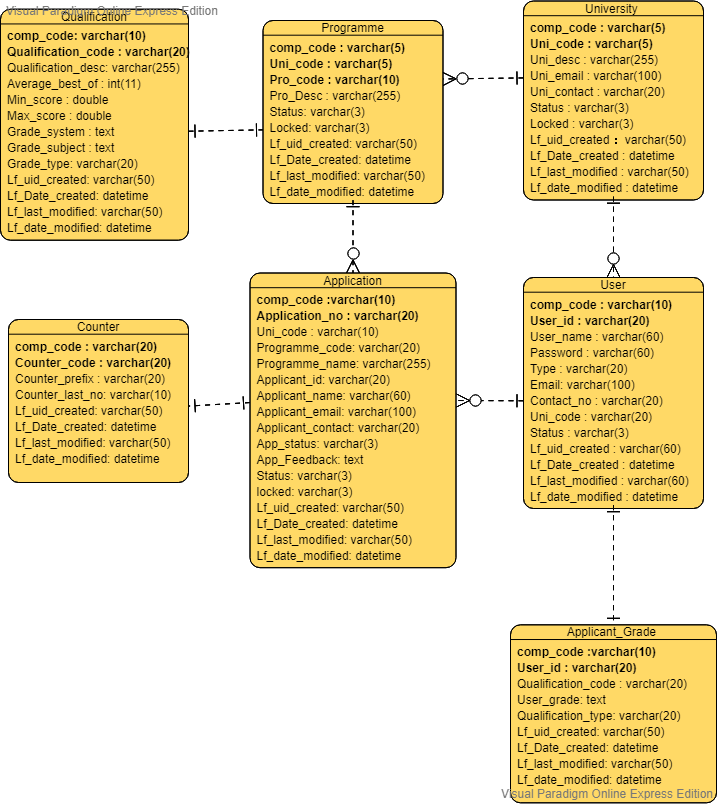
Functional Requirement

|  |  |
| --- | --- |
| Use Case | Check Application Status |
| Actor(s) | Applicant |
| Description | Allow applicant to check the application status throw the MK System Website. |

##### Update Design Class Diagram



##### Update ERD



##### Update Database Design

Lf\_gbl\_user Table (User of MK System)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Filed Name | DataType | Constraint | Example | Description |
| comp\_code | Varchar(10) | PK | MY | This is to specific this system using in ‘MY’ branch |
| User\_id | Varchar(20) | PK | HELPAdmin | This is the unique ID for every user. |
| User\_name | Varchar(60) | Not Null | HELP ADMIN | Description of the user. |
| Password | Varchar(60) | Not Null | 123456 | Store the password user key in |
| Type | Varchar(20) | Not Null | UNI | Store the type of user belong to (APP = Applicant, UNI = University) |
| Email | Varchar(100) | Not Null | helpadmin@help.edu.my | This is the unique email for each user in the system. |
| Contact\_no | Varchar(20) | Not Null | 012-3456789 | Store the user Contact Number |
| Uni\_code | Varchar(20) | Not Null | HELP | To define this UNI Admin is belong to which university. |
| Status | Varchar(3) | Not Null | N | This is to Indicate the user status is (T = Terminate, N = Normal) |
| Lf\_uid\_created | Varchar(60) | Not Null | SYSTEM | Automatic Generate by the SYSTEM |
| Lf\_Date\_created | Datetime | Not Null | 2019-03-04 | This date is belong to the user create follow the time in the server. |
| Lf\_last\_modified | Varchar(60) | Not Null | - | When Edit will update the user in this field |
| Lf\_date\_modified | Datetime | Not Null | - | When Edit will update the user in this field |

Lf\_gbl\_University Table (University of MK System)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Filed Name | DataType | Constraint | Example | Description |
| comp\_code | Varchar(5) | PK | MY | This is to specific this system using in ‘MY’ branch |
| Uni\_code | Varchar(5) | PK | HELP | This is the unique ID for every University. |
| Uni\_desc | Varchar(255) | Not Null | HELP Univeristy | Description of the University. |
| Uni\_email | Varchar(100) | Not Null | HELP@HELP.EDU.My | Email address of the University |
| Uni\_contact | Varchar(20) | Not Null | 0123455789 | Contact Number of the University |
| Status | Varchar(3) | Not Null | N | This is to Indicate the user status is (T = Terminate, N = Normal) |
| Locked | Varchar(3) | Not Null | 0 | To check this Uni\_code does bind with other stuff |
| Lf\_uid\_created | Varchar(50) | Not Null | SYSTEM | Automatic Generate by the SYSTEM |
| Lf\_Date\_created | Datetime | Not Null | 2019-03-04 | This date is belong to the user create follow the time in the server. |
| Lf\_last\_modified | Varchar(50) | Not Null | - | When Edit will update the user in this field |
| Lf\_date\_modified | Datetime | Not Null | - | When Edit will update the user in this field |

Lf\_gbl\_qualification Table (Qualification of MK System)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Filed Name | DataType | Constraint | Example | Description |
| comp\_code | Varchar(10) | PK | MY | This is to specific this system using in ‘MY’ branch |
| Qualification\_code | Varchar(20) | PK | STPM | This is the unique ID for every qualification. |
| Qualification\_desc | Varchar(255) | Not Null | Sijil Tinggi Persekolahan Malaysia | Description of the Qualification. |
| Average\_best\_of | Int(11) | Not Null | 3 | Average the grade by using how many subject. |
| Min\_score | Double | Not Null | 0.0 | Store the type of user belong to (APP = Applicant, UNI = University) |
| Max\_score | Double | Not Null | 4.0 | This is the unique email for each user in the system. |
| Grade\_system | Text | Not Null | A=4.00;A-=3.67;B+=3.33;B=3.00;B-=2.67;C+=2.33;C=2.00;C-=1.67;D+=1.33;D=1.0;F=0.00; | The grade indicate how many point obtain. |
| Grade\_subject | Text | Not Null | Science,Mathematic,Business | This qualification contain subject for user select. |
| Grade\_type | Varchar(20) | Not Null | SCORE / GRADE | This is to determine the qualification running grade or score calculation. |
| Lf\_uid\_created | Varchar(50) | Not Null | SYSTEM | Automatic Generate by the SYSTEM |
| Lf\_Date\_created | Datetime | Not Null | 2019-03-04 | This date is belong to the user create follow the time in the server. |
| Lf\_last\_modified | Varchar(50) | Not Null | - | When Edit will update the user in this field |
| Lf\_date\_modified | Datetime | Not Null | - | When Edit will update the user in this field |

lf\_applicant\_grade Table (Applicant\_Grade of MK System)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Filed Name | DataType | Constraint | Example | Description |
| comp\_code | Varchar(10) | PK | MY | This is to specific this system using in ‘MY’ branch |
| User\_id | Varchar(20) | PK | B1401800 | This is the unique ID for every applicant |
| Qualification\_code | Varchar(20) | Not Null | STPM | The Type of qualification |
| User\_grade | Text | Not Null | BM=A;BI=B;BC=C;BIO=C;  0-100%; | The user qualification score or grade |
| Qualification\_type | Varchar(20) | Not Null | SCORE / GRADE | This is to determine the qualification running grade or score calculation. |
| Lf\_uid\_created | Varchar(50) | Not Null | SYSTEM | Automatic Generate by the SYSTEM |
| Lf\_Date\_created | Datetime | Not Null | 2019-03-04 | This date is belong to the user create follow the time in the server. |
| Lf\_last\_modified | Varchar(50) | Not Null | - | When Edit will update the user in this field |
| Lf\_date\_modified | Datetime | Not Null | - | When Edit will update the user in this field |

lf\_gbl\_application Table (Application of MK System)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Filed Name | DataType | Constraint | Example | Description |
| comp\_code | Varchar(10) | PK | MY | This is to specific this system using in ‘MY’ branch |
| Application\_no | Varchar(20) | PK | HELPAPP00001 | This is the unique running number for every application |
| Uni\_code | Varchar(10) | Not Null | HELP | Determine this application from which university |
| Programme\_code | Varchar(20) | Not Null | BMC | Programme that user want to enrol |
| Programme\_name | Varchar(255) | Not Null | Bachelor In Mobile Computing | The Full Description of the Programme |
| Applicant\_id | Varchar(20) | Not Null | B1401800 | User ID to apply the Programme |
| Applicant\_name | Varchar(60) | Not Null | B1401800 User | User Name to apply the Programme |
| Applicant\_email | Varchar(100) | Not Null | [B1401800@gmail.com](mailto:B1401800@gmail.com) | User Email to apply the Programme |
| Applicant\_contact | Varchar(20) | Not Null | 0123456789 | User Contact to apply the Programme |
| App\_status | Varchar(3) | Not Null | C (Cancel) , R (Reject)  A (Approve), P (Pending) | Determine the Application Status return to the user. |
| App\_Feedback | Text | Not Null | Congratulation, You have been successful enrol into this Programme. | This is the feedback from University Admin to the Applicant. |
| Status | Varchar(3) | Not Null | C (Cancel)  R (Reject)  A (Approve)  P (Pending) | Determine the status of application |
| locked | Varchar(3) | Not Null | 0 | To check this Application\_no does bind with other stuff |
| Lf\_uid\_created | Varchar(50) | Not Null | SYSTEM | Automatic Generate by the SYSTEM |
| Lf\_Date\_created | Datetime | Not Null | 2019-03-04 | This date is belong to the user create follow the time in the server. |
| Lf\_last\_modified | Varchar(50) | Not Null | - | When Edit will update the user in this field |
| Lf\_date\_modified | Datetime | Not Null | - | When Edit will update the user in this field |

##### 

lf\_gbl\_programme Table (Programme of MK System)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Filed Name | DataType | Constraint | Example | Description |
| comp\_code | Varchar(5) | PK | MY | This is to specific this system using in ‘MY’ branch |
| Uni\_code | Varchar(5) | PK | HELP | This is the University of the programme |
| Pro\_code | Varchar(10) | PK | BMC | Programme code is unique by every single University. |
| Pro\_Desc | Varchar(255) | Not Null | Bachelor In Mobile Computing | The Full Description of the Programme |
| Status | Varchar(3) | Not Null | N | Determine the status of Programme |
| Locked | Varchar(3) | Not Null | 0 | To check this Application\_no does bind with other stuff |
| Lf\_uid\_created | Varchar(50) | Not Null | SYSTEM | Automatic Generate by the SYSTEM |
| Lf\_Date\_created | Datetime | Not Null | 2019-03-04 | This date is belong to the user create follow the time in the server. |
| Lf\_last\_modified | Varchar(50) | Not Null | - | When Edit will update the user in this field |
| Lf\_date\_modified | Datetime | Not Null | - | When Edit will update the user in this field |

lf\_gbl\_Counter Table (Counter of MK System)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Filed Name | DataType | Constraint | Example | Description |
| comp\_code | Varchar(20) | PK | MY | This is to specific this system using in ‘MY’ branch |
| Counter\_code | Varchar(20) | PK | HELP | This is the unique ID for every counter |
| Counter\_prefix | Varchar(20) | Not Null | HELP | This counter use what prefix for their document no |
| Counter\_last\_no | Varchar(10) | Not Null | 100000 | This is the document running number. |
| Lf\_uid\_created | Varchar(50) | Not Null | SYSTEM | Automatic Generate by the SYSTEM |
| Lf\_Date\_created | Datetime | Not Null | 2019-03-04 | This date is belong to the user create follow the time in the server. |
| Lf\_last\_modified | Varchar(50) | Not Null | - | When Edit will update the user in this field |
| Lf\_date\_modified | Datetime | Not Null | - | When Edit will update the user in this field |

##### Update Site Map

**Home Page**

**Login**

**Sign Up**

**SAS Admin Login**

**Applicant Login**

**University Admin Login**

**Applicant Register**

**Setup Menu Option**

**Apply Programme**

**Record Programme**

**Setup Qualification**

**Review Application**

**Applicant Qualification**

**Setup User**

**Setup University**

**Application Enquiry**

### Screenshot

|  |
| --- |
| **MainPage** |
|  |

|  |
| --- |
| **Home** |
|  |

|  |
| --- |
| **Register** |
|  |

|  |
| --- |
| **Applicant First Time Insert Qualifiaction (Applicant Level)** |
|  |

|  |
| --- |
| **Login** |
|  |

|  |
| --- |
| **Applicant Default Page (Applicant Level)** |
|  |

|  |
| --- |
| **Applicant Qualification (Applicant Level)** |
|  |

|  |
| --- |
| **Application Enquiry (Applicant Level)** |
|  |

|  |
| --- |
| **Apply Programme (Applicant Level)** |
|  |

|  |
| --- |
| **Setup Programme (University Level)** |
|  |

|  |
| --- |
| **Review Application (University Level)** |
|  |

|  |
| --- |
| **Review Application Form (University Level)** |
|  |

|  |
| --- |
| **Setup Qualification [Part 1](SAS Level)** |
|  |

|  |
| --- |
| **Setup Qualification [Part 2](SAS Level)** |
|  |

|  |
| --- |
| **Setup University (SAS Level)** |
|  |

|  |
| --- |
| **Setup User Code (SAS Level)** |
|  |

### Division of Work

|  |  |  |
| --- | --- | --- |
| Use Case Name | Use Case Description | Assigned to |
| Record Programme | Implement the table input filed to let university admin record a programme. | Zhou Yu Jie |
| Review Application | Provide a table to show the application list and add a button to allow university admin to edit the application status. | Lim Peir Wei |
| Register Applicant | Provide an input filed to get the applicant profile information data. | Lim Peir Wei |
| Apply Programme | Implement the input filed table to get the detail information for applicant apply. | Zhou Yu Jie |
| Check Application Status | Add an “Application Enquiry” table allow applicant to check the application status after their application. | Lim Peir Wei |
| Record Qualification | Implement a table filed to let applicant input the qualification details. | Lim Peir Wei |

### Test Objectives

1. To ensure that all components work in accordance with functional specifications.
2. To make sure that the achievement of non-functional system requirements in the system testing.
3. To test the new functions which haven’t implemented in Iteration 1.
4. To search any errors, bugs or faults of the system.
5. To ensure that all the pages in the system able to link correctly.
6. To get confidence and offer information about the quality level.
7. To find the defects that developers may make during developing website.
8. To verify the consistency of functional requirements and plan with determinations and structure particulars.

### Update Test Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Technique | Approach | Use Case | Schedule | Aim |
| Unit Testing | White Box | Record Programme & Register Applicant & Apply Programme & Check Application Status & Record Qualification | 20/03 ~ 27/03 | * It's easier to find errors in the code of small parts. * Provide flexibility while encoding components, especially when changing code. * Make sure all the parts of the code are correct before consolidating them. |
| Integration Testing | Black Box | Record Programme &  Apply Programme | 27/03 ~ 03/04 | * To test the overall structure of whole system to make sure that modules of unit testing able to work well after combining. * To make sure that proper transition to other pages in correct way. |
| System Testing | Black Box | Record Programme & Register Applicant & Apply Programme & Check Application Status & Record Qualification | 03/04 ~ 06/04 | * To ensure that all the use cases in MK System website enable to meet the functional and non-functional requirements. |

### Test Cases

##### Unit Testing

| **Unit Testing** | 1 |
| --- | --- |
| **Test Case Objective** | Programme code will not populate |
| **Test Case Description** | This testing to test the user unable to select programme, if the user haven’t select university. |
| **Input** | University = “” |
| **Source Code** | xhttp = new XMLHttpRequest();  xhttp.onreadystatechange = function() {  if (this.readyState == 4 && this.status == 200) {  strReturn = this.responseText;  tmpStr = strReturn.split("||");  arrayCount = tmpStr.length -1;  document.getElementById("drpProgramme").options.length = 1;  var select = document.getElementById("drpProgramme");  var x;  for(i = 0; i < arrayCount; i++){  tmpStrItem = tmpStr[i].split(",");  var option = document.createElement('option');  option.value = tmpStrItem[2].toUpperCase();  option.text = tmpStrItem[3].toUpperCase();  select.add(option, 1);  }  document.getElementById("drpProgramme").disabled=false;  }  }; |
| **Expected Result** | Unable to open the drop down list in programme. |
| **Outcome** |  |

| **Unit Testing** | 2 |
| --- | --- |
| **Test Case Objective** | Return data from Ajax |
| **Test Case Description** | This testing to test the data pass in by javascript, using ajax method to post back the current page. |
| **Input** | University = “HELP UNIVERSITY” |
| **Source Code** | function getProgramme() {  var xhttp;  var strTable = 'lf\_gbl\_programme';  var strcolName = 'uni\_code';  var strColVal = '';  var tmpStr = '';  var tmpStrItem = '';  var strReturn = '';  var arrayCount;  var i;  strColVal = document.getElementById("drpUni").value;  document.getElementById("drpProgramme").value = "";  if (strColVal== ""){  document.getElementById("drpProgramme").disabled=true;  document.getElementById("drpProgramme").value = "";  document.getElementById("txtProgrammeDesc").value = "";  document.getElementById("btnSubmit").disabled=true;  }else{  xhttp = new XMLHttpRequest();  xhttp.onreadystatechange = function() {  if (this.readyState == 4 && this.status == 200) {  strReturn = this.responseText;  tmpStr = strReturn.split("||");  arrayCount = tmpStr.length -1;  document.getElementById("drpProgramme").options.length = 1;  var select = document.getElementById("drpProgramme");  var x;  for(i = 0; i < arrayCount; i++){  tmpStrItem = tmpStr[i].split(",");  var option = document.createElement('option');  option.value = tmpStrItem[2].toUpperCase();  alert(tmpStrItem[3].toUpperCase());  option.text = tmpStrItem[3].toUpperCase();  select.add(option, 1);  }  document.getElementById("drpProgramme").disabled=false;  }  };  }  xhttp.open("GET", "lf\_drpdown\_function.php?v="+strColVal+"&t="+strTable+"&c="+strcolName, true);  xhttp.send();  } |
| **Expected Result** | Popup message = “BACHELOR IN BUSINESS E-COMMERCE”  Popup message = “BACHELOR IN INFORMATION TECHNOLOGY”  Programme drop down menu contain  1)BACHELOR IN INFORMATION TECHNOLOGY  2)BACHELOR IN BUSINESS E-COMMERCE |
| **Outcome** |  |

| **Unit Testing** | 3 |
| --- | --- |
| **Test Case Objective** | Grade re-arrange back in order for user easy view in proper way. |
| **Test Case Description** | This testing to test the user grade has been save in to database, when user hope to edit on the data, the subject and grade will sort into some textbox and display. |
| **Input** | Applicant ID =”B1401800”  btnEdit = true (when onlick) |
| **Source Code** | function showhideDIVEdit(){  var number;  var i;  number = document.getElementById("txtTotalSubject").value;  number = Number(number) +1;  var existing = document.getElementById("txtTextArea").value;  var existingdata = existing.split(";");    for (i = 0; i < existingdata.length - 1;i++){  showDIV(number);  document.getElementById("txtScore"+number).value = existingdata[i];  number += 1;  }    document.getElementById("txtTotalSubject").value = number;  // this is to lock / unlock the field which disabbled  document.getElementById("btnEdit").disabled = true;  document.getElementById("drpQualification").disabled = false;  document.getElementById("drpSubject").disabled = false;  document.getElementById("txtScore").disabled = false;  document.getElementById("btnAmend").disabled = false;  document.getElementById("btnSubmit").style.display='';  getGradeSubject();  } |
| **Expected Result** | The result will be from 1 line sentences, sort into a textbox, and display to the user  Textbox1 = “HISTORY=D”.  Textbox2 = “BM=C”.  Textbox3 = “BC=B”.  Textbox4 = “MATCHEMATIC=A”.  Textbox5 = “SCIENCE=A”.  Textbox6 = “BI=A-“  txtTotalSubject = ‘6’  btnSubmit = visible (show out) |
| **Outcome** | **BEFORE**    **AFTER** |

| **Unit Testing** | 4 |
| --- | --- |
| **Test Case Objective** | Reset information |
| **Test Case Description** | This testing to test if the user edit, and change the qualification, it will reset everything in the screen. Such as (reset grade and textfield contain subject and grade) |
| **Input** | Qualification = “A LEVEL CERTIFICATE” |
| **Source Code** | function getGradeSubject() {  var xhttp;  var strTable = 'lf\_gbl\_qualification';  var strcolName = 'qualification\_code';  var strColVal = '';  var tmpStr = '';  var tmpStrItem = '';  var strReturn = '';  var arrayCount;  var i;  strColVal = document.getElementById("drpQualification").value;  if (strColVal== ""){  document.getElementById("btnSubmit").disabled=true;  document.getElementById("drpSubject").options.length=1;  document.getElementById("drpSubject").disabled=true;  }else{  xhttp = new XMLHttpRequest();  xhttp.onreadystatechange = function() {  if (this.readyState == 4 && this.status == 200) {  strReturn = this.responseText;  tmpStr = strReturn.split(",");  arrayCount = tmpStr.length -1;    document.getElementById("drpSubject").options.length = 1;  var select = document.getElementById("drpSubject");  document.getElementById("txtMinimumSubject").value=tmpStr[3];  document.getElementById("txtCurrentGradingType").value=tmpStr[8];  if(strColVal != document.getElementById("txtCurrentQualification").value){  document.getElementById("txtTextArea").value = "";  showDIV(0);  }  tmpStrItem = tmpStr[7].split(";");  document.getElementById("btnSubmit").disabled=true;  arrayItemCount = tmpStrItem.length -1;  for(i = 0; i < arrayItemCount; i++){  var option = document.createElement('option');  option.value = option.text = tmpStrItem[i].toUpperCase();  select.add(option, 1);  }  document.getElementById("drpSubject").disabled=false;  document.getElementById("txtScore").disabled=false;  document.getElementById("btnAmend").disabled=false;  // alert("This Qualification required (" + tmpStr[3] + ") to proceed.");  }  };  }  xhttp.open("GET", "lf\_drpdown\_function.php?v="+strColVal+"&t="+strTable+"&c="+strcolName, true);  xhttp.send();  } |
| **Expected Result** | Reset everything in the page  Total Subject = ‘ ’ Grade = ‘’ |
| **Outcome** | **BEFORE**    **AFTER** |

| **Unit Testing** | 5 |
| --- | --- |
| **Test Case Objective** | Autogenerate the subject List |
| **Test Case Description** | This testing to test subject list is based on the qualification grading Subject |
| **Input** | Qualification = “A LEVEL CERTIFICATE” |
| **Source Code** | function getGradeSubject() {  var xhttp;  var strTable = 'lf\_gbl\_qualification';  var strcolName = 'qualification\_code';  var strColVal = '';  var tmpStr = '';  var tmpStrItem = '';  var strReturn = '';  var arrayCount;  var i;  strColVal = document.getElementById("drpQualification").value;  if (strColVal== ""){  document.getElementById("btnSubmit").disabled=true;  document.getElementById("drpSubject").options.length=1;  document.getElementById("drpSubject").disabled=true;  }else{  xhttp = new XMLHttpRequest();  xhttp.onreadystatechange = function() {  if (this.readyState == 4 && this.status == 200) {  strReturn = this.responseText;  tmpStr = strReturn.split(",");  arrayCount = tmpStr.length -1;    document.getElementById("drpSubject").options.length = 1;  var select = document.getElementById("drpSubject");  document.getElementById("txtMinimumSubject").value=tmpStr[3];  document.getElementById("txtCurrentGradingType").value=tmpStr[8];  if(strColVal != document.getElementById("txtCurrentQualification").value){  document.getElementById("txtTextArea").value = "";  showDIV(0);  }  tmpStrItem = tmpStr[7].split(";");  document.getElementById("btnSubmit").disabled=true;  arrayItemCount = tmpStrItem.length -1;  for(i = 0; i < arrayItemCount; i++){  var option = document.createElement('option');  option.value = option.text = tmpStrItem[i].toUpperCase();  select.add(option, 1);  }  document.getElementById("drpSubject").disabled=false;  document.getElementById("txtScore").disabled=false;  document.getElementById("btnAmend").disabled=false;  // alert("This Qualification required (" + tmpStr[3] + ") to proceed.");  }  };  }  xhttp.open("GET", "lf\_drpdown\_function.php?v="+strColVal+"&t="+strTable+"&c="+strcolName, true);  xhttp.send();  } |
| **Expected Result** | Subject List contain 1)ADD.SCIENCE 2)SCIENCE  3)ADD.MATHEMATIC  4)MATHEMATIC  5)ENGLISH |
| **Outcome** |  |

| **Unit Testing** | 6 |
| --- | --- |
| **Test Case Objective** | Applicant Enquiry Form |
| **Test Case Description** | This testing is to check application status of the applicant has been apply |
| **Input** | Applicant ID =”B1401800” |
| **Source Code** | <?php  $stmt = $conn->prepare("select lf\_date\_created,uni\_code,application\_no, programme\_name, app\_status from lf\_gbl\_application where comp\_code = ? and applicant\_id = ? order by lf\_date\_created");  $stmt->bind\_param("ss",$comcode,$Session\_UserID);  $stmt->execute();  $stmt->bind\_result($token2,$token3,$token4,$token5,$token6);  while ( $stmt-> fetch() ) { ?>  <tr><td width ='5%'><?php echo $SeqNo ?></td>  <td width='15%'><?php echo date('d/m/Y',strtotime($token2)) ?></td>  <td width='20%'><?php echo $token3 ?></td>  <td width='20%'><?php echo $token4 ?></td>  <td width='35%'><?php echo $token5 ?></td>  <td width='20%'><?php echo $token6 ?></td></tr>  <?php  $SeqNo += 1;}  $stmt->close(); ?> |
| **Expected Result** | Display application result in a table format  (no | Date | University | Application No | Programme | Status) eg.  (1 | 12/03/2019 | HELP | HELP100001 | BACHELOR IN BUSINESS E-COMMERCE | A) (2 | 12/03/2019 | TARUC | TARUC100001| BACHELOR IN BUSINESS E-COMMERCE | P) |
| **Outcome** |  |

| **Unit Testing** | 7 |
| --- | --- |
| **Test Case Objective** | Applicant Qualification (Edit Button) |
| **Test Case Description** | This testing is to check when applicant want to edit on the grade / score, data will be populate into the textbox for easy viewing purpose. |
| **Input** | Applicant ID =”B1401800”  Button onlick (Edit) |
| **Source Code** | function showhideDIVEdit(){  var number;  var i;  number = document.getElementById("txtTotalSubject").value;  number = Number(number) +1;  var existing = document.getElementById("txtTextArea").value;  var existingdata = existing.split(";");    for (i = 0; i < existingdata.length - 1;i++){  showDIV(number);  document.getElementById("txtScore"+number).value = existingdata[i];  number += 1;  }  number -= 1; // this is because last record still adding 1  document.getElementById("txtTotalSubject").value = number;  // this is to lock / unlock the field which disabbled  document.getElementById("btnEdit").disabled = true;  document.getElementById("drpQualification").disabled = false;  document.getElementById("drpSubject").disabled = false;  document.getElementById("txtScore").disabled = false;  document.getElementById("btnAmend").disabled = false;  document.getElementById("btnSubmit").style.display='';  getGradeSubject();  } |
| **Expected Result** | GradeSystem = “AMM=50;MATHEMATIC=60;ASC=30;SC=50;”  Assign into textbox  textBox1 = “AMM=50”  textBox2 = “MATHEMATIC =60”  textBox3 = “ASC =30”  textBox4 = “SC =50” |
| **Outcome** | **BEFORE**    **AFTER** |

| **Unit Testing** | | 8 |
| --- | --- | --- |
| **Test Case Objective** | | Applicant Qualification (Edit Button) |
| **Test Case Description** | | This testing is to check when applicant want to edit on the qualification code |
| **Input** | | Applicant ID =”SIJIL TINGGI PERSEKOLAHAN MALAYSIA” |
| **Source Code** | | function getGradeSubject() { …  strColVal = document.getElementById("drpQualification").value;  if(strColVal != document.getElementById("txtCurrentQualification").value){  document.getElementById("txtTextArea").value = "";  showDIV(0);}…  } |
| **Expected Result** | | gradeSystem = “”  textbox1 = “” (hide)  textbox2 = “” (hide)  textbox3 = “” (hide)  textbox4 = “” (hide) |
| **Outcome** | | **BEFORE**    **AFTER** |
| **Unit Testing** | 9 | |
| **Test Case Objective** | Application Show Feedback (Application Enquiry) | |
| **Test Case Description** | This testing is to check when the application has been response with feedback. | |
| **Input** | applicationNo = “HELP100005”  button onclick(VIEW) | |
| **Source Code** | <input type="button" name="btnView" id="btnView" value="View" onclick="showFeedback('<?php echo $token7;?>');">  function showFeedback(str){  alert(str);  } | |
| **Expected Result** | Popup Message = “You Have Been Approve” | |
| **Outcome** |  | |

| **Unit Testing** | 10 |
| --- | --- |
| **Test Case Objective** | Show Applicant qualification |
| **Test Case Description** | This testing is to return applicant qualification. |
| **Input** | applicantID = “B1501802”  button onclick(VIEW) |
| **Source Code** | function showFullGrade(){  var grade = document.getElementById("txtAppGrade").value;  var applicantID = document.getElementById("txtApplicantID").value;  var gradeArray = grade.split(";")  var final = '(' + applicantID + ') \n \n' ;  var i;  for(i = 0; i<(gradeArray.length-1);i++){  final += gradeArray[i] + "\n";  }  alert(final);  } |
| **Expected Result** | Popup Message = “  (B1501802)  ADDITIONAL SCIENCE=C  SCIENCE=B  ADDITIONAL MATHEMATIC=A  MATHEMATIC=A  HISTORY=C  BC=C  BI=B  BM=C;” |
| **Outcome** |  |

| **Unit Testing** | 11 |
| --- | --- |
| **Test Case Objective** | Calculate Applicant qualification |
| **Test Case Description** | This testing is to calculate applicant qualification result best of average |
| **Input** | applicantID = “B1501802”  button onclick(VIEW) |
| **Source Code** | function calculateAverageRatingScore(){  var appQualification = document.getElementById("txtAppGrade").value;  var qualificationCount = appQualification.split(";");  var arrayListAppQualification=[];  var i;  var x;  var totalScore = 0;  var FinalBestOfAverage= 0.00;    var bestOfAverage = document.getElementById("txtGradeBestOfAverage").value;  var gradingSystem = document.getElementById("txtGradeSystem").value;  for(i=0;i< qualificationCount.length-1;i++){  var qualificationInfo = qualificationCount[i].split("=");  arrayListAppQualification.push(qualificationInfo[1]);  }  arrayListAppQualification.reverse();  var gradingSystemCount = gradingSystem.split(";");  var gradeSystemInfo = '';  for(i=0; i< bestOfAverage;i++){  totalScore += parseFloat(arrayListAppQualification[i]);  }  FinalBestOfAverage = totalScore / bestOfAverage;  document.getElementById("txtBestOfAverage").value = FinalBestOfAverage;  } |
| **Expected Result** | Best of average field = “3.6666666666666665” |
| **Outcome** |  |

| **Unit Testing** | 12 |
| --- | --- |
| **Test Case Objective** | Lock Programme code |
| **Test Case Description** | This testing to test the user unable to select programme, if the user haven’t select university. |
| **Input** | University = “” |
| **Source Code** | xhttp = new XMLHttpRequest();  xhttp.onreadystatechange = function() {  if (this.readyState == 4 && this.status == 200) {  strReturn = this.responseText;  tmpStr = strReturn.split("||");  arrayCount = tmpStr.length -1;  document.getElementById("drpProgramme").options.length = 1;  var select = document.getElementById("drpProgramme");  var x;  for(i = 0; i < arrayCount; i++){  tmpStrItem = tmpStr[i].split(",");  var option = document.createElement('option');  option.value = tmpStrItem[2].toUpperCase();  option.text = tmpStrItem[3].toUpperCase();  select.add(option, 1);  }  document.getElementById("drpProgramme").disabled=false;  }  }; |
| **Expected Result** | Unable to open the drop down list in programme. |
| **Outcome** |  |

| **Unit Testing** | 13 |
| --- | --- |
| **Test Case Objective** | Unlock Submit Button |
| **Test Case Description** | This testing to test the user fulfil the criteria of the page, the submit button will unlock. |
| **Input** | University = “SEGI COLLEGE”  Programme = “BACHELOR IN MOBILE COMPUTING” |
| **Source Code** | function unlockFieldSubmit(){  document.getElementById("btnSubmit").disabled =true;  if (document.getElementById("drpProgramme").value != ""){  document.getElementById("btnSubmit").disabled =false;  }  } |
| **Expected Result** | Submit button will be turn into green colour (indicate as unlock). |
| **Outcome** |  |

##### Integration Testing

| **Unit Testing** | Auto generate programme after select university |
| --- | --- |
| **Test Case Objective** | Select the different university will generate the programme belong the university |
| **Test Case Description** | This testing to test the Programme will auto generate based on the university has been selected, result will be return from ajax. |
| **Input** | **Setup University**  University Code = “HELP”  University Desc = “HELP UNIVERSITY”  Email = “HELP@edu.my”  Contact Number = “1234567890”  **Setup Programme**  Programme Code = “BIMD”  Programme Desc = “Bachelor In Multimedia Design”  **Apply Programme**  University = “HELP UNIVERSITY” |
| **Expected Result** | **Setup University**  Popup message = “Record Has been Successful…”  **Setup Programme**  Popup message = “Record Has been Successful…”  **Apply Programme**  1)BACHELOR IN MULTIMEDIA DESIGN  2)BACHELOR IN MOBILE COMPUTING  3)BACHELOR IN INFORMATION TECHNOLOGY  4)BACHELOR IN GAME MULTIMEDIA  5)BACHELOR IN BUSINESS E-COMMERCE |

| **Outcome 1** | **Setup University** |
| --- | --- |

| **Outcome** | **Setup Programme**    **Apply Programme** |
| --- | --- |

##### System Testing

| **Unit Testing** | 1 |
| --- | --- |
| **Test Case Objective** | Apply Programme, and display application Enquiry |
| **Test Case Description** | This testing is to apply programme, and navigate to application enquiry. |
| **Input** | University = “SEGI COLLEGE”  Programme = “BACHELOR IN MOBILE COMPUTING”  ApplicantID = “B1401800”  Name=”kevin”  Email =[b1401800@gmail.com](mailto:b1401800@gmail.com)  ContactNumber=”0101401800” |
| **Expected Result** | Popup message : “Record has been successful” redirect : Application Enquiry |
| **Outcome** |  |

| **Unit Testing** | 2 |
| --- | --- |
| **Test Case Objective** | Check apply duplicate Programme |
| **Test Case Description** | This testing is to check applicant apply the same programme twice |
| **Input** | University = “SEGI COLLEGE”  Programme = “BACHELOR IN MOBILE COMPUTING”  ApplicantID = “B1401800”  Name=”kevin”  Email =[b1401800@gmail.com](mailto:b1401800@gmail.com)  ContactNumber=”0101401800” |
| **Expected Result** | Popup message : “Applicant has been apply once before.” Popup message : “Record has been failed, please try again.”  redirect : Application Enquiry |
| **Outcome** |  |

| **Unit Testing** | 3 |
| --- | --- |
| **Test Case Objective** | Applicant Enquiry |
| **Test Case Description** | This testing is to check application status of the applicant has been apply |
| **Input** | Applicant ID =”B1401800” |
| **Expected Result** | Display application result in a table format  (no | Date | University | Application No | Programme | Status) eg.  (1 | 12/03/2019 | HELP | HELP100001 | BACHELOR IN BUSINESS E-COMMERCE | A) (2 | 12/03/2019 | TARUC | TARUC100001| BACHELOR IN BUSINESS E-COMMERCE | P) |
| **Outcome** |  |

| **Unit Testing** | 4 |
| --- | --- |
| **Test Case Objective** | Applicant Enquiry View Feedback |
| **Test Case Description** | This testing is to check application s contain feedback, click view and display |
| **Input** | Applicant ID =”B1401800”  Onlick Button (View) |
| **Expected Result** | Popup Message = “Fake Information” |
| **Outcome** |  |

| **Unit Testing** | 5 |
| --- | --- |
| **Test Case Objective** | Applicant Qualification (Edit) |
| **Test Case Description** | This testing is to maintain or edit qualification information |
| **Input** | Applicant ID =”B1401800”  Qualification = “A LEVEL”  SubjectGrade = “AMM=50;MATHEMATIC=60;ASC=30;SC=50;” |
| **Expected Result** | Popup Message = “Record Has Been Successful…”  Redirect = Home Page |
| **Outcome** |  |

| **Unit Testing** | 6 |
| --- | --- |
| **Test Case Objective** | Setup Programme Code |
| **Test Case Description** | This testing is to Setup Programme Code for their own university |
| **Input** | Applicant ID =”HELP” (hidden)  Programme Code= “BIGM”  Programme Desc = “Bachelor In Game Multimedia” |
| **Expected Result** | Popup Message = “Record Has Been Successful…”  Redirect = Home Page |
| **Outcome** |  |

| **Unit Testing** | 7 |
| --- | --- |
| **Test Case Objective** | Review Application |
| **Test Case Description** | This testing is to Review the application that applicant has submit. |
| **Input** | University\_code =”HELP” |
| **Expected Result** | Display application result in a table format  (no | Date | Application No | Programme | Status | Feedback) eg.  (1 | 12/03/2019 | HELP100001 | BACHELOR IN BUSINESS E-COMMERCE | A)  (2 | 12/03/2019 | HELP100002 | BACHELOR IN INFORMATION TECHNOLOGY | C)  (3 | 02/04/2019 | HELP100003 | BACHELOR IN INFORMATION TECHNOLOGY | R)  (4 | 07/04/2019 | HELP100004 | BACHELOR IN MOBILE COMPUTING | A)  (5 | 07/04/2019 | HELP100005 | BACHELOR IN MOBILE COMPUTING | A) |
| **Outcome** |  |

| **Unit Testing** | 8 |
| --- | --- |
| **Test Case Objective** | Review Application (update feedback and status) |
| **Test Case Description** | This testing is to Review the application and update the feedback and status |
| **Input** | ApplicationNo =”HELP100005”  ApplicationStatus = “Approve”  AppicationFeedback= “You Have Been Approve” |
| **Expected Result** | Popup Message = “Record Has Been Successful…”  Redirect = Review Application |
| **Outcome** |  |

##### Installation Testing

We are running a Window Server 2012 R2 as the server to hosting the Web Application. At the same time we using the software to maintaining the Server, software (No-IP) for website hosting and software (Filezilla Server) to setting the upload and download speed, location, and access right to the user.

**Software infomation**

|  |  |  |
| --- | --- | --- |
| Software | Version | Description |
| Window Server 2012 R2 |  | This is the Operating System to hosting a server machine. |
| No-IP (DUC) | V4.1.1 | This is the free domain name hosting software |
| FileZilla Server | 0.9.53 Beta | This is to set the gateway to allow user only upload and download file from a folder that administrator has setting up. |
| FileZilla Client | 3.3532 | This is to let user login with the username and password to proceed the read and write access right. |
| Mysql Workbench | 8.0 | This is the database. |

**Setup FileZilla Server**

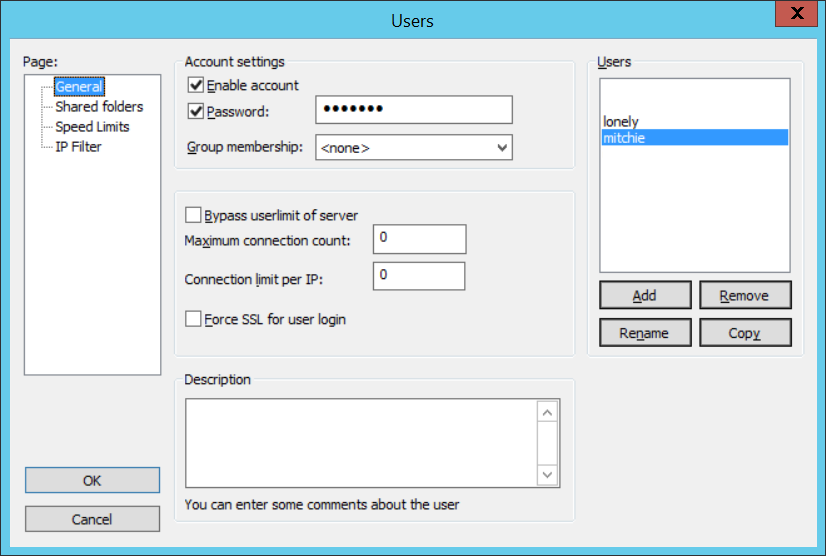


Diagram 1.0 Create a user and setting up the password

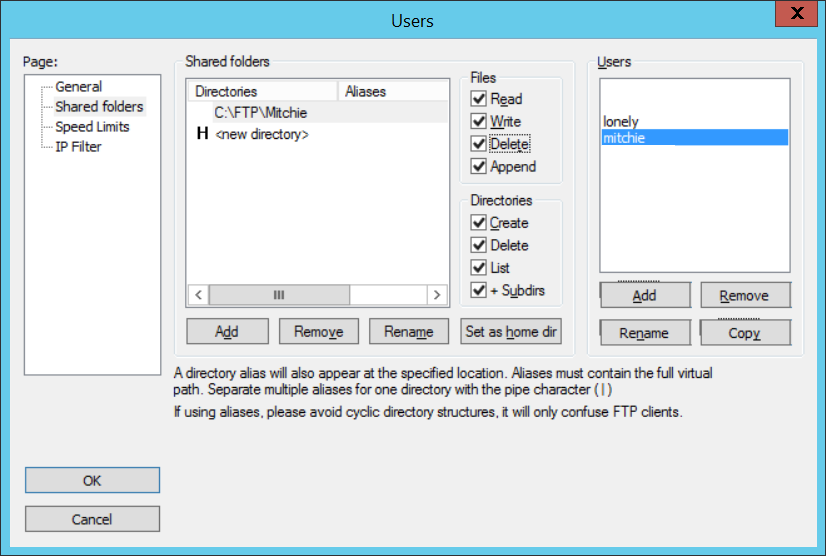


Diagram 1.1 Setup the directory for the user will be pointing location.



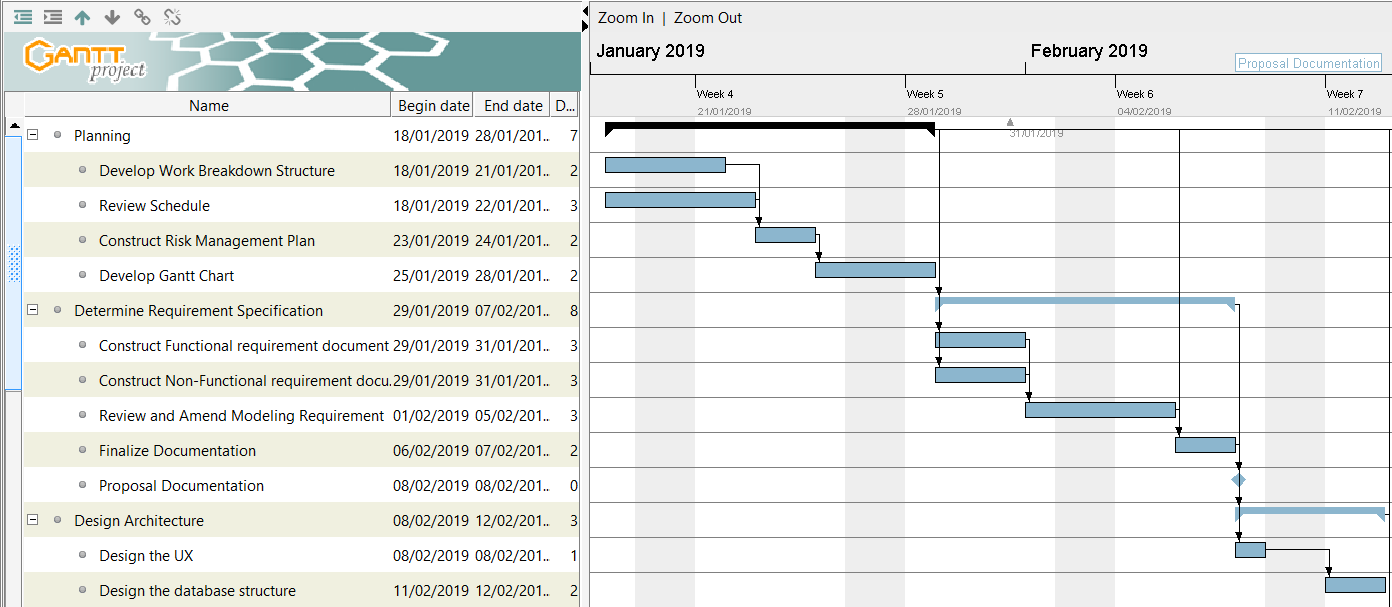
Diagram 1.2 Setup the transferring speed in constant speed limit

### Test Analysis Report

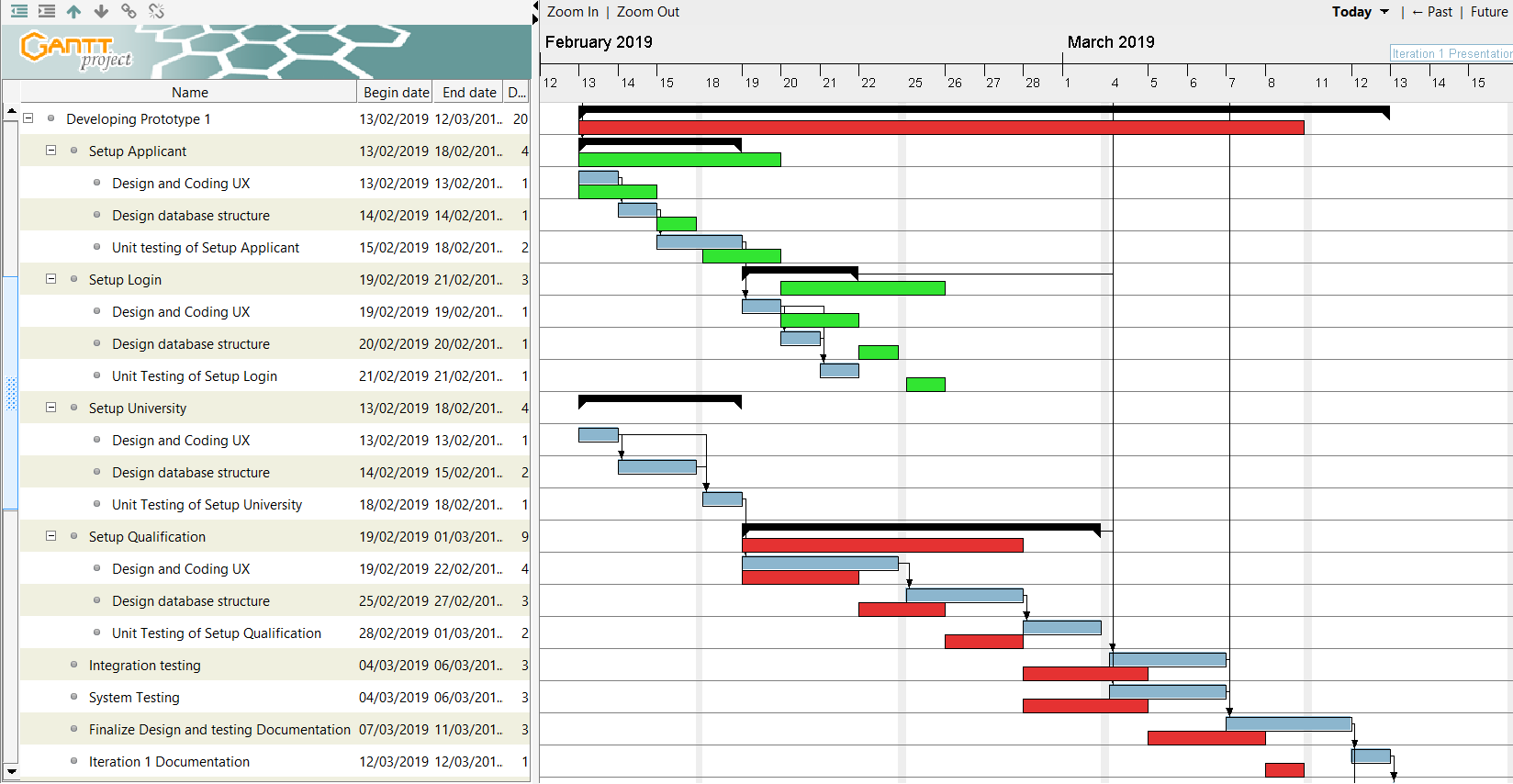
In this test plan, we have done the unit testing, integration testing, system testing, and installation testing. In Iteration 2, we have planned to finish the least use cases which are Record Programme, Review Application, Register Applicant, Apply Programme, Check Application Status and Record Qualification. Make sure every use case works fine and able to achieve the functional requirements. There are some non-functional requirement we haven’t achieve. Overall test results are good because the test results almost are same as the expected outcome. We have tested for the authentication for the input of applicant’s username and password in the register page, based on these to ensure that there are will not repeat the username and password. We test the login page for the applicant after their registered to make sure that they can login successfully. We tested the validity of the applicant login page to authenticate the username is existing in the database. After the user confirms their username and password, MK System will say “Failed Login, Please Try Again!” which in the situation if the username and password are incorrect with the register one. If not, it will go to the specific user’s page. We have tested for applicants will see the applicant’s navigation and drop-down menu which is according to the applicant’s permission. In the page of record programme which belongs to university admin, we have tested for adding the data into the database and auto-generated the Date/UID which input by university admin. When university admin wants to edit the programme, the new line will be replaced with auto-generated new Date/UID to be stored in the database. We also tested for the programme list, used the “search” button to see the programme list to confirm the programme data in the database. In the page of review application, we have tested the after applicant’s application, the application data will store into the database, and university admin to review the application and check it to “successfully” or “unsuccessfully”, the checkbox data will be stored into the database. In the page of the apply programme which belongs to the applicant, we have tested for submission of the application when applicant input data of apply programme, those data will save into the database and auto-generated the application ID. In the page of check application status, we have tested the application data is stored in the database, and show the application enquiry into the table. In the page of record qualification, we have tested for the submission of editing the qualification, when applicant clicks the “edit” button and input the data of applicant qualification, those data will be saved and updated into the database.

### Update Gantt Chart

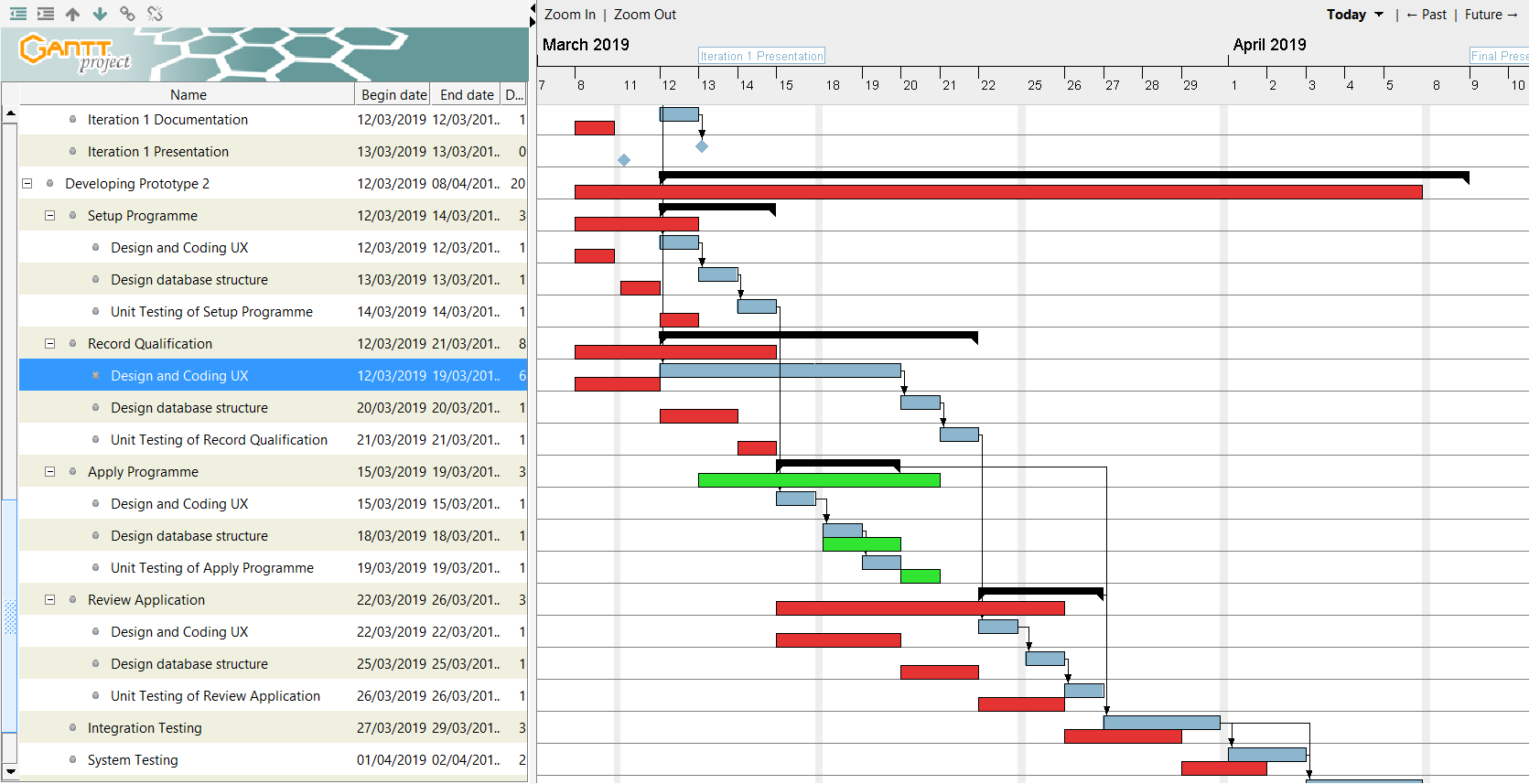
Gantt Chart (Part 1)



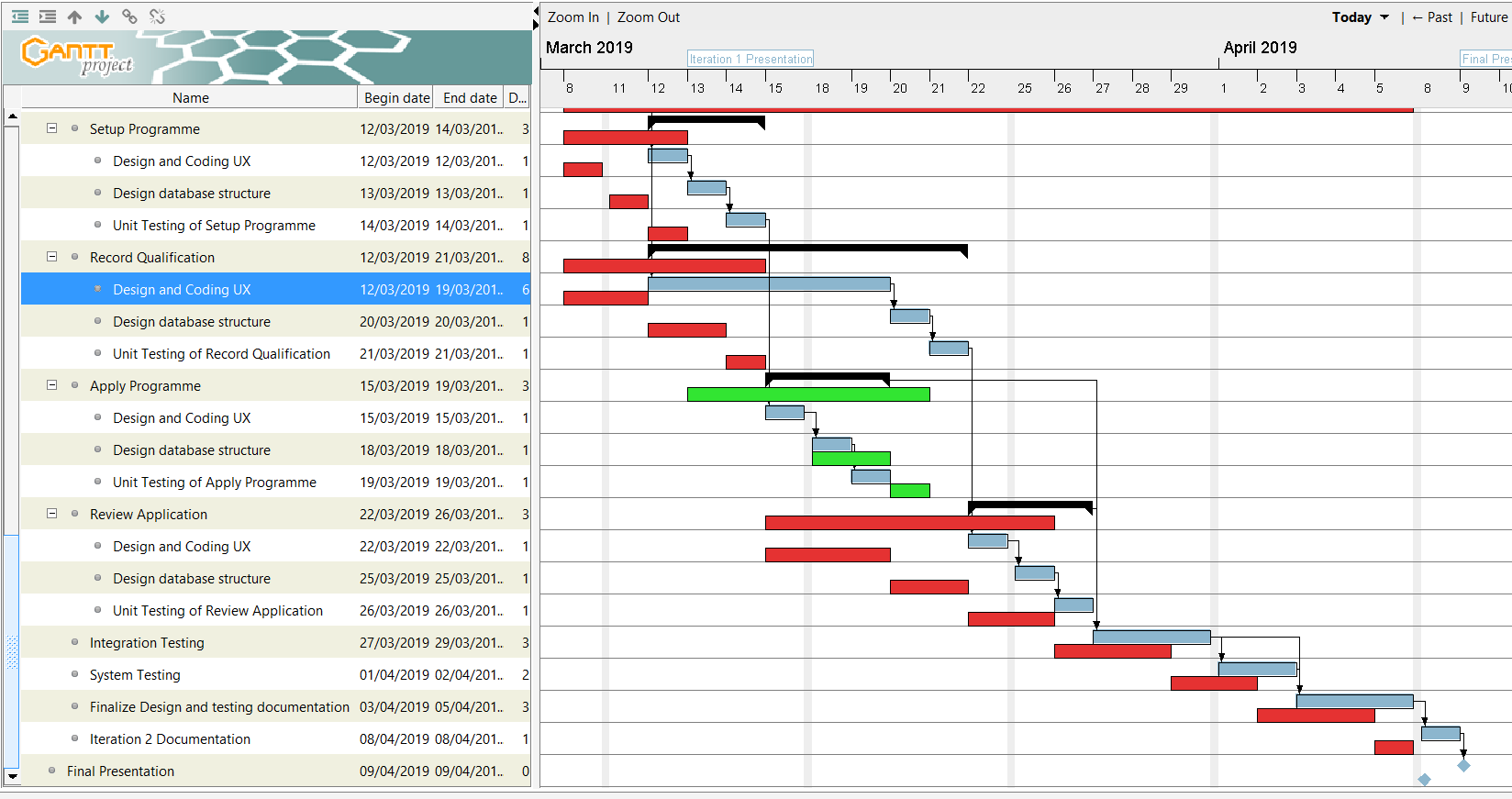
Gantt Chart (Part 2)



Gantt Chart (Part 3)



Gantt Chart (Part 4)



### Conclusion

**Did the group meet the objectives defined in Assignment 1?**

After several weeks of work, we finally developed a website system named “MK System” to meet our objectives in the project proposal. But we just achieved the part of the objective which is to produce a simple and easy-to-use user interface design to facilitate user to query information, online applications, record programme, review application status and other activities. We haven’t achieved that the objective of suggesting the relevant programmes and details information when the applicant is applying for a programme by entering search criteria, based on the most effective search methods. The “MK System” work effectively that enable the applicant to get programme qualification information easily and make it convenient for students to the online application. It made a big convenience between the university and applicant student to apply the programme online, it’s saving a lot of time to search online. Moreover, “MK System” offered a more efficient way to get the application status, let the applicants enable to get the application information as soon as possible. Furthermore, “MK System” make the university admin easier to review the application to improve their work efficiently.

**What went wrong?**

There are few things we went wrong, one of it is that based on the iteration 1 discussion during the presentation, we found that the qualification is too complicated and confusing on the particular parts such as which qualification having different by grading and scoring. In this part as first were designing only accept the grading only. Next thing is that when we created the primary or unique running number they were not auto-generated, they were created using the “unique code” by ourselves. Another thing is that due to we were not very familiar with the testing methods and misunderstand the qualification process, it caused some development errors.

**What went right?**

Things that went right is that we testing all the webpage twice the time and we planned the operation of the system from the user's standpoint, based on the user’s view to try to think which stage will have a better user experience. The testing results can almost match our expected results, and we added the “Check Application Status” use cases, it will be very helpful for applicants to check the application status. We discussed and resolved the problems together when each of us got difficulties during the system developing and report processing. We helped each other to debug the system and gave suggestions for each other to develop the system.

**What you would have done differently?**

We would more likely to make the “MK System” with a more simpler and easier to use way for students to apply for their dreaming university and ideal programme online. We hope that the website system will able to help them to get useful information and make their application become more efficient. We learned that instead of using auto-generate primary code, we prefer to use the user input record the data as a primary key currently. Furthermore, we will provide simple page navigation and prompt a pop-up message to let the user know the system running processes.