Table of Contents

[**DEFINITION OF THE PROBLEM** 2](#_Toc25663982)

[**CUSTOMER REQUIREMENT SPECIFICATION DOCUMENT** 3](#_Toc25663983)

[**CUSTOMER’S ACCEPTANCE CRITERIA** 5](#_Toc25663984)

[**PROJECT PLAN** 6](#_Toc25663985)

[**DOCUMENT DESIGN** 9](#_Toc25663986)

[**INTERFACE DESIGN DOCUMENT** 10](#_Toc25663987)

[**PROCESS DESCRIPTION DOCUMENT** 20](#_Toc25663988)

[**CODING STANDARDS DOCUMENT** 37](#_Toc25663989)

[**TASK SHEET** 39](#_Toc25663990)

[**INTEGRATION TESTING REPORT** 41](#_Toc25663991)

[**REVIEW REPORT** 42](#_Toc25663992)

[**FINAL CHECKLIST** 43](#_Toc25663993)

[**USER GUIDE** 44](#_Toc25663994)

**Module**

# **DEFINITION OF THE PROBLEM**

**Problem definition of CERTIFICATE GENERATION SYSTEM Soft Application:**

The Certificate Department works for the "ABC" institute to developing the certification system for the students who have completed the exam. For each student who go to the “ABC” institute for register.

After that the student had registered, they must take an exam for each subject. The role of the exam center will be producing the certificate for the students.

It should also check whether the students have given the exam, if he/she has not given one exam, the certificate for the particular should not get printed instead it should generate the appropriate message to the students for that.

This mean the students have one year after he/she given one exam date, If the exam of the students has been failed or he/she have a special case and must take pause, they will be to take one more exam. The center exam will message the students which extend the date of the exam. If that times the exam is not successful, he/she needs to return registry for that certificate.

This form should be designed in such a way that it should contain name of the institute, name of the student, name of the certificate and exam date.

However, they also wish that when he/she have issued certificate for the students. So, when he/she get a job or apply to a company. Because of the concern that students are granted a certificate at this institute, that company may be look up the information of her/him certificate when it was contact to institute.

The Certificate management system application will be provided the functions:

* User can search Class view description of a selected Class, and Class, update Class and delete Class.
* User will be able to search and generate report of Certificate Type, Student, and Certificate.
* User will be register for the student.

# **CUSTOMER REQUIREMENT SPECIFICATION DOCUMENT**

**Client/ Project Undertaken: Certificate Generation System**

1. **List of inputs to the system**

STUDENT (**StudentID**, StudentName, Address, DOB, Phone, Email, IdentityCard, Gender)

CERTIFICATE (**CertificateID**, CertificateName, CertificateNum)

STUDENT\_CER (**StudentID, CertificateID**, **Exam\_Date,** ID\_Num, Avg\_Mark, Result)

DETAIL\_CER (**SubjectID**, SubjectName, ***CertificateID***)

EXAMMARK (**ID\_Num, SubjectID**, Mark)

CER\_MANAGER (**No\_ID**, ID\_Num, SubjectID)

USER (**UserID**, Fullname, Username Password, Address, DOB, Phone, Email, Gender,Roles).

COURSES (**Exam\_Date**, Course\_Name)

1. **List of outputs expected from the system**

* It should contain fields like (Student, Certificate and Result) will be display in Certificate for the students.
* After testing the student’s detail of each certificate. Once the marks of the students is evaluated then the entry of the marks is done in this application system. Exam Center will be entering mark for detail of each certificate then the status of the pass or fail will be display (exam course, mark, detail certificate).

1. **Overview of processes involved in the system**

* Check registry input from student and login, logout user.
* Get the students selected by admin add to form registered.
* Get detail of each certificate selected by the students add to form result of exam.
* Calculate the mark of each subject for the students and check exam of the student.
* Check valid of the certificate for the students.
* User must create to list of the student who was completed or failed the exam.
* User must create to list of the student who need to reserve that exam course after test date for within a year.
* Check domain, reference of database.
* HELP *(It should display the short explanation of the project in order to guide the end user).*

1. **Hardware and software required for implementing the project**

* **Hardware Requirements:**
  1. **Client Hardware**:
     + CPU: Intel Core i3-2130U, 2.5GHz
     + RAM: 2GB or higher
     + HDD: 80GB Hard Disk space or higher
* **Software Requirements:**
  1. Client **Software**:
     + Windows 7 (or higher)
     + Microsoft SQL Server 2012 (or higher).
     + JDBC, JDBC Driver, JDK 1.8 (or higher).
     + Netbeans IDE 8.2 (or higher)

# **CUSTOMER’S ACCEPTANCE CRITERIA**

|  |  |
| --- | --- |
| **S. No** | **Customer’s Acceptance Criteria** |
| **1.** | User Login, logout check valid of form |
| **2.** | Enter the mark detail of each certificate from form Marks |
| **3.** | Create, update, delete and edit a registry form for the students, users |
| **4.** | Design, create, update, modify, security and maintain database system |
| **5.** | Produce the certificate for the students |
| **6.** | It should generate the appropriate message to the students for the particular should not get certificate for the students. |
| **7.** | Here the students result is to be displayed. |
| **8.** | Valid of certificate for the students. |
| **9.** | Database must back up, update, upgrade regularly. |

# **PROJECT PLAN**

1. **Project Details**
   * **Name of the Client:** *Certificate Generation System*
   * **Date of Project Plan:** October 9th, 2019
   * **Project Vision/ Objectives:**

**Project Vision***: The Certificate Department works for the "ABC" institute to developing the certification system for the students who have completed the exam.*

**Project Objectives:**

* + The solutions to overcome difficult problem of current. They need to increase the efficiency and accuracy.
  + Talking pressure off staff who have too much paperwork.
  + Ensuring better communication between staff and Certificate department.
  + Reducing boring and repetitive task.
  + Reducing loss of data.
  + **Scope:**

We will develop and implement certificate maps for the Certificate Department works for some center exam. Further, we will analyze and design user admin support processes and areas that increase students’ ability to successfully navigate their way through their educational experience. We will be finished this project about November 2019.

* + **Our understanding of the client organization:**

To overcome the above problem the organization had finally decided to computerize its certificate department. Organization wants following things to be implemented in the system. Here should be two logins.

* + - Admin registry for the student, and management system about certificate, subject and courses of exam and result of the students pass or fail.
    - User entry mark of the student who have registered.
  + **Project Organization with Responsibilities and Authorities:**

|  |  |
| --- | --- |
| **Members** | **Role and Responsibilities** |
| Nghi Dang Quang | **Leader/ Analyst/ Designer/ Developer/ Tester/ Implementation** |
| Khanh Nguyen Le | **Analyst/ Implementation / Developer/ Tester** |
| Thang To Toan | **Analyst/ Designer/ Developer/ Tester** |
| Thanh Nguyen Phuoc | **Analyst/ Implementation** / **Developer/ Tester** |

1. **Project Initiation/ Requirement Documents**

Generate information about Certificate for the students.

[1] SQL for smarties advanced SQL programming <https://epdf.pub/joe-celkos-sql-for-smarties-advanced-sql-programming7145.html>

[2] Data Management Using Microsoft SQL Server – Aptech Limited.

[3] A Programmer's Guide to Java SCJP Certification - A Comprehensive Primer, 3rd edition (Addison-We).

[4] Lab Java Programming with JDBC SQL Server – HUTECH.

[5] How to Use JDateChooser or JCalendar in Java - <https://www.youtube.com/watch?v=C0ZZpkliwVk>

[6] Java Code Examples for org.apache.poi.xssf.usermodel.XSSFSheet - <https://www.programcreek.com/java-api-examples/?api=org.apache.poi.xssf.usermodel.XSSFSheet>

[7] Java and DPF with iText - <https://www.vogella.com/tutorials/JavaPDF/article.html?fbclid=IwAR1bWzE3oXl0YmAxgghObKri0wSi1nTMq50U_Qq6bb225IYhNQEb84VanqE>

[8] Java and Apache - <https://www.apache.org/dyn/closer.lua/poi/release/bin/poi-bin-4.1.1-20191023.tar.gz> - <https://www.youtube.com/watch?v=RyvrpTelJ4w>

1. **Deliverables**

* Drive google store about Documents and Source code: <https://drive.google.com/open?id=1d15liqbwws5qoIdXTuiuqLT6ouRZ73fy>
* All files \*.exe, \*.jar, \*.sql, \*.mdf, \*.ldf

1. **Project Dependencies**

* Depend on Data input
* Depend on System performance
* Depend on User
* Depend on Security Procedures

1. **Major Project Milestones**

* Keep Satisfied
* Keep Interested
* Data entry Staff Receptionists
* Manage closely

1. **Quality Plan**
   * **Review Activities:** We will be meet participants about at least twice a week, each meeting there is a check for each member in group to review processing, help to solve and trouble
   * **Testing Activities**: Test spell design document, test case design DFD and database, Function, Coding and Demo.
   * **Backup and recovery strategies:** Save files on device storage (USB, HDD) and use GitHub to synchronize all document relative of project. Each member store a copy of project in their own computer and one in their mail drive.

# **DOCUMENT DESIGN**

|  |  |
| --- | --- |
| **PROPERTIY** | **VALUE** |
| Document theme and color scheme | None |
| Form-Background color | Light Gray |
| Title-Font Size | 16 |
| Title-Font Color | Black |
| Title-Font Style | Times New Roman |
| Title-Alignment | Center |
| Background color of Controls on the form | White |
| Foreground color of Controls on the form | Black |
| Control Caption –  Font Size | 16 |
| Control Caption –  Font Color | Black |
| Control Caption –  Font Style | Times New Roman |
| Control caption and controls-Alignment | Left |
| Command button-Alignment | Center |

# **INTERFACE DESIGN DOCUMENT**

1. **List of form to be created**

|  |  |  |
| --- | --- | --- |
| **Document Name** | **Description** | **Controls on the Document** |
| **frLogSQL** | The interface used to connect database |  |
| **frUserlogin** | The interface used to log in |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **frManager** | The interface used to manage main | **A screenshot of a cell phone  Description automatically generated** |
| **frRegister** | The interface used to manage students register |  |
| **frMarkEntry** | The interface used to manage mark | **A screenshot of a cell phone  Description automatically generated** |

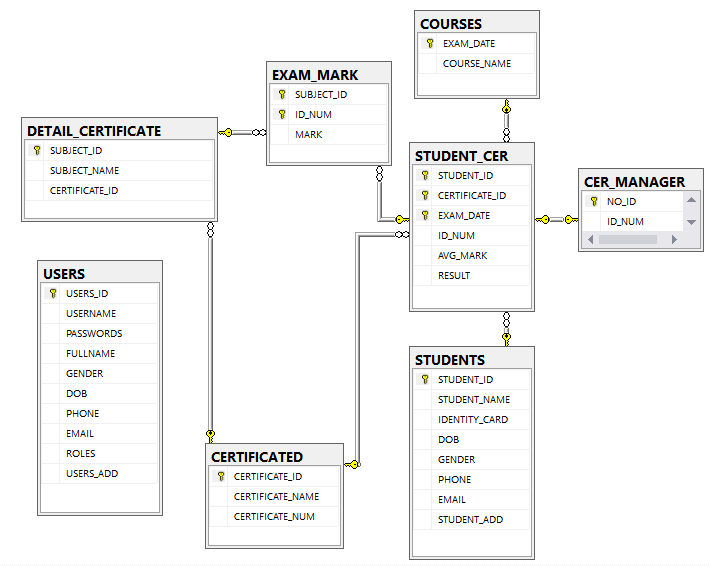
|  |  |  |
| --- | --- | --- |
| **frCerManager** | The interface used to manage certificate store | **A screenshot of a cell phone  Description automatically generated** |
| **frMCertificate** | The interface used to manage certificate | **A screenshot of a cell phone  Description automatically generated** |
| **frMStudent** | The interface used to manage students | **A screenshot of a cell phone  Description automatically generated** |

|  |  |  |
| --- | --- | --- |
| **frMSubject** | The interface used to manage subject | **A screenshot of a social media post  Description automatically generated** |
| **frMReport** | The interface used to report | **A screenshot of a cell phone  Description automatically generated** |
| **frMPrint** | The interface used to manage print certificate | A screenshot of a cell phone  Description automatically generated |
| **frMStudentPass** | The interface used to manage students pass exam | A screenshot of a cell phone  Description automatically generated |
| **frMStudentPass** | The interface used to manage students fail exam | A screenshot of a cell phone  Description automatically generated |
| **fmCourse** | The interface used to manage course exam | A screenshot of a cell phone  Description automatically generated |

1. **Database Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name of the Table** | **Table Description** | **Number of Fields** | **Primary Key** | **Related Tables** | **Foreign Key** |
| Student |  | 8 | **StudentID** | Detail\_Certificate, Exammark, Cer\_manager |  |
| Certificate |  | 3 | **CertificateID** | Student\_Cer, Detail\_Certificate, Exammark |  |
| Student\_Cer |  | 6 | **StudentID,**  **ExamDate CertificateID** | Student, Certificate, Cer\_manager, Exammark | ***StudentID,***  ***CertificateID***  ***ExamDate*** |
| Detail\_Certificate |  | 3 | **SubjectID** | Certificate,  Examresult, Exammark | ***CertificateID*** |
| ExamMark |  | 5 | **ID\_Num, SubjectID** | Student\_Cer,  Detail\_Certificate | ***ID\_Num, SubjectID*** |
| Cer\_manager |  | 3 | **No\_ID** | Student\_Cer, Student |  |
| Users |  | 10 | **UserID** |  |  |
| Courses |  | 2 | **Exam\_Date** | Student\_Cer |  |

1. **Algorithms and data structures**



**Figure 2.1 Database Diagrams**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field name** | **Data type** | **Key** | **Constraint** | **description** |
| **Table USERS** | | | | |
| **USERS\_ID** | **varchar(20)** | **Primary key** | **not null** |  |
| **USERNAME** | **nvarchar(40)** |  | **not null** |  |
| **PASSWORDS** | **nvarchar(40)** |  | **not null** |  |
| **FULLNAME** | **nvarchar(100)** |  | **not null** |  |
| **GENDER** | **nvarchar(10)** |  | **not null** |  |
| **DOB** | **Date** |  | **not null** |  |
| **PHONE** | **varchar(20)** |  | **not null, UNIQUE** |  |
| **EMAIL** | **nvarchar(40)** |  | **not null** |  |
| **ROLSE** | **nvarchar(40)** |  | **not null** |  |
| **USERS\_ADD** | **nvarchar(100)** |  | **not null** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field name** | **Data type** | **Key** | **Constraint** | **description** |
| **Table STUDENTS** | | | | |
| **STUDENT\_ID** | **varchar(20)** | **Primary key** | **not null** |  |
| **STUDENT\_NAME** | **nvarchar(100)** |  | **not null** |  |
| **GENDER** | **nvarchar(10)** |  | **not null** |  |
| **DOB** | **Date** |  | **not null** |  |
| **PHONE** | **varchar(20)** |  | **not null, UNIQUE** |  |
| **EMAIL** | **nvarchar(20)** |  | **not null** |  |
| **INDENTITY\_CARD** | **varchar(20)** |  | **not null, UNIQUE** |  |
| **STUDENT\_ADD** | **nvarchar(100)** |  | **not null** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field name** | **Data type** | **Key** | **Constraint** | **description** |
| **Table COURSES** | | | | |
| **EXAM\_DATE** | **Date** | **Primary key** | **not null** |  |
| **COURSE\_NAME** | **nvarchar(100)** |  | **not null** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field name** | **Data type** | **Key** | **Constraint** | **description** |
| **Table CERTIFICATED** | | | | |
| **CERTIFICATE\_ID** | **varchar(20)** | **Primary key** | **not null** |  |
| **CERTIFICATE\_NAME** | **nvarchar(100)** |  | **not null** |  |
| **CERTIFICATE\_NUM** | **nvarchar(20)** |  | **not null** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field name** | **Data type** | **Key** | **Constraint** | **description** |
| **Table STUDENT\_CER** | | | | |
| **STUDENT\_ID** | **nvarchar(20)** | **Primary key** | **not null, foreign key, on delete, update cascade** |  |
| **CERTIFICATE\_ID** | **nvarchar(20)** | **Primary key** | **not null, foreign key, on delete, update cascade** |  |
| **EXAM\_DATE** | **Date** | **Primary key** | **not null, foreign key, on delete, update cascade** |  |
| **ID\_NUM** | **nvarchar(20)** |  | **not null, UNIQUE** |  |
| **AVG\_MARK** | **float** |  | **null** |  |
| **RESULT** | **nvarchar(20)** |  | **null** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field name** | **Data type** | **Key** | **Constraint** | **description** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table CER\_MANAGER** | | | | |
| **NO\_ID** | **nvarchar(20)** | **Primary key** | **not null** |  |
| **ID\_NUM** | **nvarchar(20)** |  | **not null, UNIQUE, foreign key, on delete, update no action** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field name** | **Data type** | **Key** | **Constraint** | **description** |
| **Table DETAIL\_CERTIFICATE** | | | | |
| **SUBJECT\_ID** | **varchar(20)** | **Primary key** | **not null** |  |
| **SUBJECT\_NAME** | **nvarchar(100)** |  | **not null** |  |
| **CERTIFICATE\_ID** | **nvarchar(20)** |  | **not null, foreign key, on delete, update cascade** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field name** | **Data type** | **Key** | **Constraint** | **description** |
| **Table EXAM\_MARK** | | | | |
| **SUBJECT\_ID** | **nvarchar(20)** | **Primary key** | **not null, foreign key, on delete, update cascade** |  |
| **ID\_NUM** | **nvarchar(20)** | **Primary key** | **not null, foreign key, on delete, update cascade** |  |
| **MARK** | **float** |  | **null** |  |

# **PROCESS DESCRIPTION DOCUMENT**

1. **ER Diagram**

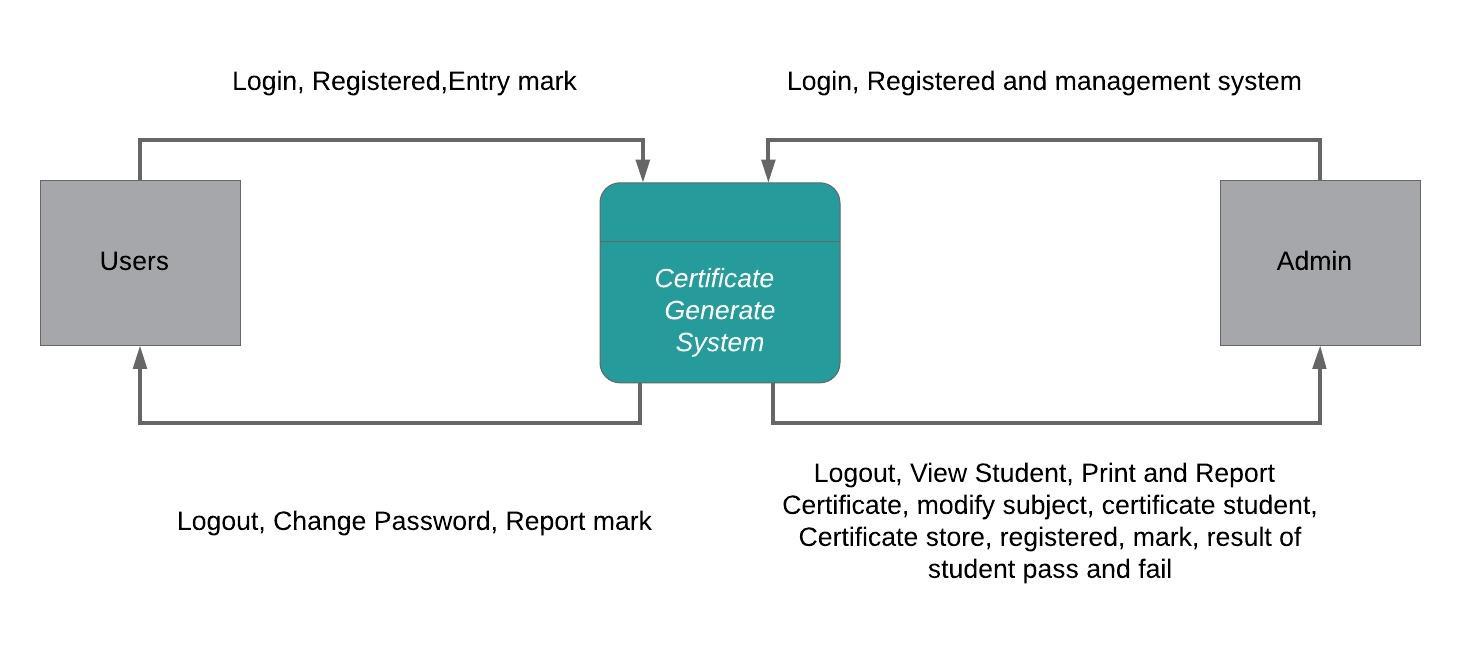
**A close up of a map

Description automatically generated**

**Figure FormNo7.1: ERD**

1. **Data Flow Diagram Context.**

**Process Name: DFD context.**

****

**Figure FormNo7.2: DFD Level 1**

**Process Description and Details:**

1. **DFD System:**

**A close up of a map

Description automatically generated**

**Figure FormNo7.2: DFD Level 1**

1. **DFD process 1**

**A close up of a map

Description automatically generated**

**Figure FormNo7.2: DFD Level 1.1**

1. **DFD process 5**

**A screenshot of a cell phone

Description automatically generated**

**Figure FormNo7.2: DFD Level 1.5**

1. **DFD process 8**

**A screenshot of a cell phone

Description automatically generated**

**Figure FormNo7.2: DFD Level 1.8**

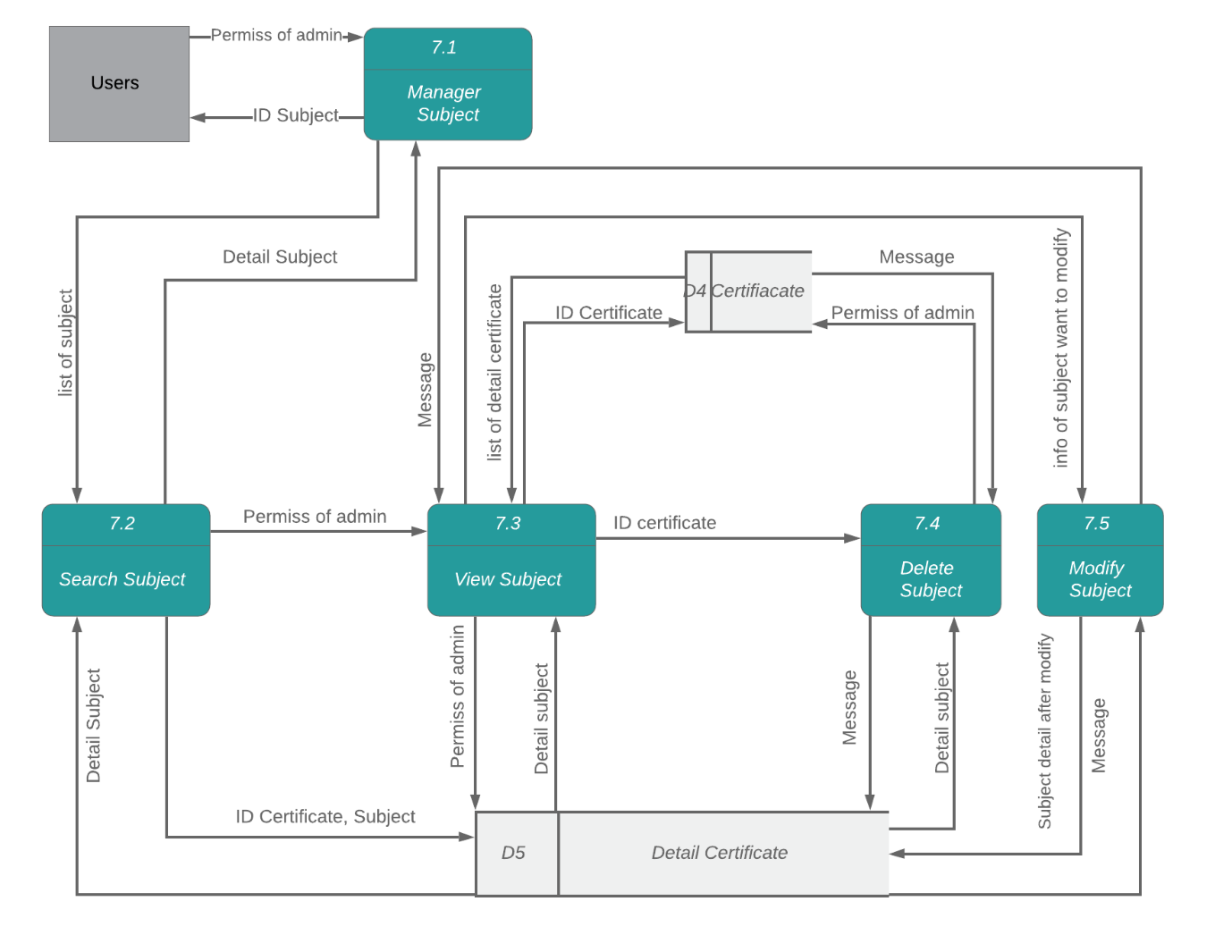
1. **DFD process 6**

**A screenshot of a cell phone

Description automatically generated**

**Figure FormNo7.2: DFD Level 1.6**

1. **DFD process 7**

****

**Figure FormNo7.2: DFD Level 1.7**

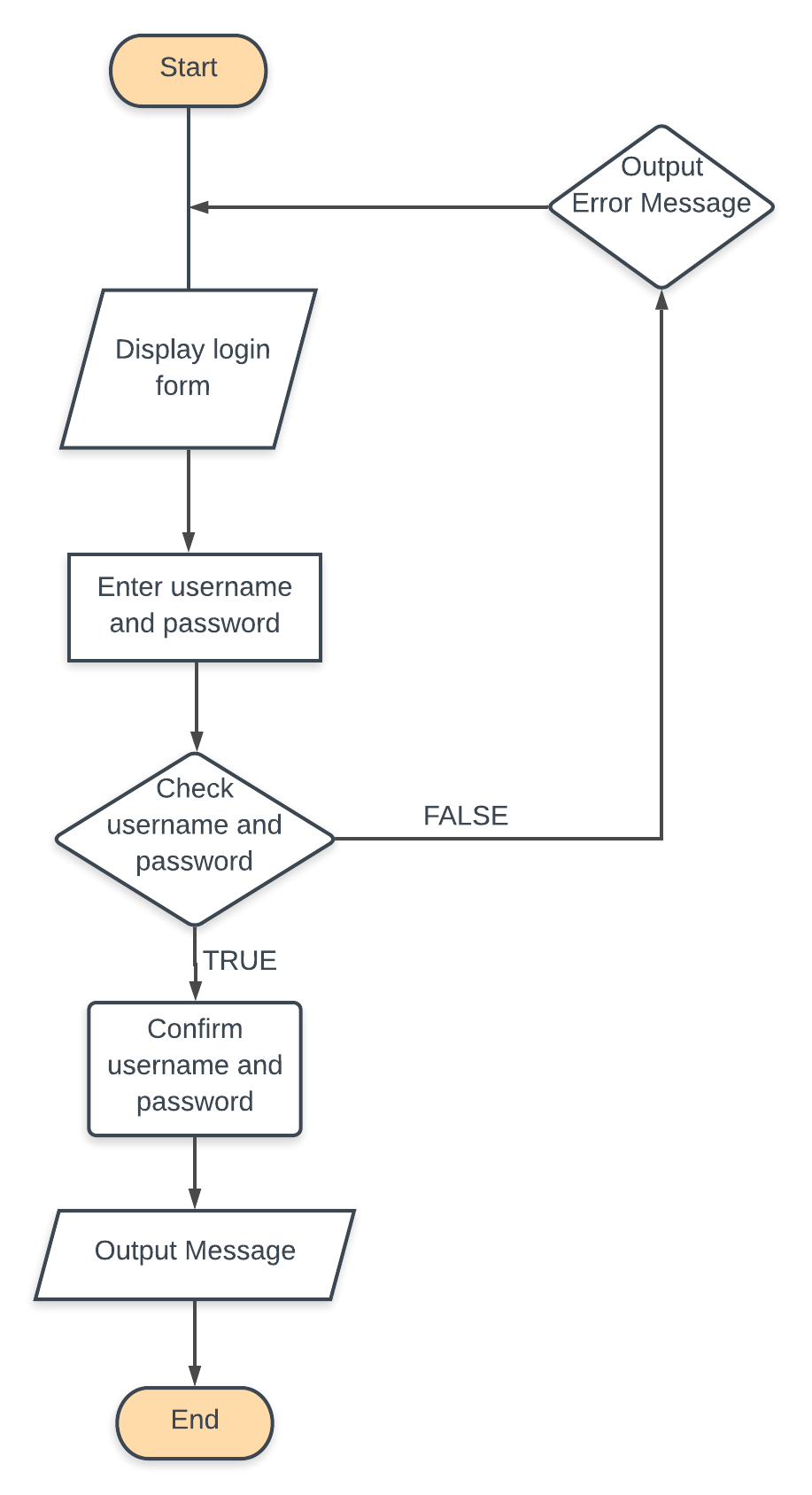
1. **A picture containing screenshot

   Description automatically generatedDFD process 10**

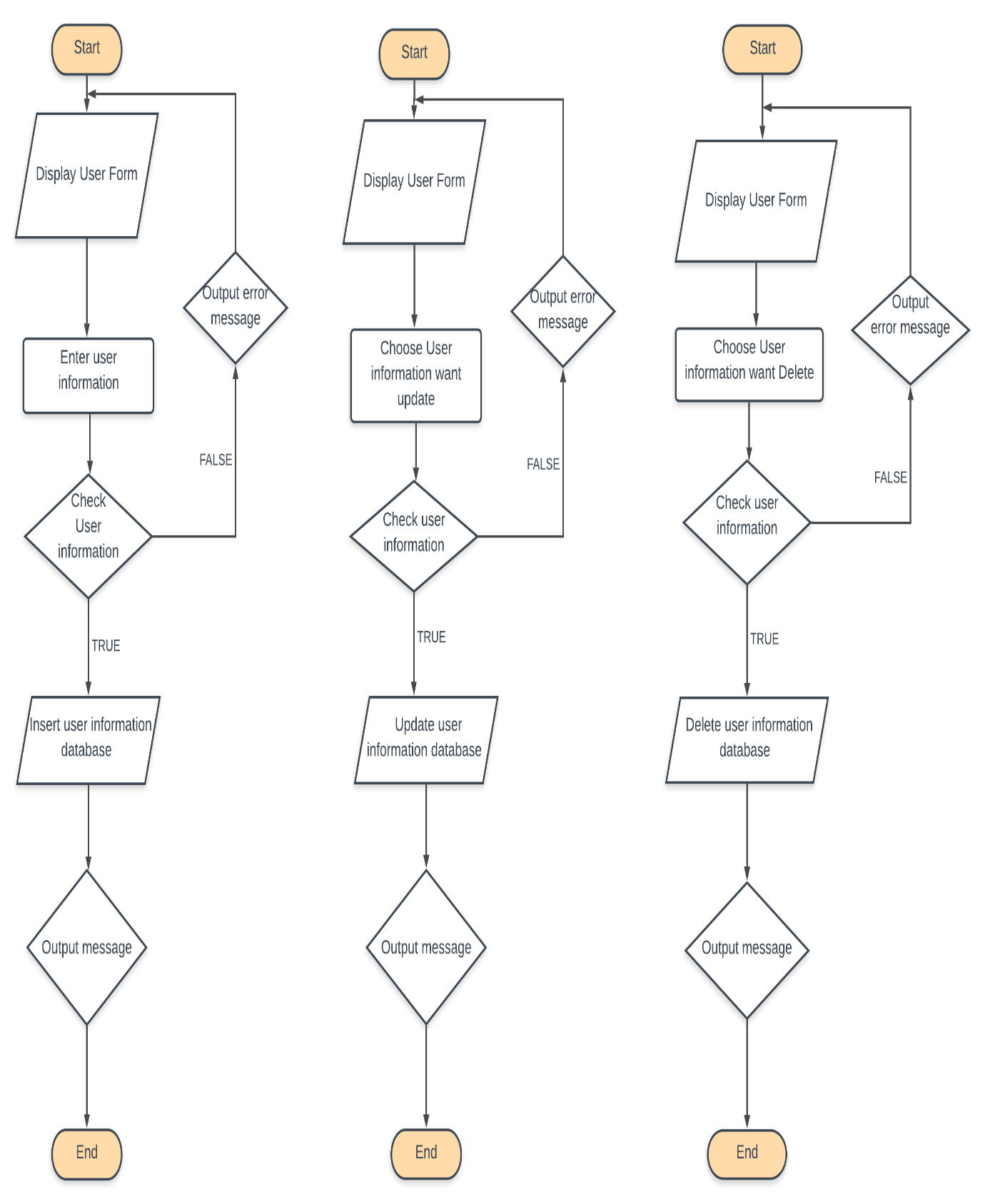
**Figure FormNo7.2: DFD Level 1.10**

1. **Process Name Flow Chart:**

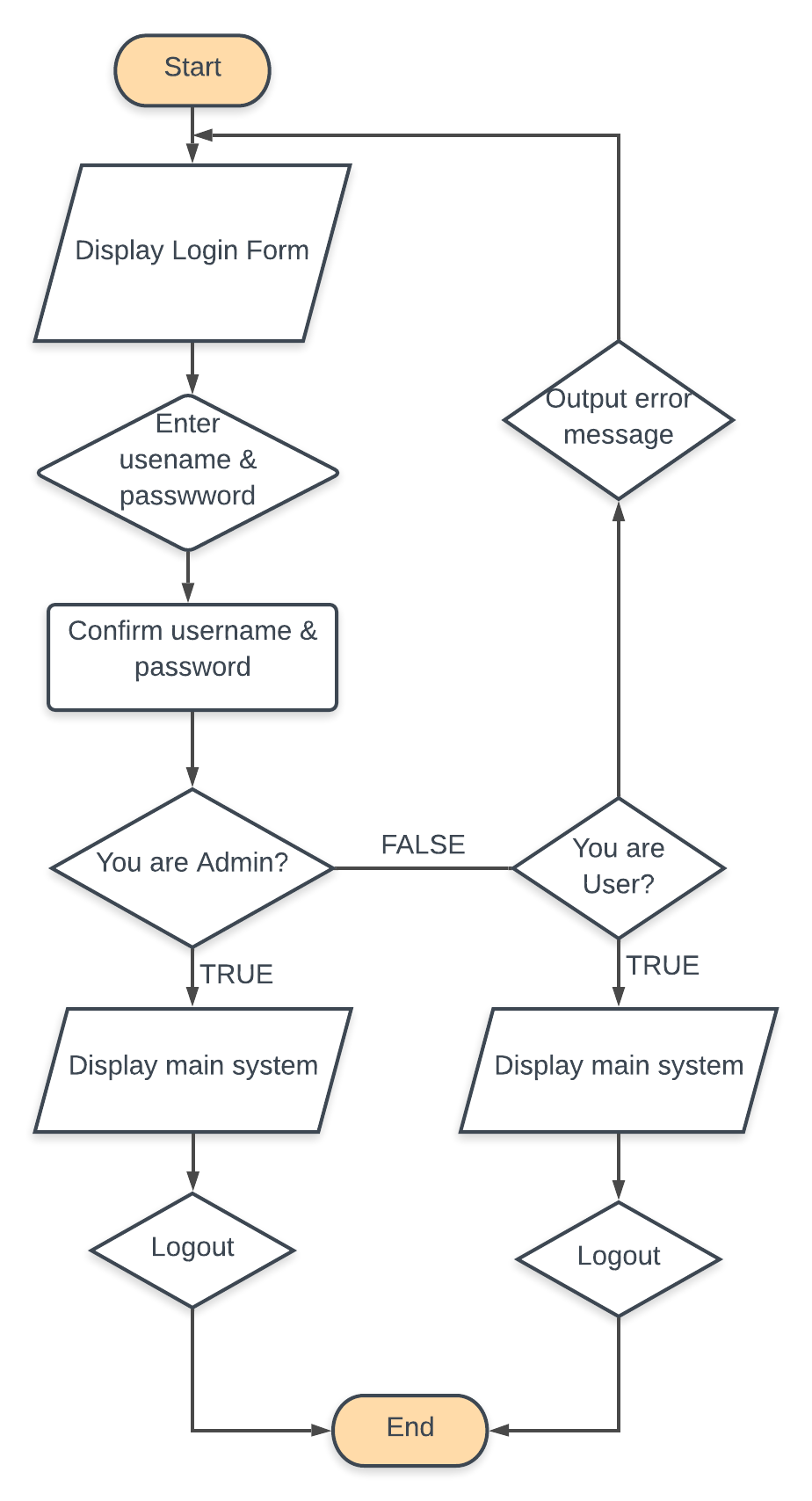
**Process Description and Details:**

1. **Login System. **

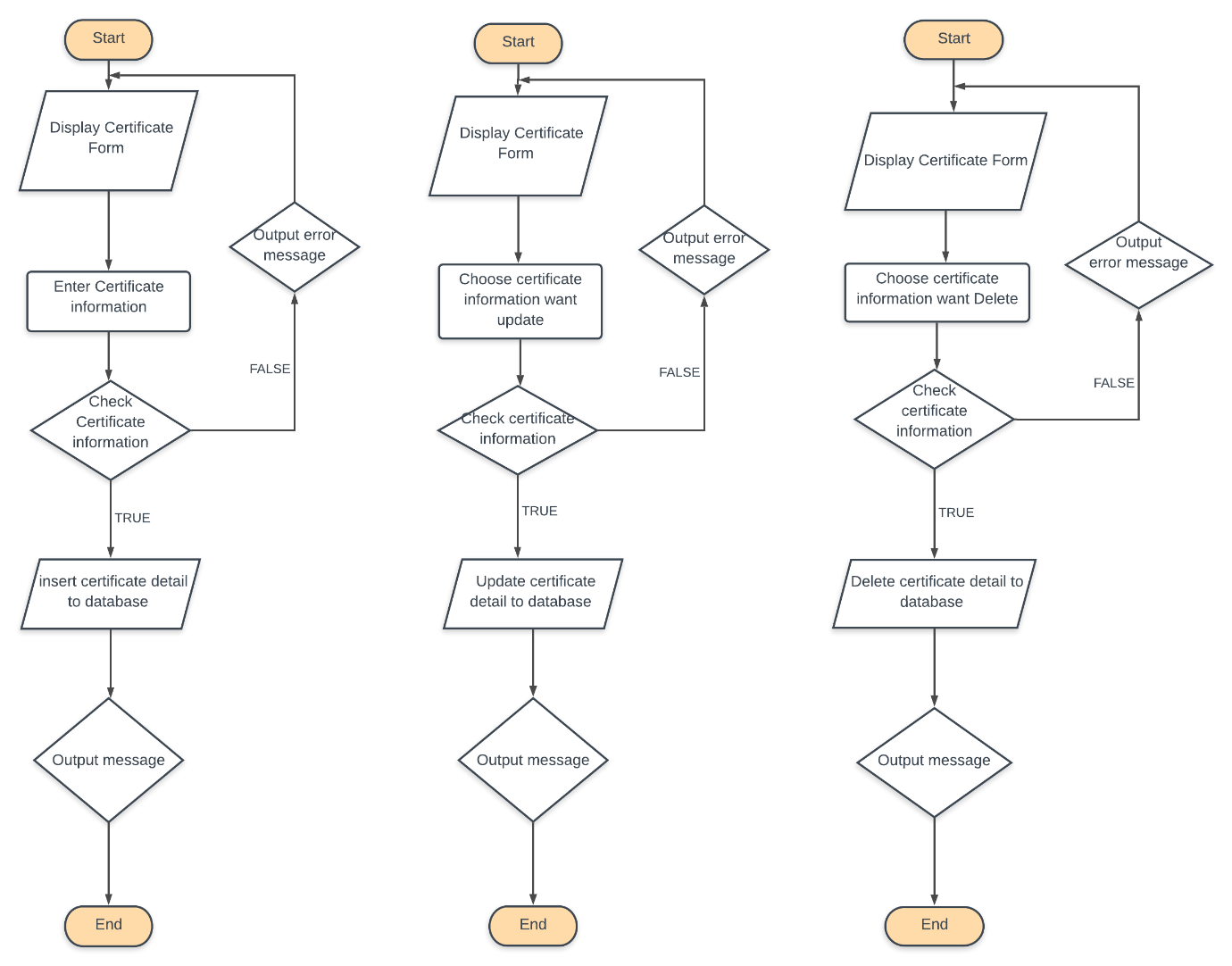
1. **User management**

****

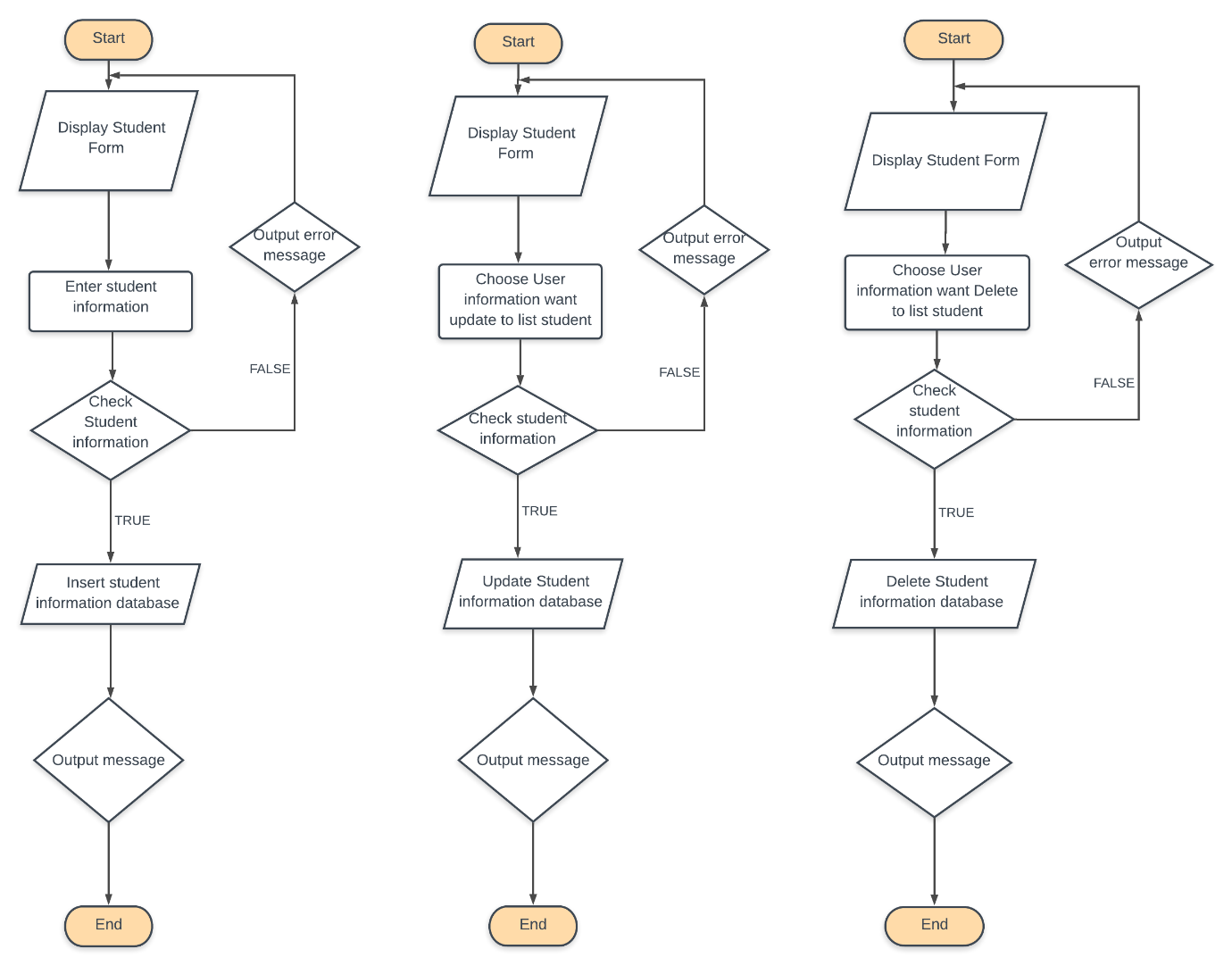
1. **Permits management**

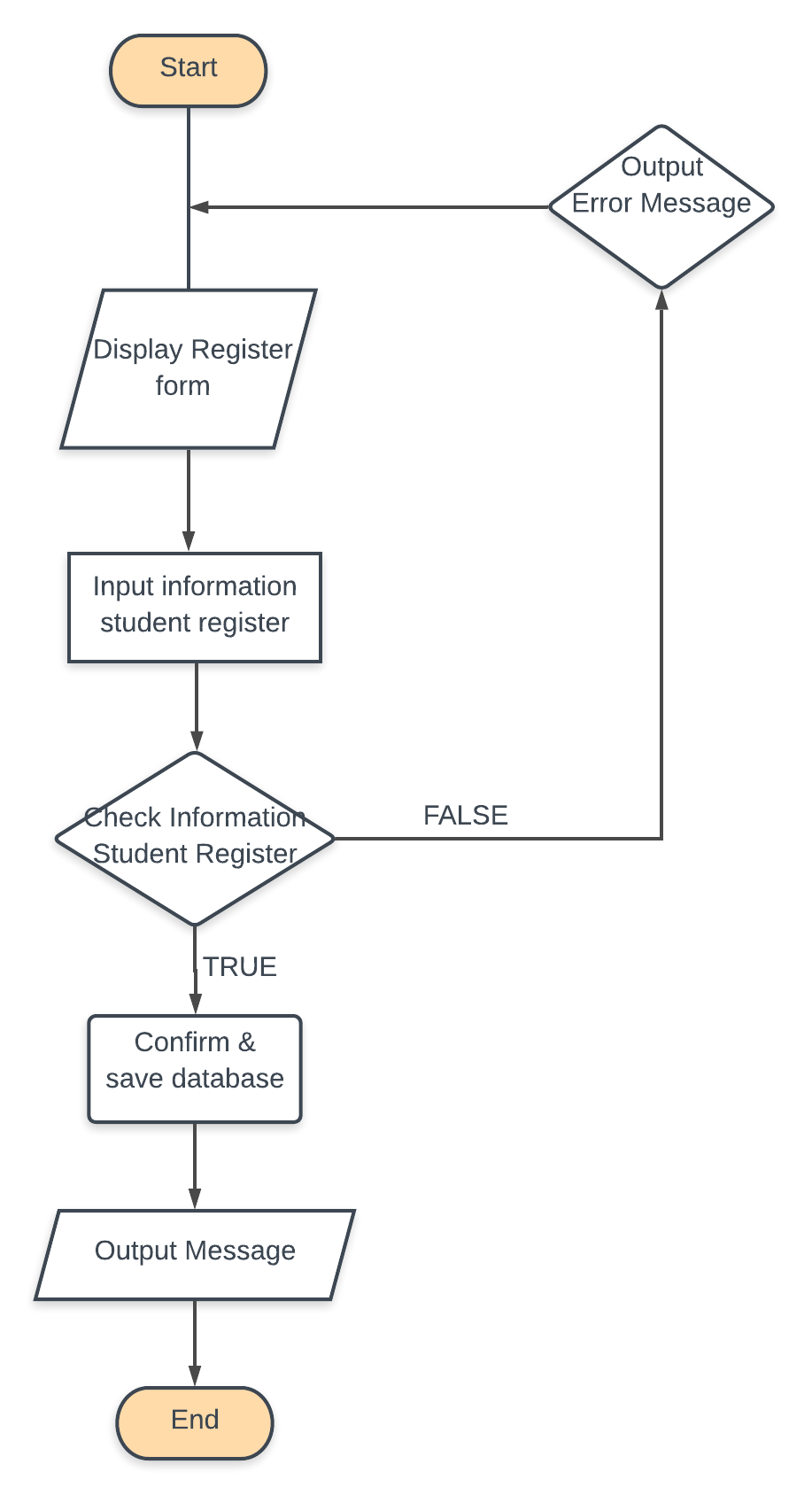
****

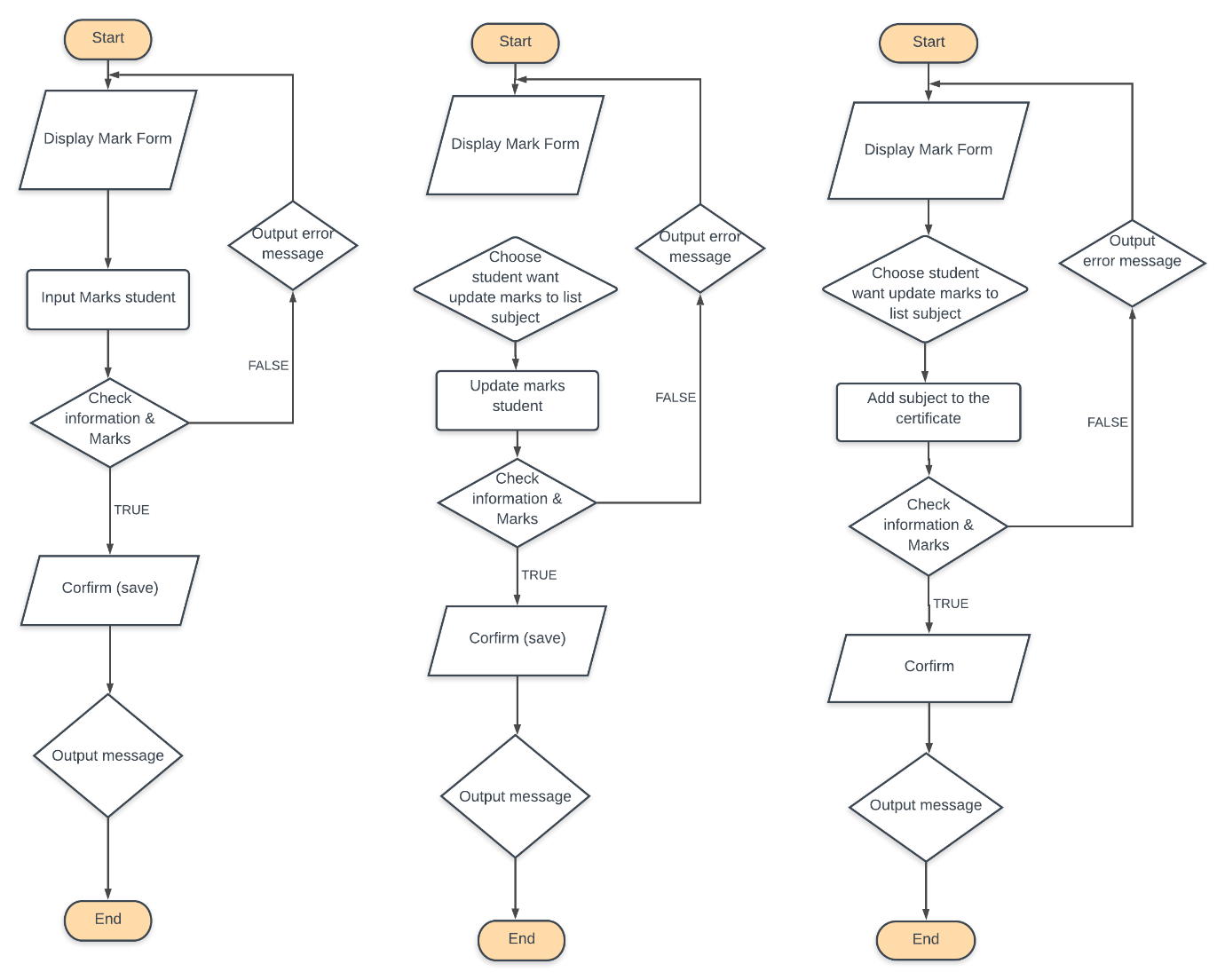
1. **Management certificate**

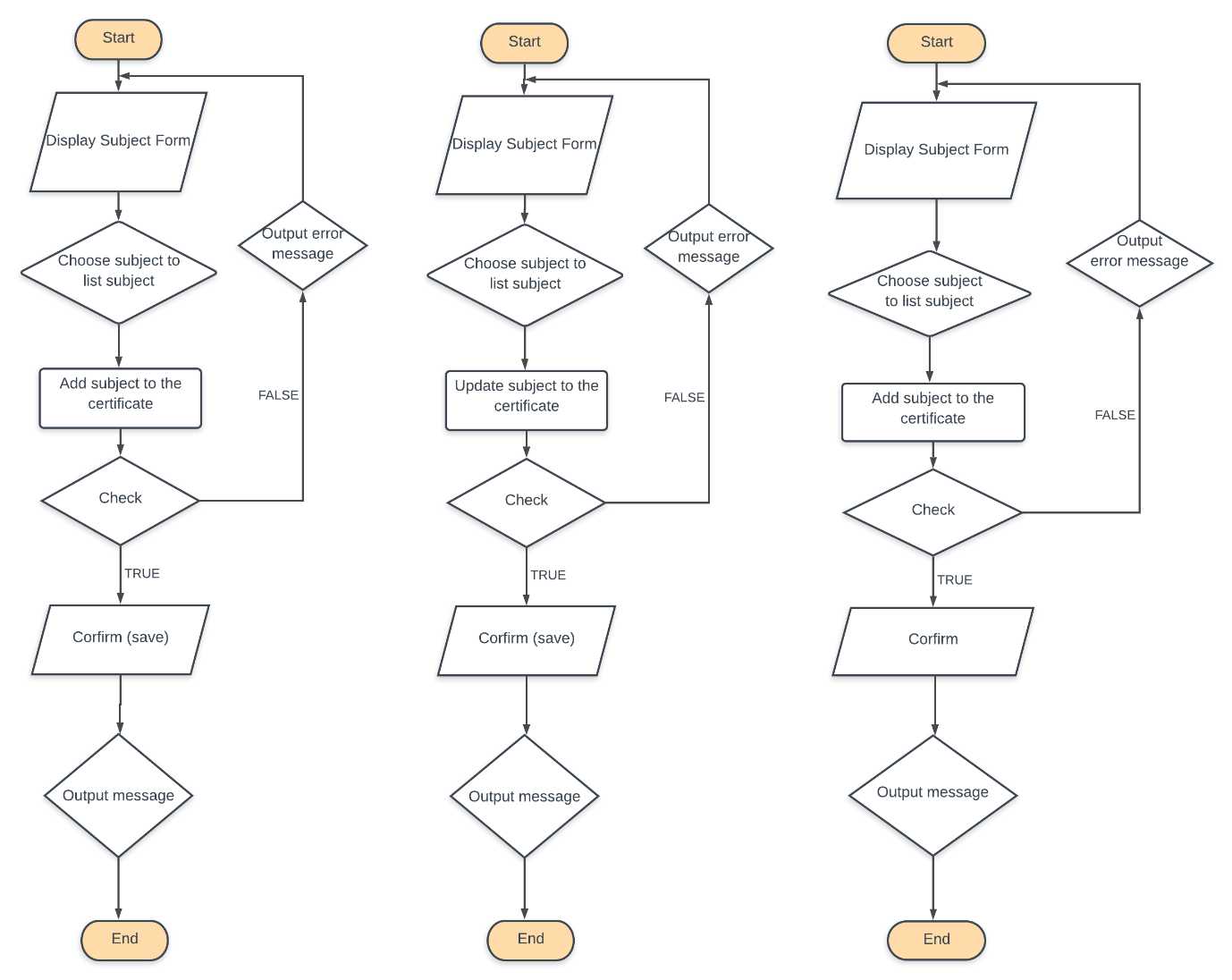
****

1. **Management Student**

****

1. **Management Registry**
2. **Management Exam Mark**

****

1. **Management subject  
   **

# **CODING STANDARDS DOCUMENT**

1. **Programming Standards**

* Every button and label should have an Icon for expressing. It must set before a text, image icons have to store in the folder img in source “[Project\_Name]/src/img” and they are imported using “Image chooser, Image within Project, Image using for project have extension \*.png.
* File, module guidelines and code layout: Code is in the Application folder including the folders (build, nbproject, src, test, build. xml, mainfest.mf, config.properties, and USERGUIDE.pdf).
* In the src folder it contains (application, database, and interfaces):
  + In application package: Contains the object class \*.java
  + In database package: Contains the class \*.java connect the database (manipulating queries on the database, add edit delete DATABASE).
  + In interfaces package: Contains the words using the Java Swing
* Must have declaration part for classes, packages using for the project.

1. **Standards for code writing style:**

* Statement of each group must indent a Tab.
* Using “{}” to block a group of statements or only one statement even and the

“{” is right after the declaration. For example:

* + if (a == 0){
    - return 0;
  + }
* Each group of a work should have a line gap.
* Database guidelines: database name, table name and column name in the database are capitalize each word. If the two become separated by underscore. The SQL statements are all required to uppercase completely.
* Comment line descrying the variable declaration function mode and end command. For example:

// Comment here

/\*\*

@Author

@Date writing

@Author modify

@Date modify

\*/

1. **Standards for declaring Variables:**

Every Variable are meaningful, the first word is lowercase and each meaning word later have the first letter is uppercase. For example: String studentID

Variables are using for loop can be “i” (Index) and loops inner can use “j” and then “k”, “m”, “n”. For example:

for (int i = 0; i < n; i++){

for (int j = 0; … //Allowed

for (int n = 0; … //Now allowed, next of “j” is “k”

}// end for

In addition, in some cases, the variable name needs to clearly represent the data type of that variable. For example, a variable of type List should be named studentList, a variable of type Set should be named studentSet, a variable of type Map should be named studentMap, and a variable of type Array should be named studentArray.

1. **Standards for Function Declarations:**

Must have declaration for any functions and it declares return value, parameters if necessary. For example:

//function return age

public int getAge (int n) {

return n;

}// end getAge

Any lines have to fix in 80 character, if not, detach it.

1. **Other Standards:**

The name of the constant must follow the general rule and must be capitalized. If the constant name has two words or more, there must be an underscore between words. For example: PI, NO\_FIBONACCI.

The first letter of the first word in the method name must be lowercase and a verb, and the first letter of the next words must be capitalized (like the variable naming rule). For example: addStudent.

Package names must follow the general rule and must be in lowercase.

Class names should have words that have suffixes to make them more expressive, such as AddTwoNumber.

Rules for coding: Required comments (author, date of writing, note).

If explicit edit, write to comment (author edit, day edit).

If using the try – catch should use display error messages.

# **TASK SHEET**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project Ref. No.:** | **Project Title:** | **Activity Plan**  **Prepared By:**  **Nghi Dang Quang** | | **Date of Preparation of Activity Plan: Oct 9th, 2019** | | |
| **Task Sub division** | **Description** | **Planned Start Date** | **Actual Start Date** | **Actual Days** | **Team Member Names** | **Status** |
| Form  No: 1 | Problem Definition | Oct 9th, 2019 | Oct 10th, 2019 | 2 | All member | Done |
| Form  No: 2A | CRS | Oct 11th, 2019 | Oct 13th, 2019 | 3 | All member | Done |
| Form  No: 2B | Customer Acceptance Criteria | Oct 14th, 2019 | Oct 15th, 2019 | 2 | All member | Done |
| Form  No: 9 | Task Sheet | Oct 15th, 2019 | Oct 15th, 2019 | 1 | Nghi  Dang Quang | Done |
| Form  No: 3 | Project Plan | Oct 16th, 2019 | Oct 16th, 2019 | 1 | Nghi  Dang Quang,  Khanh  Nguyen Le | Done |
| Form  No: 4 | Process Design Document | Oct 17th, 2019 | Oct 18th, 2019 | 2 | Thang  To Toan,  Thanh  Nguyen Phuoc | Done |
| Form  No: 5 | Process Description Document | Oct 19th, 2019 | Oct 19th, 2019 | 1 | Khanh  Nguyen Le | Done |
| Form  No: 6 | Interface Design Document | Oct 20th, 2019 | Oct 25th, 2019 | 6 | All member | Done |
| Form  No: 7 | Table Design Document | Oct 26th, 2019 | Oct 29th, 2019 | 4 | All member | Done |
| Form  No: 8 | GUI Standards Document | Oct 30th, 2019 | Oct 30th, 2019 | 1 | Thang  To Toan | Done |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
| Form  No: 10 | Coding Standards Document | Oct 31st, 2019 | Nov 12th,2019 | 13 | Thanh  Nguyen Phuoc | Done |
| Form  No: 11 | Integration Testing Report | Nov 13th,2019 | Nov 17th,2019 | 5 | All member | Done |
| Form  No: 12 | Review Report | Nov 18th,2019 | Nov 18th,2019 | 1 | Khanh  Nguyen Le | Done |
| Form  No: 13 | Final Checklist | Nov 19th,2019 | Nov 26th,2019 | 8 | All member | Done |

# **INTEGRATION TESTING REPORT**

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Features Tested** | **Remarks** |
| **1** | **Design all form** | **Not good (maximize …)** |
| **2** | **Design database** | **Not create trigger, view, identity and function.** |
| **3** | **Design and Coding form connect to database** | **Using Resource Bundle \*.properties** |
| **4** | **Design and coding form users login** | **Not option each roles** |
| **5** | **Design and coding form home** | **Not responsive and note date coding modify, search for identity card** |
| **6** | **Design and coding form management student for add, delete, edit and report** | **Not create transaction for student impact of online system. Not responsive** |
| **7** | **Design and coding form management certificate for add, delete, edit and report** | **Verify that the graphical displays use color to the maximum extent possible.** |
| **8** | **Design and coding form management subject for add, delete, edit and report** | **Verify that the graphical displays use color to the maximum extent possible and trigger** |
| **9** | **Design and coding form management course of exam for add, delete, edit and report** | **Done, date must days 21 to 1 year.** |
| **10** | **Design and coding form management registry for add, delete, edit and report list of student.** | **Not login with student, must add student first.** |
| **11** | **Design and coding form management exam mark for add, delete, edit and report** | **Must registry first.** |
| **12** | **Design and coding form management certificate store for add, delete, edit and report** | **Cannot insert duplicate key in object (unique, primary key)** |
| **13** | **Report List student fail** | **Not title on file \*.xlsx** |
| **14** | **Report List student pass** | **Not title on file \*.xlsx** |
| **15** | **Print certificate by Roll No** | **Files Save as \*.pdf** |
| **16** | **Verify that offline help is available for all features.** | **Display file \*.pdf** |
| **17** | **Design user guide document** | **Include files \*.docx, \*.pdf** |
| **18** | **Setup and training for customer.** | **Include files \*.exe, \*.pptx** |

# **REVIEW REPORT**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date:** | **Project Plan Activity/ Milestone** | **Work Specification** | **Status of the Activity** | **Remarks** | **Responsibility** |
| Oct 9th, 2019 | Oct 15th, 2019 | Take Customer’s information | Done |  | Thang To Toan |
| Oct 16th, 2019 | Oct 30th, 2019 | Design | Done |  | Thanh Nguyen Phuoc |
| Oct 31th, 2019 | Nov 12th,2019 | Coding | Done |  | Nghi Dang Quang |
| Nov 13th,2019 | Nov 24th,2019 | Testing | Done |  | Nghi Dang Quang |
| Nov 25th,2019 | Nov 26th,2019 | Training | Done |  | Khanh Nguyen Le |

# **FINAL CHECKLIST**

**(This document has to be filled by the teacher only. The teacher will do a complete functionality testing of the application and his/ her findings and suggestions to this form.)**

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Aspect tested** | **Suggestion/ Remarks** |
| 1. | Have all the modules been properly integrated and are they completely functional? |  |
| 2. | Does each unit meet its objective and purpose?  Are all the validations happening as specified in Process Design? |  |
| 3. | Have all Design and Coding standards been followed and implemented? |  |
| 4. | Is the GUI design consistent all over? |  |
| 5. | Are the codes working as per the specification? |  |
| 6. | Does the application’s functionality resolve the client problem, and satisfy his needs completely? |  |
| 7. | Have the hardware and software been correctly chosen? |  |
| 8. | Additional features and utilities that give value addition to the entire project. |  |

# **USER GUIDE**

