**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 produce 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 wishes 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 actually 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, Cer\_Description)

STUDENT\_CER (**StudentID, CertificateID**, **Exam\_Date,** Result)

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

EXAMMARK (**StudentID, Exam\_Date, CertificateID, SubjectID**, Mark)

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

USER\_PERMISS (**UserID, PerID**)

PERMISS (**PerID,** PerName)

DETAIL\_PER (**DetPerID**, ActionName, PerDate)

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 enter 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. Server:
     + CPU: Intel Core i3-2130U, 2.5GHz
     + RAM: 4GB or higher
     + HDD: 120GB Hard Disk space or higher
  2. Client:
     + CPU: Intel Core i3-2130U, 2.5GHz
     + RAM: 2GB or higher
     + HDD: 80GB Hard Disk space or higher
* **Software Requirements:**
  1. Server:
     + Windows Server 2012 R2 (or higher).
     + Microsoft SQL Server 2012 (or higher).
     + JDBC, JDBC Driver 4, JDK 1.8 (or higher).
     + Netbeans IDE 8.2 (or higher).
  2. Client:
     + 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:**

(Define the project vision/ objectives as stated by the client)

**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:**

(Mention the scope of the project giving the locations that will be covered, processes, range of services, and so on.)

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

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

(Give the range of services, functions, overview of processes, and so on.)

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 three logins.

* + - **User registry for the student.**
    - **Certificate Cell.**
  + **Project Organization with Responsibilities and Authorities:**

(Give the name of Project team members their roles and responsibilities.

|  |  |
| --- | --- |
| **Members** | **Role and Responsibilities** |
| Nghi Dang Quang | **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**

(Information required from the Client as inputs regarding his system; could be the information about his services, processes.)

Generate information about Certificate for the students.

1. **Deliverables**

(The documents to be handed over to the client – like CRS, Design Document, Installation Manual, User Manual, Maintenance Manual, and Code documents.).

* Code and some files \*.jar, \*.sql, \*.mdf, \*.ldf

1. **Project Dependencies**

(Any event or task outside the scope of the project, which will affect the success of the project.)

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

1. **Major Project Milestones**

(Generating CRS, Building a Prototype.)

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

1. **Quality Plan**
   * **Review Activities** (Review meeting participants, frequency): 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** (Final Test): Test spell design document, test case design DFD and database, Function, Coding and Demo.
   * **Backup and recovery strategies** (in case of disk crash, network failures): 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 | Blue |
| 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** |
| **frDatabase** |  |  |
| **frLogSQL** |  |  |
| **frUserlogin** |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **frManager** |  |  |
| **frRegister** |  |  |
| **frEnMark** |  |  |

|  |  |  |
| --- | --- | --- |
| **frMCourse** |  |  |
| **frMCertifi** |  |  |
| **frMStudent** |  |  |

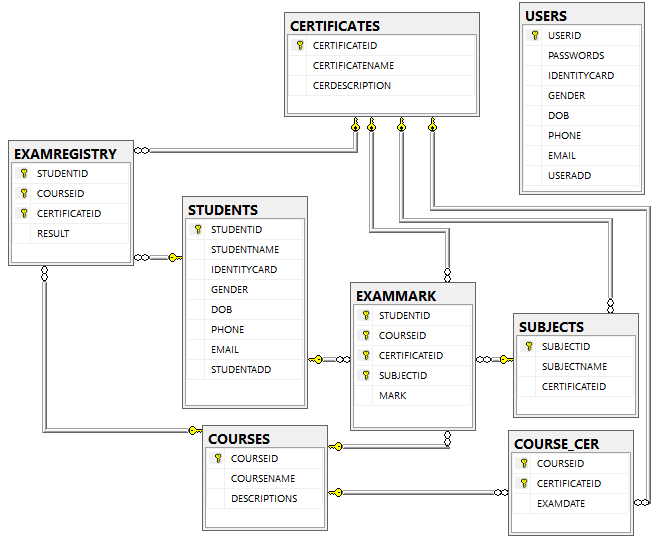
|  |  |  |
| --- | --- | --- |
| **frMSubject** |  |  |
| **frMReport** |  |  |
| **frMPrint** |  |  |

|  |  |  |
| --- | --- | --- |
| **frMSearch** |  |  |

1. **Database Structure**

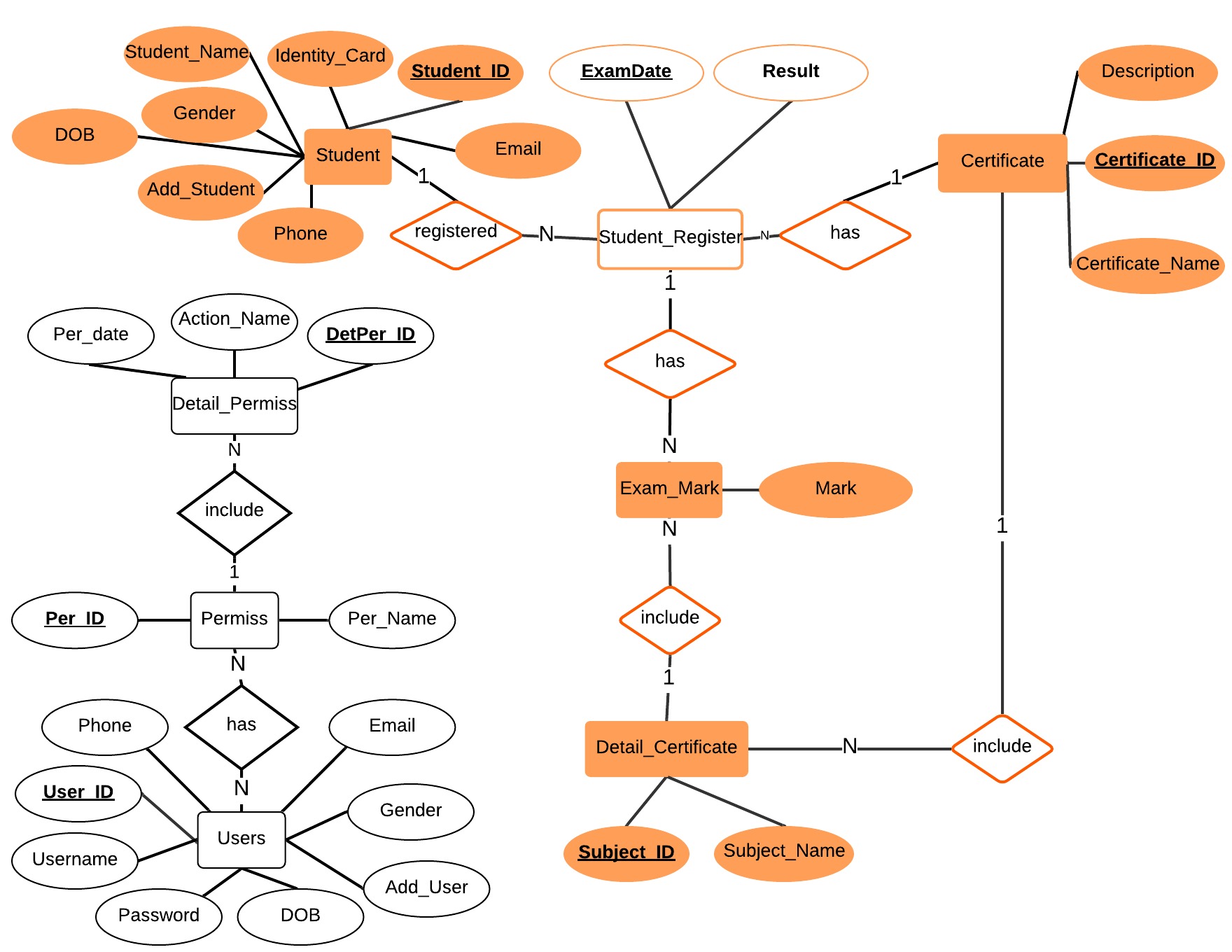
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name of the Table** | **Table Description** | **Number of Fields** | **Primary Key** | **Related Tables** | **Foreign Key** |
| Student |  | 8 | **StudentID** | Subject, Exammark |  |
| Certificate |  | 3 | **CertificateID** | Student\_CerSubject, Exammark |  |
| Student\_Cer |  | 4 | **StudentID,**  **ExamDate CertificateID** | Student, Certificate, Examresult, Exammark | ***StudentID,***  ***CertificateID*** |
| Detail\_CER |  | 3 | **SubjectID** | Certificate,  Examresult, Exammark | ***CertificateID*** |
| ExamMark |  | 5 | **StudentID, CourseID, CertificateID, SubjectID** | Student\_Cer  Subject | ***StudentID, ExamDate, CertificateID, SubjectID*** |
| Users |  | 8 | **UserID** |  |  |
| Permiss |  | 2 | **PerID** | DetailPer |  |
| User\_Per |  | 2 | **UserID, PerID** | Permis, Users | ***UserID, PerID*** |
| DetailPer |  | 4 | **DetperID** | Permiss | ***PerID*** |

1. **Algorithms and data structures**



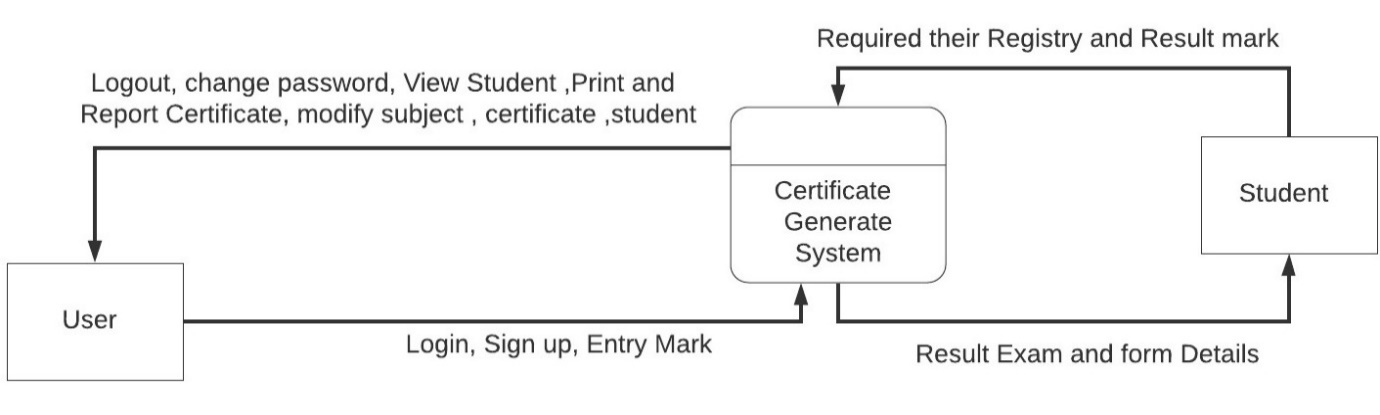
# **PROCESS DESCRIPTION DOCUMENT**

1. **ER Diagram**

****

**Figure FormNo7.1: ERD**

1. **Data Flow Diagram Context.**

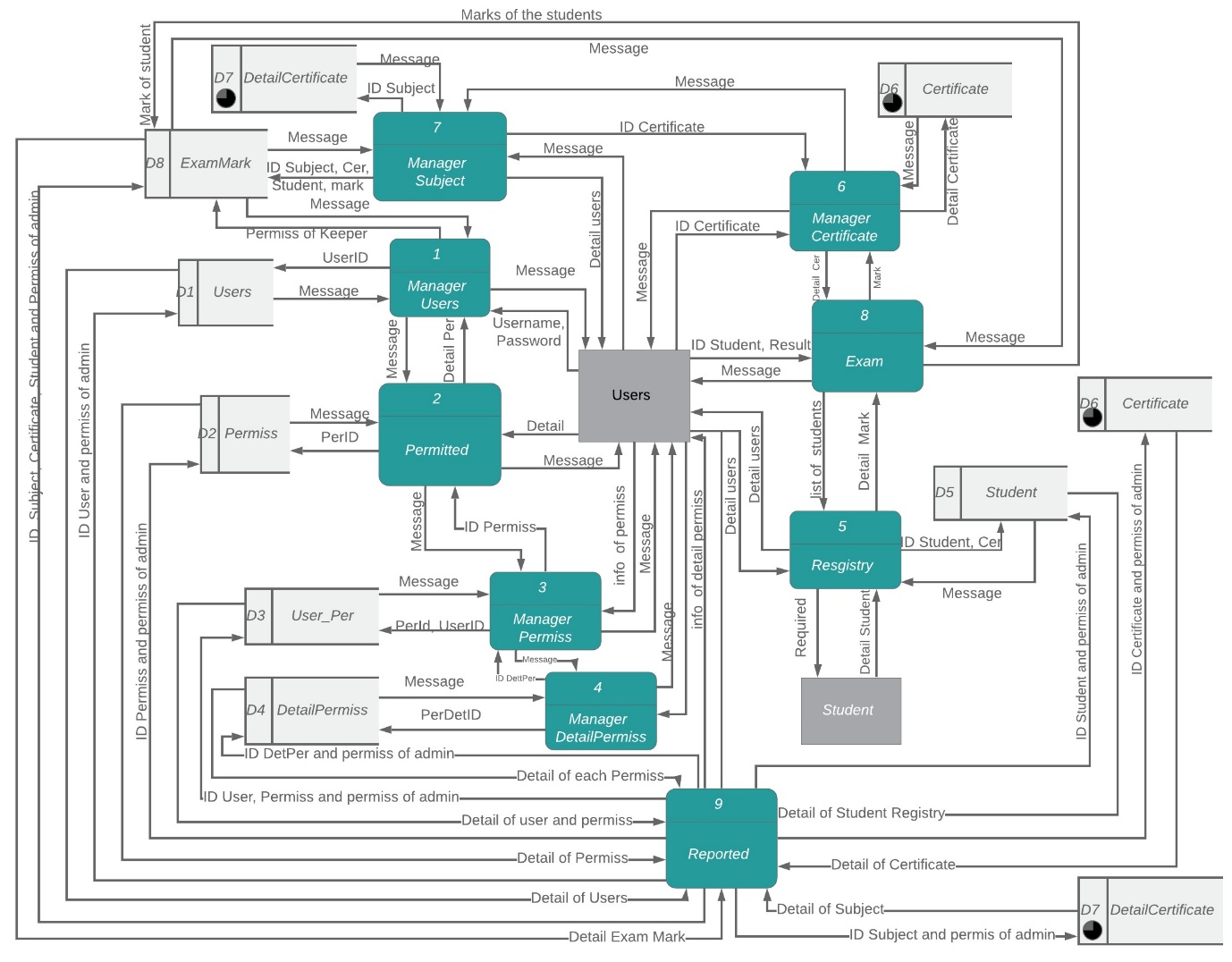
****

**Figure FormNo7.2: DFD 0 Level**

**Process Name: DFD context.**

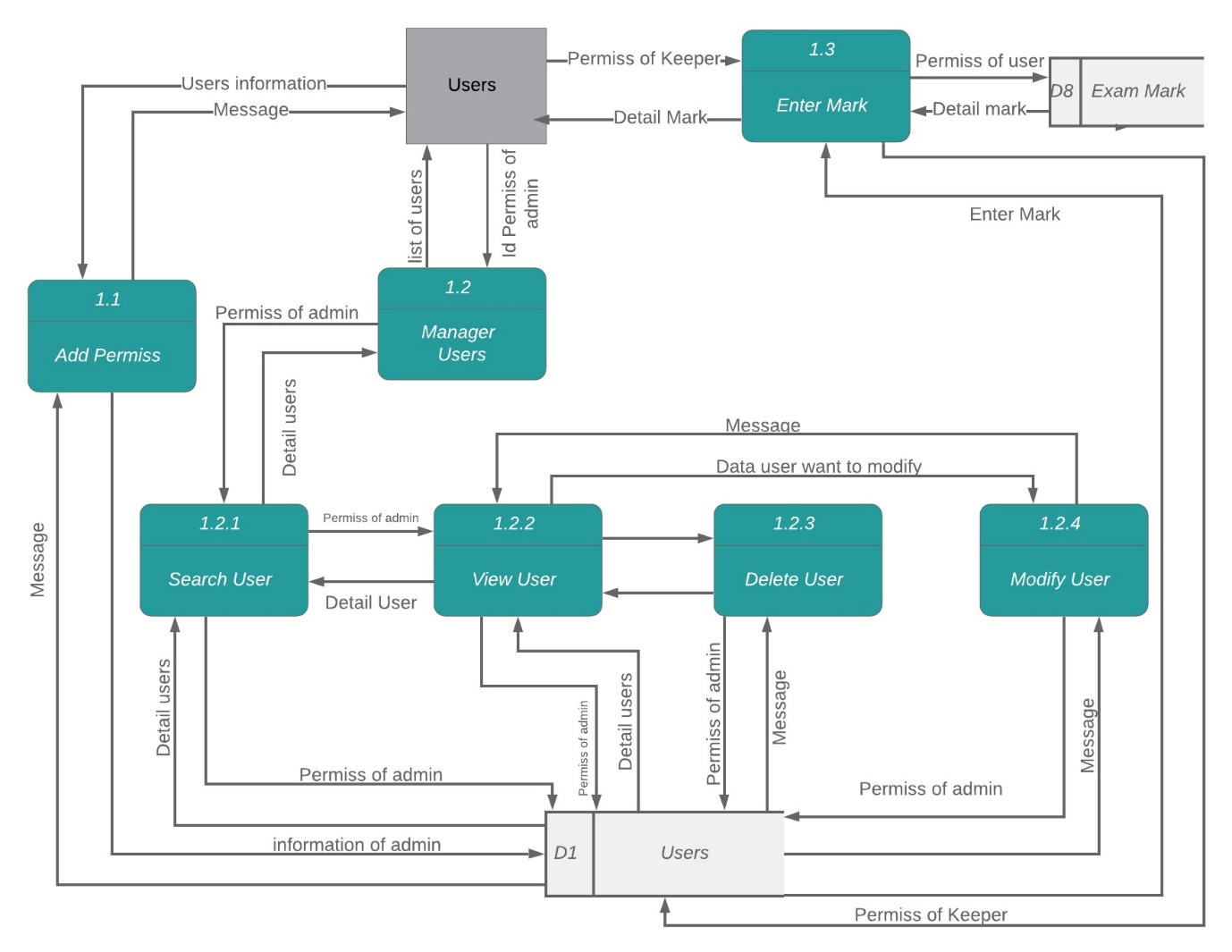
**Process Description and Details:**

1. **DFD System:**

****

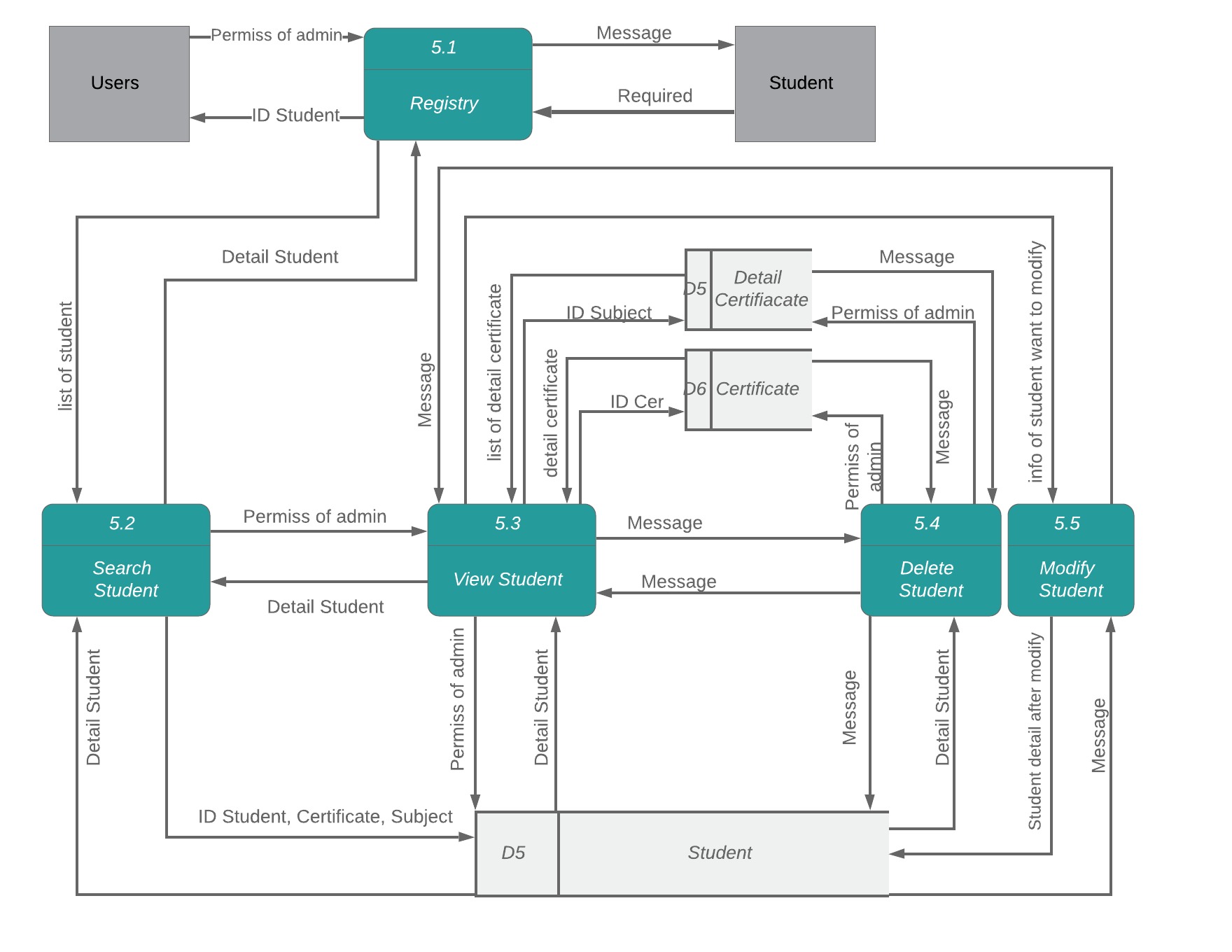
**Figure FormNo7.2: DFD 0 Level**

1. **DFD process 1**

****

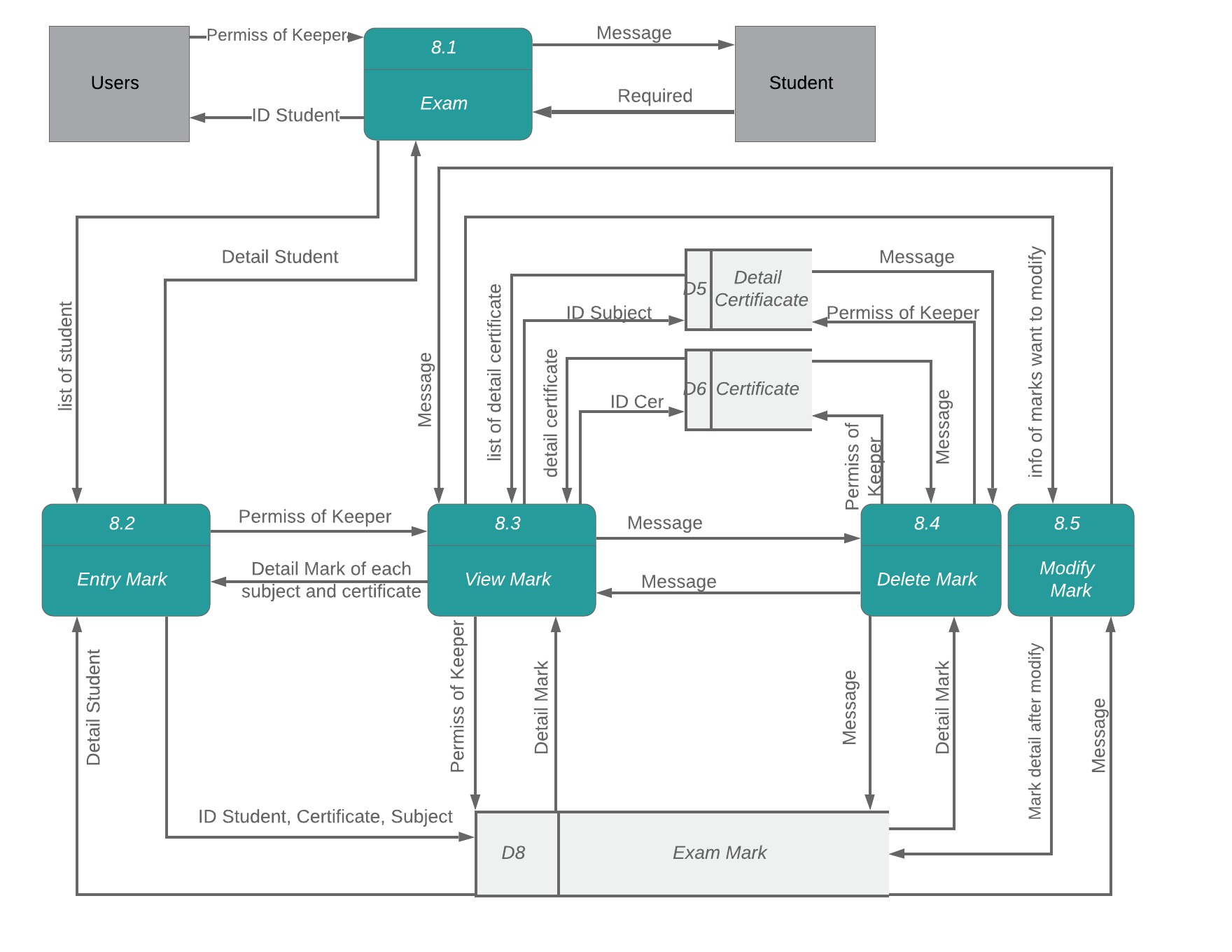
**Figure FormNo7.2: DFD 0 Level**

1. **DFD process 2**

****

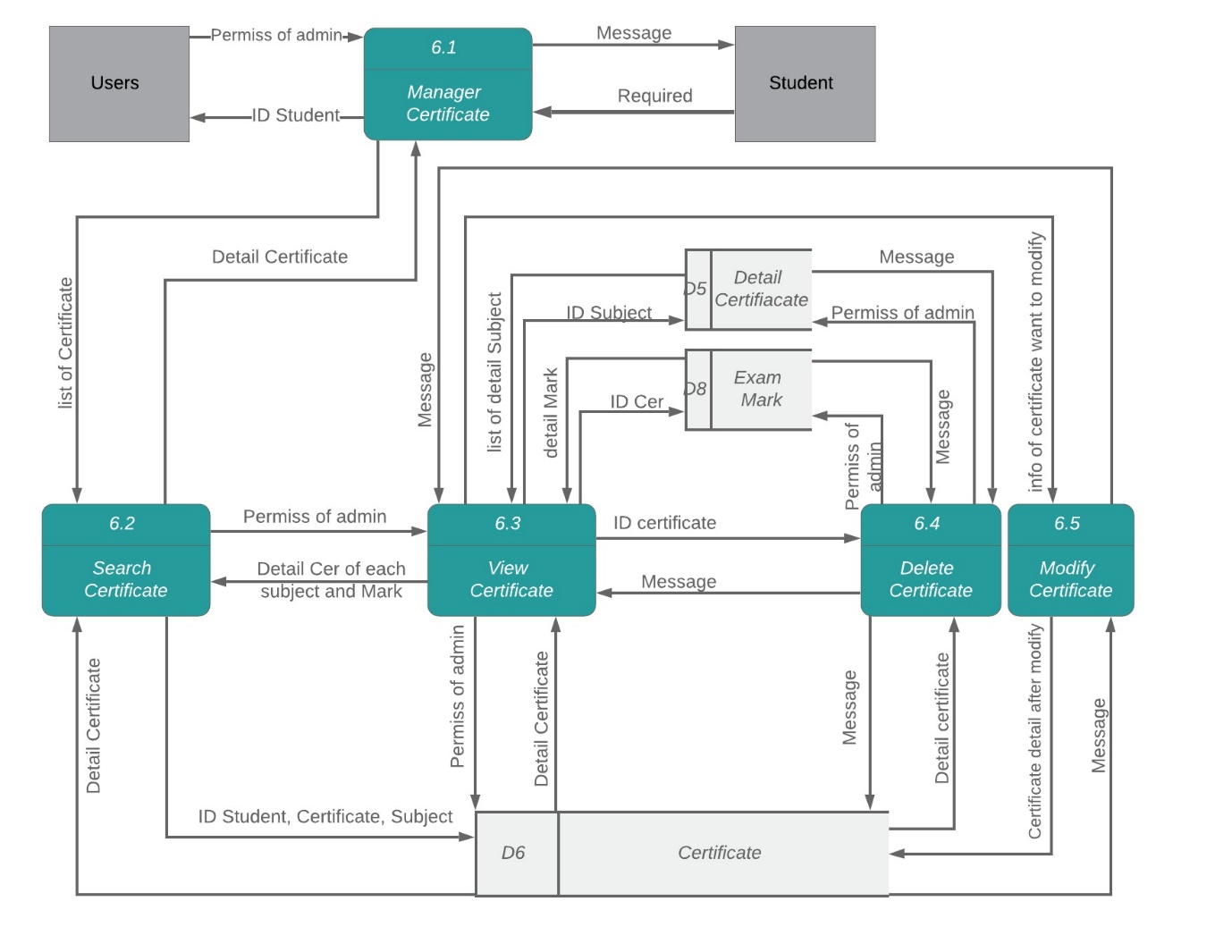
**Figure FormNo7.2: DFD 0 Level**

1. **DFD process 3**

****

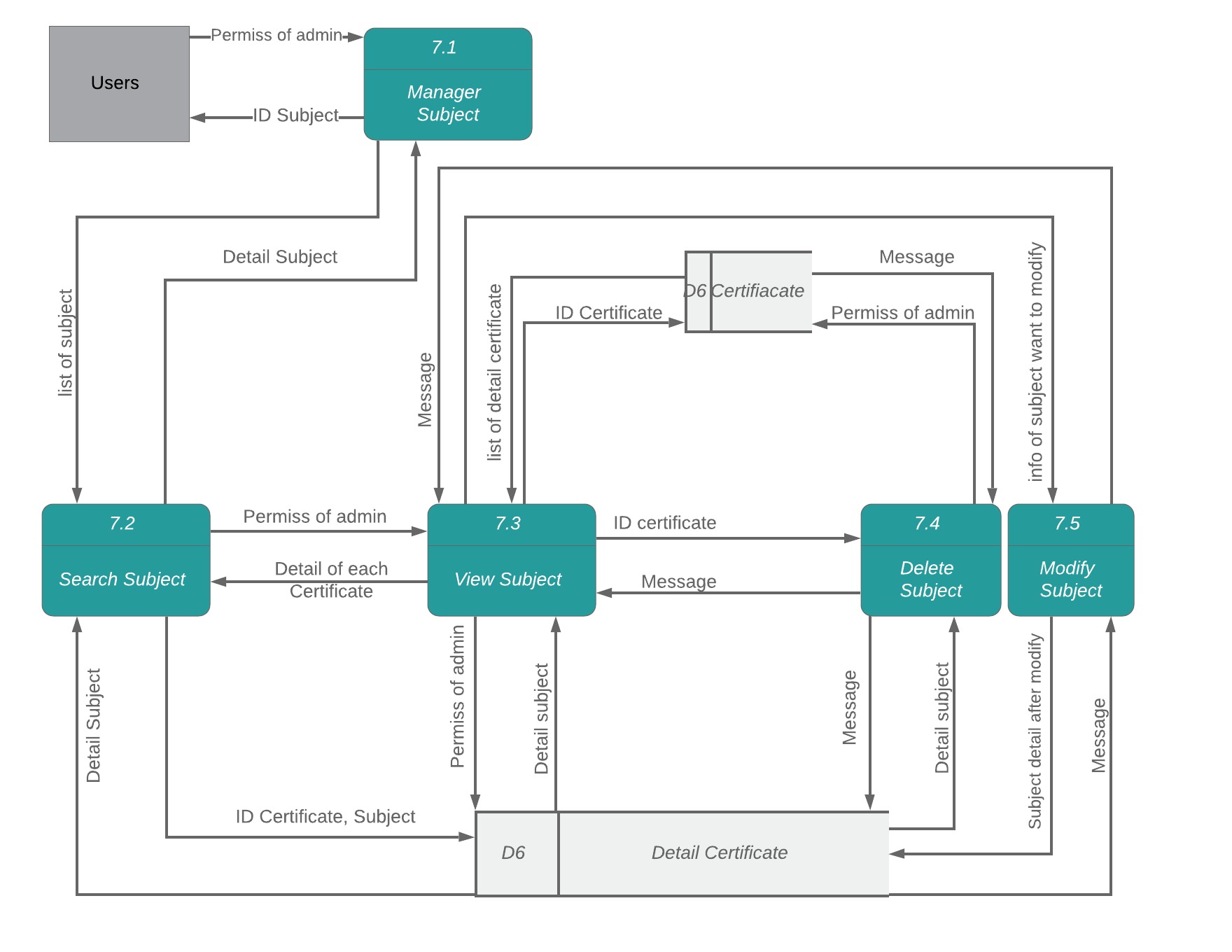
**Figure FormNo7.2: DFD 0 Level**

1. **DFD process 4**

****

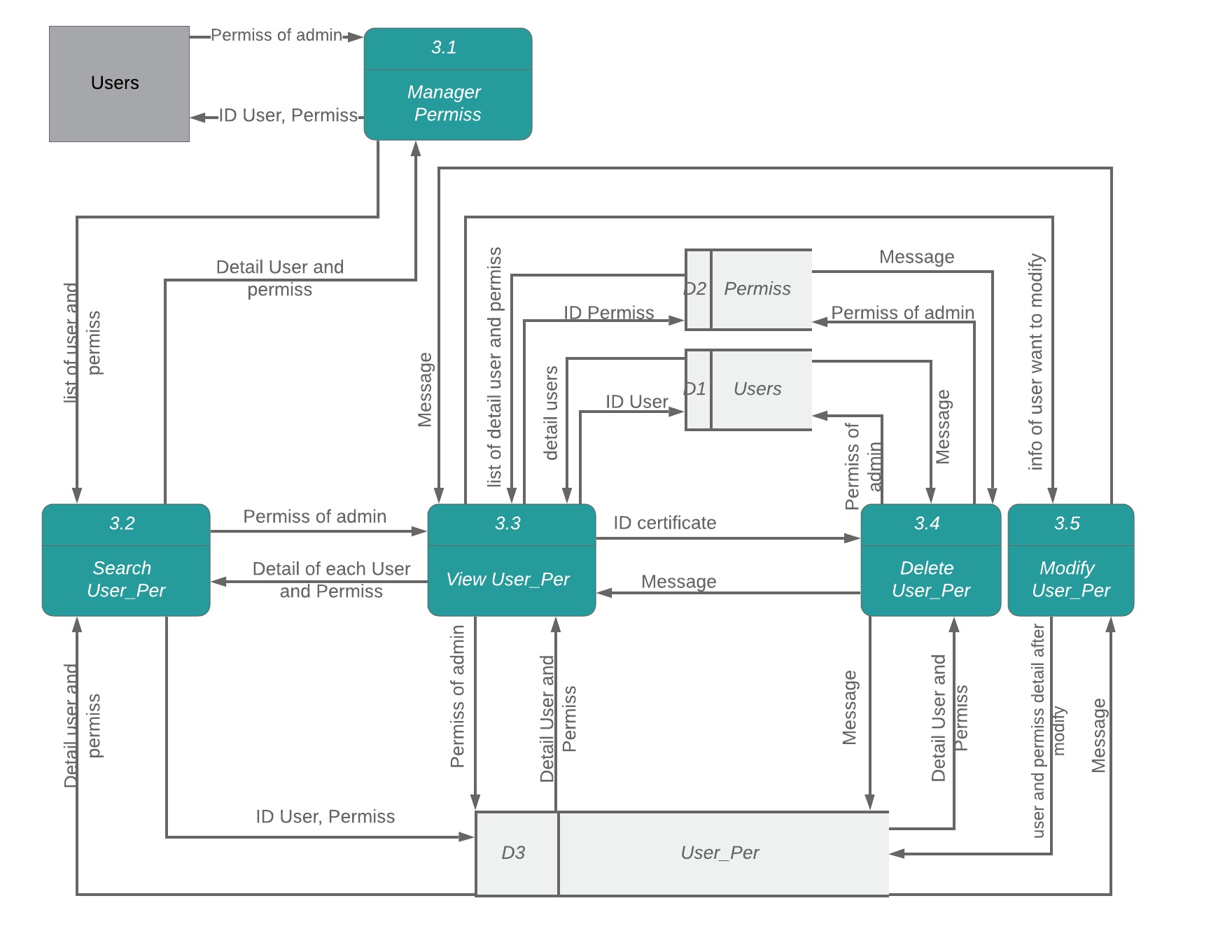
**Figure FormNo7.2: DFD 0 Level**

1. **DFD process 4**

****

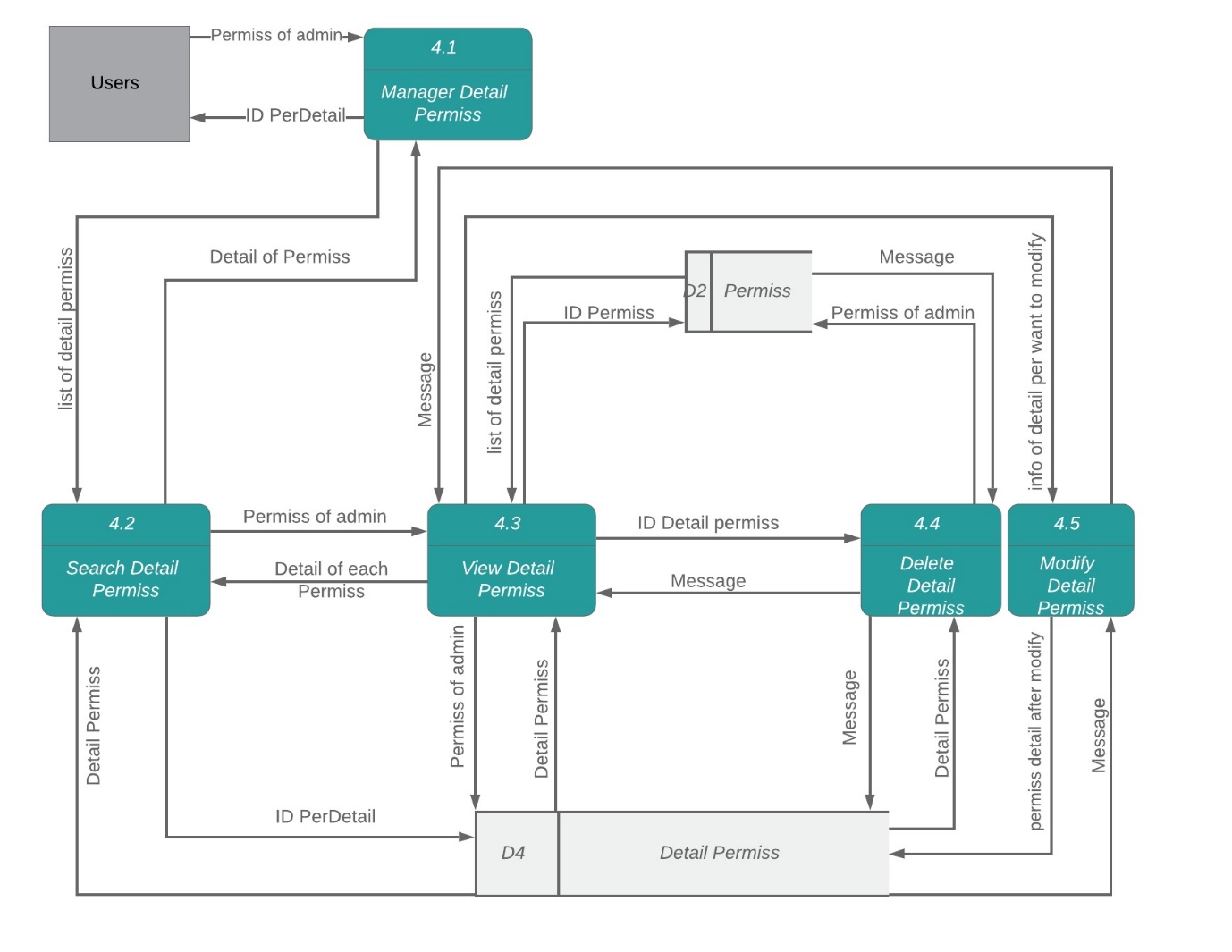
**Figure FormNo7.2: DFD 0 Level**

1. **DFD process 4**

****

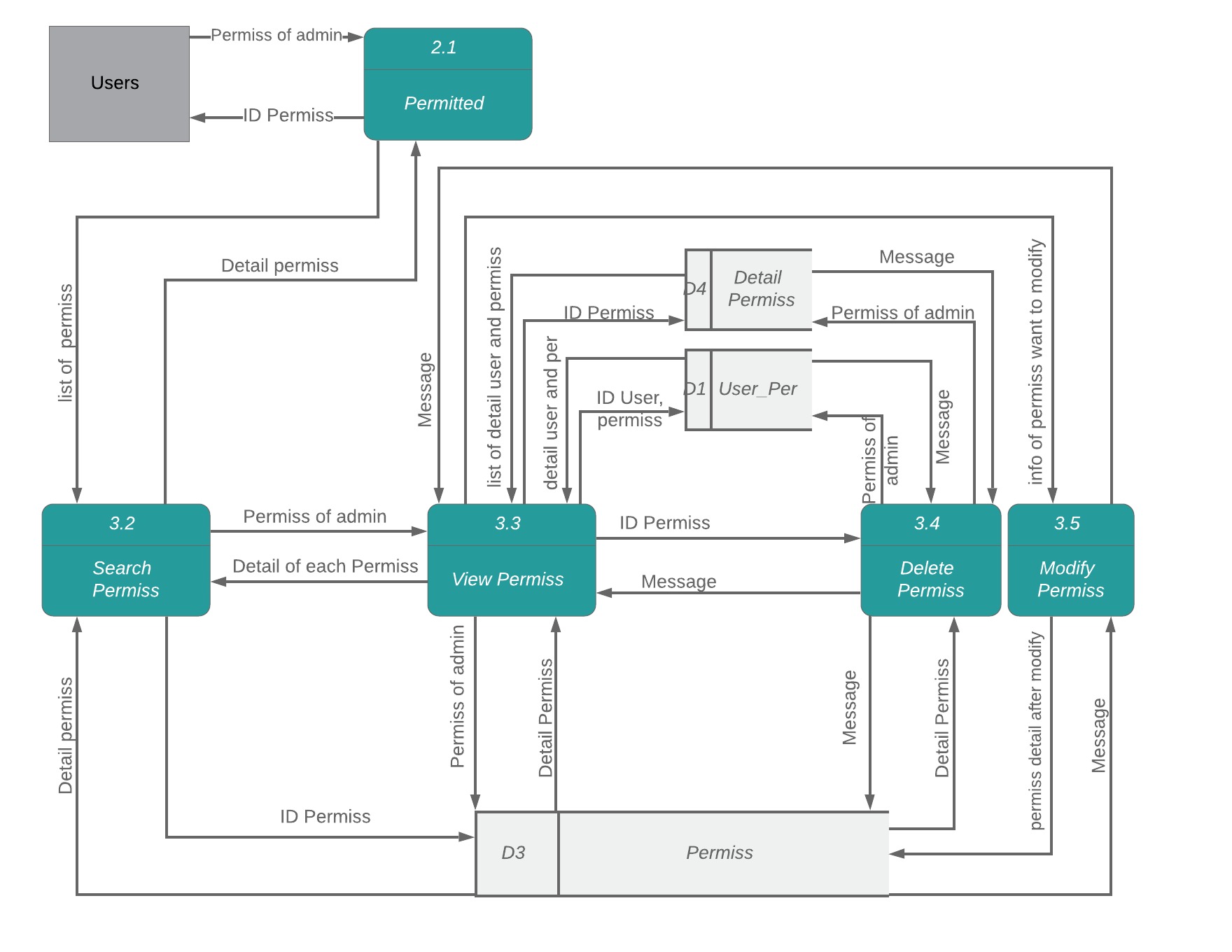
**Figure FormNo7.2: DFD 0 Level**

1. **DFD process 4**

****

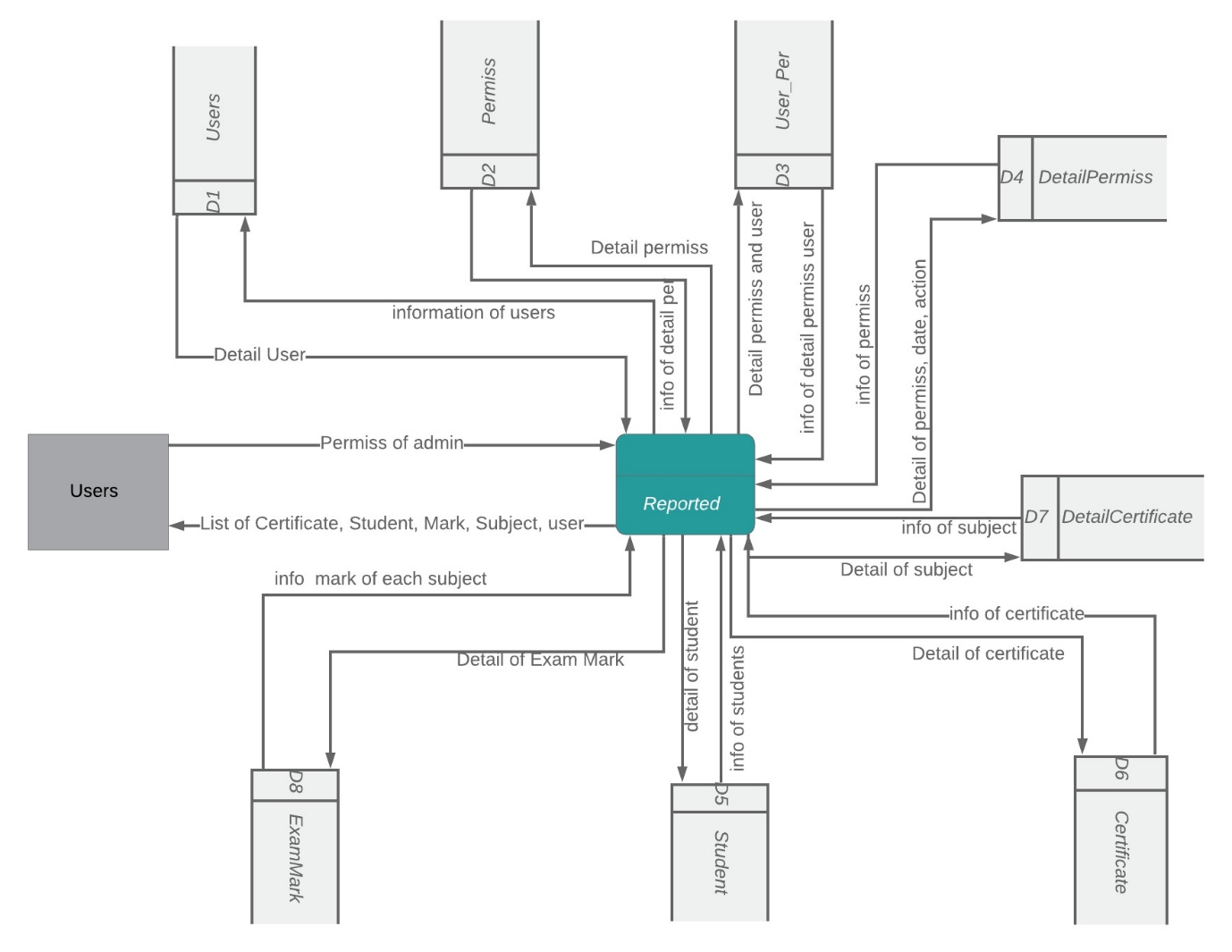
**Figure FormNo7.2: DFD 0 Level**

1. **Ds**

****

**Figure FormNo7.2: DFD 0 Level**

1. **DFD process 4**

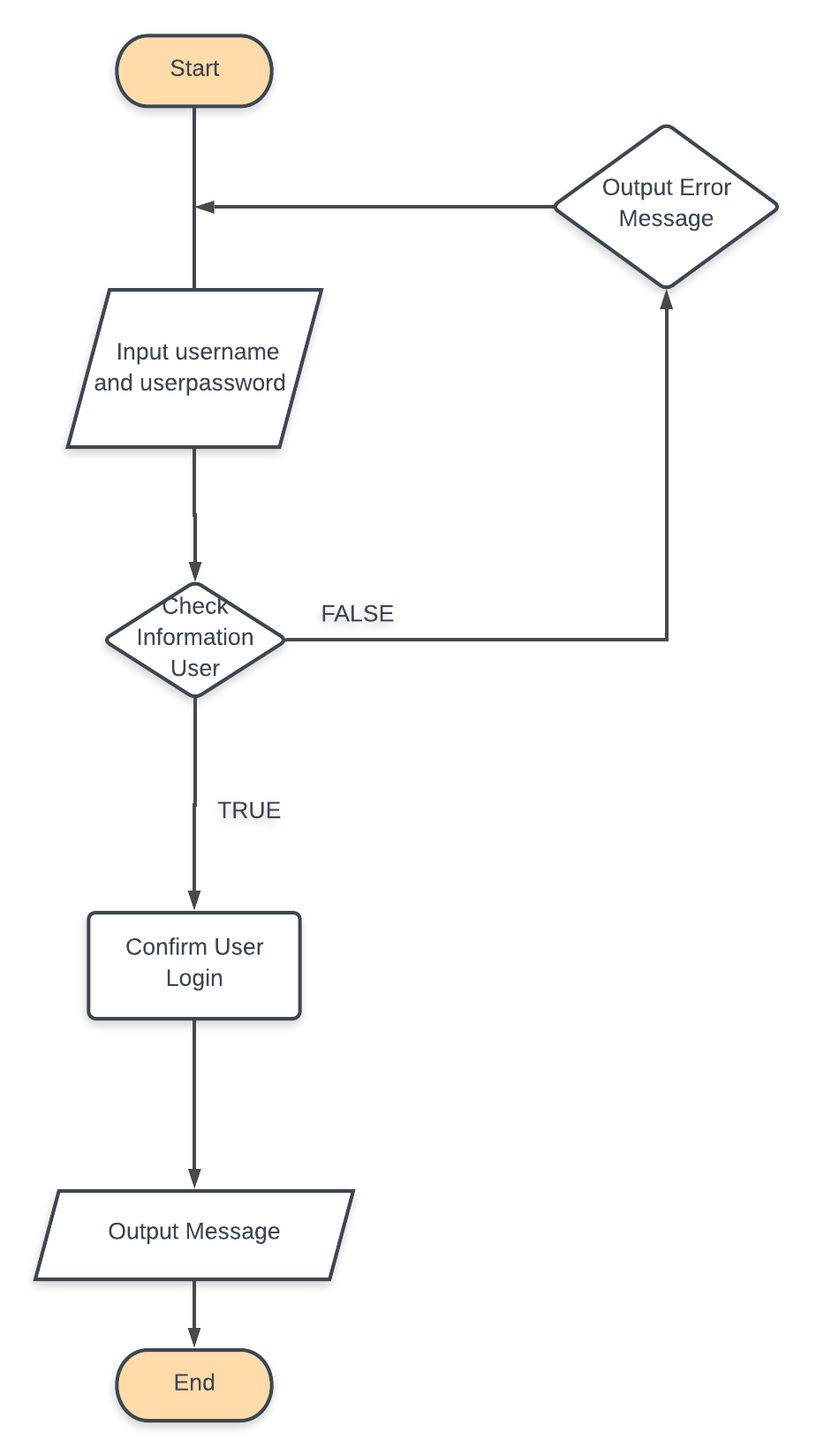
****

**Figure FormNo7.2: DFD 0 Level**

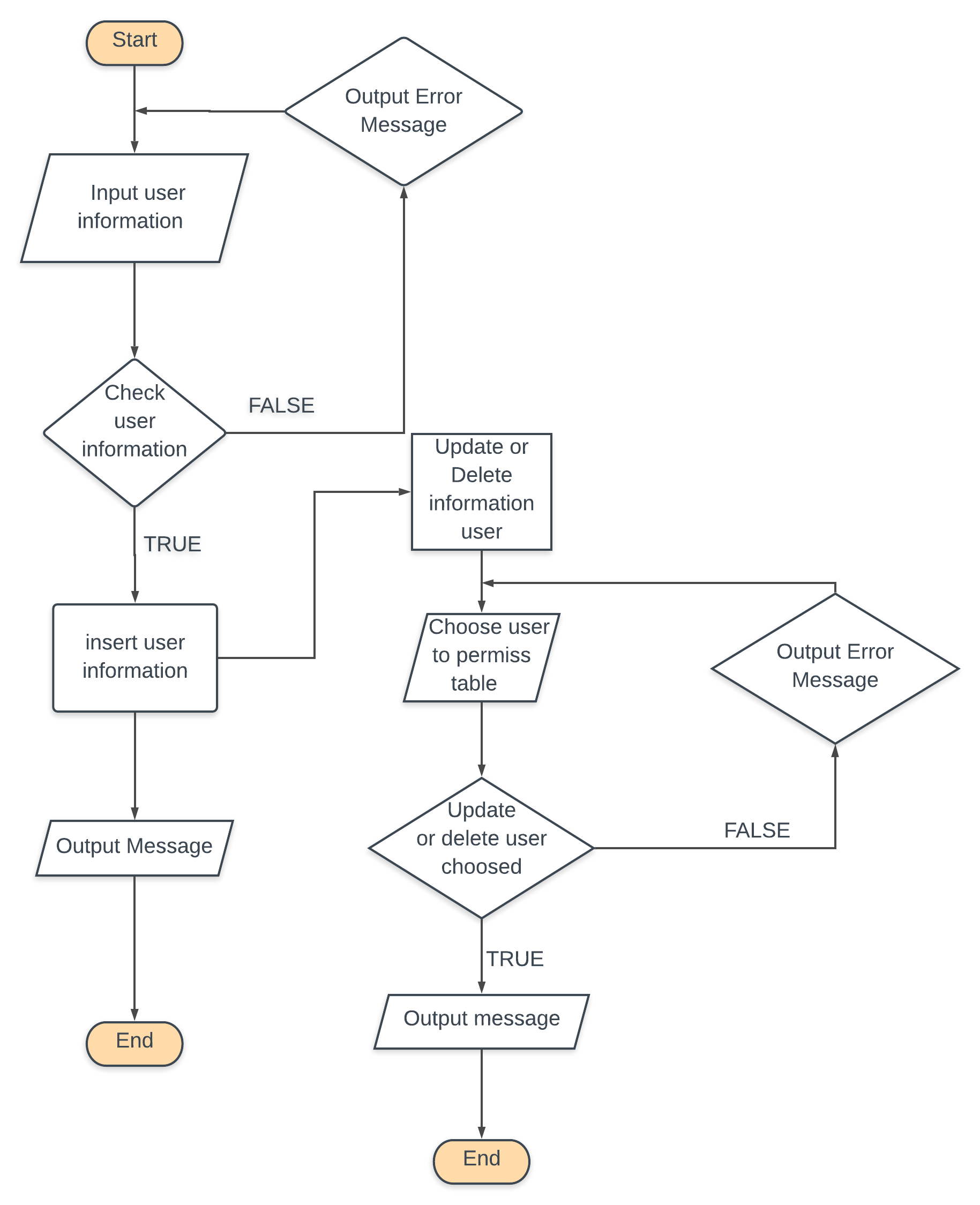
1. **Process Name Flow Chart:**

**Process Description and Details:**

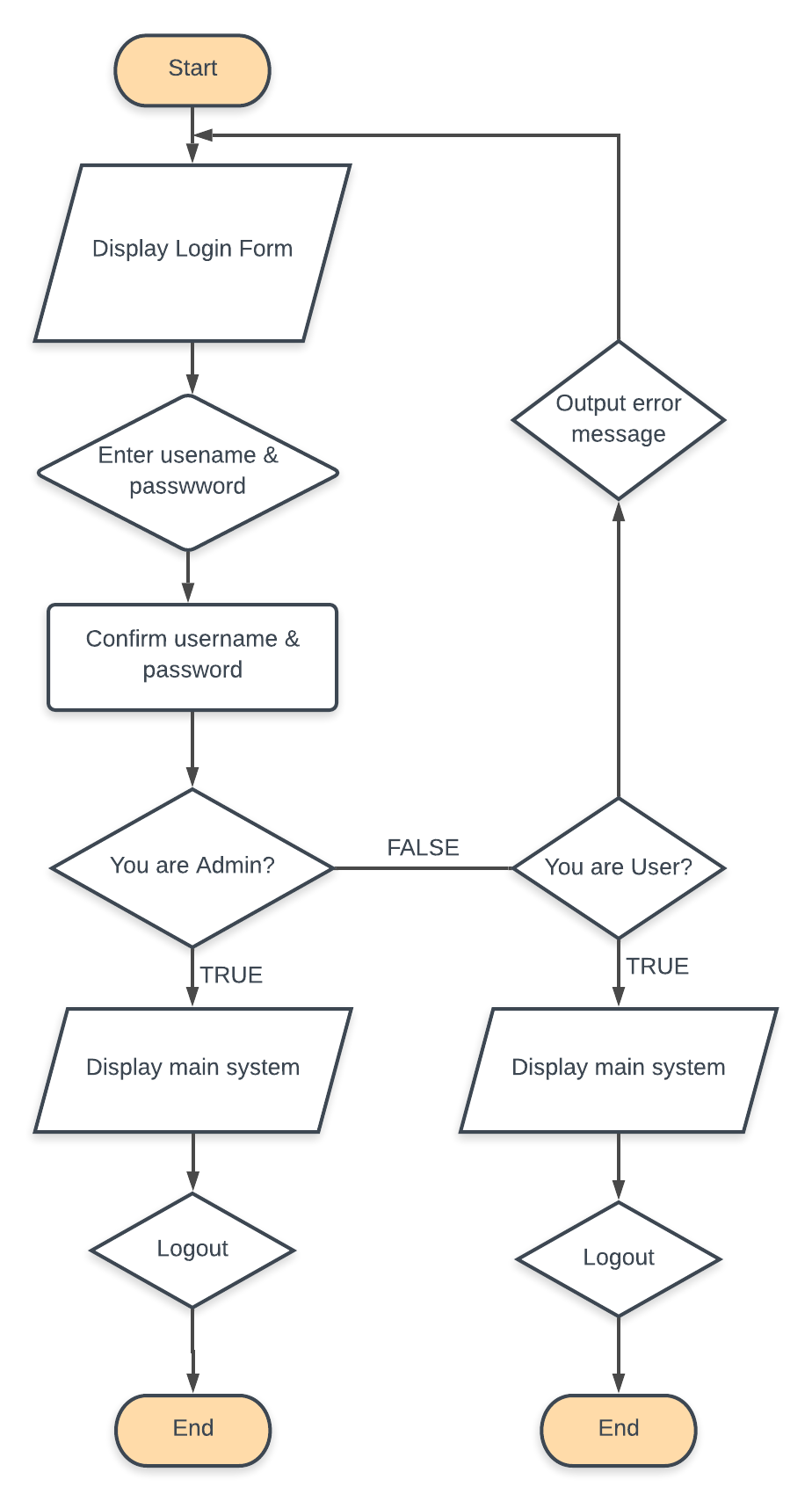
1. **Login System.**

****

1. **User management**

****

1. **Permits management**

****

1. **Management certificate**
2. **Management Student**
3. **Management Registry**
4. **Management Exam Mark**
5. **Management**
6. **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).
* 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 indented 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 | Pending |
| Form  No: 4 | Process Design Document | Oct 17th, 2019 | Oct 18th, 2019 | 2 | Thang  To Toan,  Thanh  Nguyen Phuoc | Pending |
| Form  No: 5 | Process Description Document | Oct 19th, 2019 | Oct 19th, 2019 | 1 | Khanh  Nguyen Le | Pending |
| Form  No: 6 | Interface Design Document | Oct 20th, 2019 | Oct 25th, 2019 | 6 | All member | Pending |
| Form  No: 7 | Table Design Document | Oct 26th, 2019 | Oct 29th, 2019 | 4 | All member | Pending |
| Form  No: 8 | GUI Standards Document | Oct 30th, 2019 | Oct 30th, 2019 | 1 | Thang  To Toan | Pending |

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

# **INTEGRATION TESTING REPORT**

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Features Tested** | **Remarks** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# **REVIEW REPORT**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date:** | **Project Plan Activity/ Milestone** | **Work Specification** | **Status of the Activity** | **Remarks** | **Responsibility** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# **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. |  |