1. Problem analysis and project planning

1.1 Introduction

Student mark analyzing system has been designed to carry out the mark analysis process in an educational institution. The results of respective departments can be efficiently computed without much of manual involvement.

1.2 Objectives

The purpose of this document is to define the requirements of mark analysis system. This system reduces manual work to great extent. The mark analysis is carried out by the system in an efficient manner.

1.3 Scope

This system is very essential for every educational institution as it reduces man power. This system can be used for all kinds of educational institutions to evaluate and analyze the marks and generate reports of specified criteria.

1.4 Problem Statement

For analyzing the marks obtained by students in an educational institution. We are tasked to build up student mark analyzing system.

This is done to replace the manual entering and processing of marks which are error prone and tedious. This system also maintains information about student.

The system will have a Windows based desktop interface to allow the faculty to enter marks obtained by the students, update them and generate various reports.

For security reasons, the administrator and faculty only can update the marks and other information. First the user needs to login to the system for accessing it. The system will retain information on all the students and the institution. The system analyses the marks and generate
the result reports. The marks and information about the students are stored in a database and the system works with the database.

The faculty can enter the marks and student information through a visual environment. The updated details are stored in the database. The system generates the overall result by analyzing the marks. Mark analyzer monitors this process. The application in run by the mark analyzer. The trial for illegal updation would render the system to be locked.

One of the most important features of the system is creating reports based on the given criteria. The user can create the following reports:

Overall Class, Department result, Individual student result, Toppers list, Arrears list and Improvement rate for the academic year report has to be generated by entering the register number of the student. These reports can also be viewed by the management and placement officers. The administrator is responsible for adding, deleting student details form the system and updating the marks to the system with the external queries. So, the system design will generate reports automatically and there will be no need for manual intervention.

2. Problem statement(Use case)analysis

2.1 Identified use cases

i  Login:

This use case describes how a user logs in to the mark analyzing system.

ii Marks entry:

This use case allows faculty to enter the marks into the student table.

iii Mark analysis:

This use case describes how the system generates the overall result by analyzing the marks.
iv Maintain student information:

This use case allows the administrator to maintain the student information and it also includes adding, changing and deletion of information about the students from the system.

v Create result report:

This use case allows the system to generate various reports based on the criteria specified by the user.

2.2 Identified Actors

i Faculty:

The person responsible for entering and updating the marks obtained by the students.

ii Administrator:

The person responsible for maintaining student information in the system.

iii Database:

The database that contains all the information about the student and their marks.

iv Mark analyzer:

The person responsible for monitoring the mark analyzing process.

v Student:

Details about the students are entered into the system so that the student can view the results and reports.

vi Placement Officer:

The placement officers can also view the reports based on the criteria specified.
2.3 Use Case Diagram

- Faculty
- Administrator
- Database
- Mark Analyser
- Student
- Placement Officer

- Login
- Mark Entry
- Mark Analysis
- Maintain Student Information
- Result Report
- View Reports
3. Design of Students Mark analyzing System

3.1 Design Documentation

1. Login

1.1 Brief description:

This use case describes how the user logs into the Marks Analyzing System (MAS).

1.2 Flow of events:

1.2.1 Basic flow:

This use case starts with the actor wishes to log into the MAS.

1. The system requests the actor to enter their name and password.
2. The actor enters their name and password.
3. The system validates the entire name and password and logs the actor into the system.

1.2.2 Alternative flow:

1. Invalid name and password.
2. If in Basic flow the actor enters invalid name and password, the system displays an error message. The actor can choose to either return to the beginning of basic flow or cancel the login at which point the use case ends.

1.3 Pre conditions:

None.

1.4 Post conditions:

If the use case was successful, the actor is now logged into the system, if not, the system status unchanged.
2. Marks Entry

2.1 Brief description:

This use case allows the faculty to enter the marks into the student table.

2.2 Flow of events:

2.2.1 Basic flow:

This use case starts when the faculty wishes to enter the marks obtained by the student in different subjects.

1. The system retrieves and displays the student table. If a student table does not exist, it creates a new one. The names of the student and reg. no can’t be changed by the faculty.
2. Once the faculty has entered marks, the system saves the table and adds it to the database.

2.2.2 Alternative flow:

i. Invalid Marks:

In Basic flow, if invalid marks are entered, the system displays an error message and prompts for a valid mark. The faculty must enter a valid mark or cancel the operation in which case, the use case ends.

ii. Marks already entered:

If in basic flow, the student mark has already been entered, the system displays the read only copy of marks and informs the faculty that the mark has already been entered. So, no changes can be made to it. The faculty acknowledges the message and the use case ends.

iii. Fields left empty:

If in basic flow, the field is left empty, the system prompts the faculty to correct the error. The faculty can enter the mark or mark the student as absent.
2.3 **Pre Condition:**

The faculty must be logged on to the system before the use case begins.

2.4 **Post Condition:**

If the use case was successful, the student mark is saved to system otherwise the system status is unchanged.

3. **Mark Analysis**

3.1 **Brief description:**

This use case describes how the system generates overall results by analyzing the marks, Marks Analyzer monitors this process.

3.2 **Flow of events:**

3.2.1 **Basic flow:**

This use case begins when the mark analyzer wishes to calculate the total percentage of marks obtained by the students.

1. The system retrieves and displays the current student marks information from the database.
2. The system calculates the total marks and percentage obtained by all the students.
3. The results are stored in the database.
4. The use case ends when all the student’s marks have been processed.

3.2.2 **Alternative flow:**

i. **Marks unavailable:**

If in basic flow, the information about student marks could not be located, the system displays error message and use case ends.

ii. **Results already calculated:**

If in basic flow, the result has already been calculated, the system displays the copy of the information and informs mark analyzer that
marks have already been processed. The mark analyzer acknowledges the message and the use case ends.

3.3 **Pre Condition:**

The mark analyzer must be logged on to the system before this use case begins.

3.4 **Post Condition:**

If the use case was successful, the processed mark information is saved to the system otherwise the system status is unchanged.

4. **Maintain Student Information**

4.1 **Brief description:**

This use case allows the administrator to maintain the student information. This includes adding, changing and deleting information from the system.

4.2 **Flow of events:**

4.2.1 **Basic flow:**

This use case starts when the administrator wishes to change, add or delete student information from the system.

1. The system requests the administrator to specify the function that the administrator would like to perform.
2. Once the administrator provides the requested information, one of the sub flows is executed. If administrator selects add a student, then the ‘add a student’ sub flow is executed. If administrator selects update student information, then the ‘update student information’ sub flow is executed. If the administrator selects delete a student, then the ‘delete a student’ sub flow is executed.
Add a Student:

1. The System requests the administrator to enter the student information. This includes name, department, year, semester, age and sex.

2. Once the administrator provides the requested information, the system generates and assigns a unique register number to the student. Now the student is added to the system database.

3. The system provides the administrator to write the new register number.

Update Student Information(USI):

1. The System requests the administrator to enter the register number.
2. Administrator enters the register number and then the system retrieves and displays the student mark information.
3. Administrator can make changes to the marks.
4. The System updates the student information to the database.

Delete a Student:

1. The System requests the administrator to enter the register number.
2. Administrator enters the register number and then the system retrieves and displays the information.
3. The System prompts administrator to confirm deletion of the student.
4. Administrator verifies the deletion.
5. The System deletes the student from the database.

4.2.2 Alternative flow:

i. Student not found:

If the Update Student Information(USI) or delete a student sub flow student with specific register number does not exist, then the System displays a error message.
The System administrator can re enter or cancel at which point the us case ends.

**ii. Irrelevant data:**

If in the add a student sub flow invalid information is entered, the system display an error message so that the administrator can re enter or cancel.

**iii. Delete Cancelled:**

If in the delete a student sub flow, the administrator decides not to delete the student, the delete is cancelled and the Basic flow is restarted.

**iv. Pre Condition:**

The administrator must be logged on to the system before this use case begins.

**v. Post Condition:**

If the use case was successful, the student information is added, updated or deleted.

5. **Create result/report**

**5.1 Brief description:**

This use case allows the user to create an overall class or department result. The individual student result, toppers list, arrears list and improvement rate for the academic year report is discussed.

**5.2 Flow of events:**

5.2.1 **Basic flow:**

This use case starts with the actor user wishes to create a report.

1. The system requests the user to specify the following report criteria:
   - Report type (either “Overall class/Department result”, ”Individual Student Result”, ”Toppers List”, ”Arrears List” and “Improvement Rate for the academic year”).
   - Criteria for the respective report.
2. If the user selects the “Overall class/department result” report, the system retrieves and displays the entire Student’s mark.

3. The system then requests the user to enter information he/she requires (criteria).

5. If the user selects “Individual Student Result” report, the system requests the student to enter the register number. The system validates the register number and if it is valid, the displays the report.

8. Similarly, for the “Toppers list”, “Arrears list”, “Improvement Rate” reports, the criteria are specified.

9. Once the user provides the requested information the System provides the report satisfying the report criteria.

10. The user may require saving the report.

5.2.2 Alternative flow:

Requested Information Unavailable:

If in the basic flow, the requested information is unavailable, the system will display an error message. The user can choose return to begin the basic flow or cancel it at which the use case ends.

Invalid Format or Insufficient Information:

If in the basic flow, the user has not specified sufficient information to create the report, the system will prompt the user for missing information. The user can re-enter or cancel, at which point the use case ends.
Invalid register number:

If the user enters invalid register number, the system will display an error message. The user can re-enter or cancel the operation.

5.3 Pre condition:

The user must be logged on to the system before the use case begins.

5.4 Post condition:

The System state is unchanged by the use case.
1. Login

1. User enters main form()
2. Select to login()
3. System requests username and password()
4. User enters username and password()
5. Validate username and password()
6. Refer from database()
7. Validate username and password()
8. Welcome message generate()
9. Invalid username and password()
10. Error message - relogin or cancel()
2. Marks Entry

1: Request to specify operation()

2: Enter operation (mark entry)

3: Enter mark entry form()

4: Refer from Database()

5: Request to enter marks()

6: Faculty enter marks()

7: Valid marks()

8: Valid (store in database)

9: Invalid marks()

10: Request to restart/cancel()
3. Mark Analysis

1: Request to enter the operation()  
2: Processing marks()  
3: Enters into the form()  
4: Refer student information from database()  
5: Calculation of total, Average()  
6: Valid modified table into the database()  
7: Invalid incorrect marks entry form  
8: Restart the operation()

4. Maintain Student Information

1: Request to select an option()  
2: Allows the administrator to select an option()  
3: Add option()  
4: Update option()  
5: Delete option()
4.1 Add a student

1: Request to enter name, dept, addr
2: Enter details
3: Validate Details
4: Validate generates a new reg no and add it to the database
5: Invalid or details missing
6: Displays Error message, request to re-enter or cancel

4.2 Update Student Information

1: Request to enter reg no
2: Administrator enter reg no
3: Validate reg no
4: Validate system returns corresponding record
5: Request to update form
6: Update marks
7: Invalid updation
8: Request to re-enter or cancel
4.3 Delete a student

---

1: Request to enter reg no()
2: Enter reg no()
3: Validate reg no()
4: Refer from database()
5: Valid reg no student deleted()
6: Invalid reg no
7: Re-enter reg no or cancel()
5. Create Result/Report

1: Request to enter report type
2: Enter report type
3: Validate type
4: Refer from database
5: Request to enter the criteria
6: Enter the criteria
7: Validate the criteria
8: Valid display report
9: Store to database
10: Invalid criteria
11: Invalid type
12: Restart or cancel
COLLABORATION DIAGRAM

1. Login

1: User enter mainform()
4: User enters username and password()
8: Welcome message generate()

Student

2: Select to login()
3: System request username and password()
5: Validate username and password()

Faculty

3: Refer from database()

6: Refer from Database()

7: Valid username and password()
9: Invalid username and password()
10: Error message - relogin or cancel()

2. Marks Entry

1: Request to specify operation()

Faculty

2: Enter operation(mark entry)
6: Faculty enter marks()

Entry Controller

3: Enter mark entry form()
5: Request to enter marks()
7: Valid marks()
10: Request to restart/cancel()

Marks Entry Form

4: Refer from Database()

8: Valid(store in database)

Error Message

9: Invalid marks()

: Database
3. Mark Analysis

4. Maintain Student Information

4.1. Add Student
4.2. Update Student

2: Administrator enter reg no()

3: Validate reg no

6: Update marks()

1: Request to enter reg no()

5: Request to update form()

4: Validate system returns corresponding record()

8: Request to re-enter or cancel()

4.3. Delete Student

2: Enter reg no()

3: Validate reg no()

6: Invalid reg no

1: Request to enter reg no()

4: Refer from database()

5: Valid reg no student deleted()

7: Re-enter reg no or cancel()
5. Create Result/Report

**STATECHART DIAGRAM**

1. Login

```
[Valid Details] / Enter into Welcome Screen [Invalid details] / Error message

Initialization

Validate Details

Open

[Valid Details] / Enter report type

5: Request to enter criteria

3: Validate report type

2: Enter report type

6: Enter the criteria

1: Request to enter report type

4: Refer from database

12: Restart or cancel

9: Store to database

8: Valid display report

10: Invalid criteria

7: Validate the criteria

: Administrator

: Database

Error message

Report controller

Report form

Main form

Error message
```
2. Mark Analysis

ACTIVITY DIAGRAM

1. Updating the Database
2. Adding a Student

- Request to enter the details of the student to be added.

  - Process the details
    - valid details

  - Generate register no. for the student

  - add the student to database

  - Confirmation that student have been added
3. Deleting a Student

Administrator

Request to enter reg.no

System

Process the register no.
check reg.no

Database

Confirmation that student have been deleted

register no. available

Delete

Update database

Database System Administrator
1. Login

CLASS DIAGRAM

Main form

enter operation()

Error message

• Invalid marks()
• Display error message()

Login form

• name
• password

• Username and password()
• login()
• select to login()

Welcome form

• Display to user()
• valid username and password()

Entry Controller

• Refer from database()
• Validate marks()

Login Controller

• Refer from database()
• Validate username and password()

Welcome form

• Display to user()
• valid username and password()

Login Controller

• Refer from database()
• Validate username and password()

Database

2. Marks Entry

CLASS DIAGRAM

Main form

enter operation()

Marks Entry form

• display mark entry form()
• faculty enters marks()
• Refer from database()

Entry Controller

• Validate marks()
• Refer from database()

Error message

• Invalid marks()
• Display error message()

Database

Main form

enter operation()
3. Mark Analysis

- **Mark analyser**
  - **Main form**: enter operation
  - **Error message**: Invalid marks, Display error message

**Student Information flow**
- retrieves information from database
- refers student information from database

- **Processing control**
  - Process the marks, calculation of total, average

- **Database**
- **Report Controller**
  - Validate the criteria
  - Ask whether to save or not
  - selects to save
  - selects not to save

4. Create Student Report

- **Main form**: enter operation
- **Error message**: Invalid marks, Display error message

- **Report form**
  - Request the user to enter the criteria
  - validates
  - validates type
  - retrieves information from database

- **Database**
5. Maintain Student Information

COMPONENT DIAGRAM
SOURCE CODE:
1) Login:

Option Explicit

Public NewProperty As login_form

Public Sub refer_to_database()

End Sub

Public Sub validate_user_name_and_pass_word()

End Sub

2) Mark entry:

Option Explicit

Implements main_form

Public Sub display_mark_entry_form()

End Sub

Public Sub faculty_enters_mark()

End Sub

Public Sub refer_from_data_base()

End Sub

3) Mark analysis:

Option Explicit

Public Sub process_the_marks()

End Sub

Public Sub calculatation_of_total_average()

End Sub
4) **Student report:**

Option Explicit

Public NewProperty As student

Public NewProperty2 As student

Public Sub enter_operation()

End Sub

5) **Maintain student information:**

Option Explicit

Public Sub add_option()

End Sub

Public Sub update_option()

End Sub

Public Sub delete_option()

End Sub