

Software Requirements Specification

Version 1.0

February 15, 2019

Diary Management System

Aayoush Dubey Team-leader

Harsh Agarwal

Apoorva Tiwari

Table of Contents

Table of Contents	1
List of Figures	2
1.0. Introduction	3
1.1. Purpose	3
1.2. Scope of Project	3
1.3. Glossary	3
1.4. References	3
1.5. Overview of Document	3
2.0. Overall Description	4
2.1. System Environment	4
2.2. Functional Requirements Specification	4
2.2.1. Student Use Case	5
Use case: Student	5
2.2.2. Guide Use Case	6
Use case: Guide	6
2.2.3. Admin Use Case	7
Use case : Admin	7
2.3. User Characteristics	8
2.4. Non-Functional Requirements	8
3.0. Requirements Specification	8
3.1. External Interface Requirements	8
3.2. Functional Requirements	8
3.2.1. Login	9
3.2.2. Data Entry	9
3.2.3. Upload File	10
3.2.4. Comment	10
3.2.5. Review	11
3.2.6. Enter Marks	11
3.2.7. View Marks	12
3.2.8. Add User	12
3.3. Detailed Non-Functional Requirements	13
3.3.1 Logical Structure of the Data	13
3.3.2 Security	16

List of Figures

Figure 1 - System Environment	4
Figure 2 - Student Use Case Diagram	5
Figure 3 - Guide Use Case Diagram	6
Figure 4 - Admin Use Case Diagram	7
Figure 5 - Logical Structure of the Diary Manager Data	13

1.0. Introduction

1.1.Purpose

Diary Manager is a simple application designed to take important notes for important upcoming events and for your past records. In this project, you can simply add your notes and leave it for future reminders. We have developed this app as open source to make it available for your assignment/projects.

1.2. Scope of Project

This product helps the user to set reminders, keep a track of things that he pre-plans and even set his goals. A digital diary also reduces the need to carry a physical diary around and one can write whenever they feel like and the data uploaded can also be reviewed by your mentor/guide for assessment purposes.

1.3. Glossary

Term	Definition
Admin	A User who can add users,upload files and enter marks.
Student	A sub User who can enter data upload files and view marks
Guide	A sub User who can upload file,comment,enter marks

1.4. References

1.5. Overview of Document

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

The third chapter, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product.

Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

2.0. Overall Description

2.1. System Environment

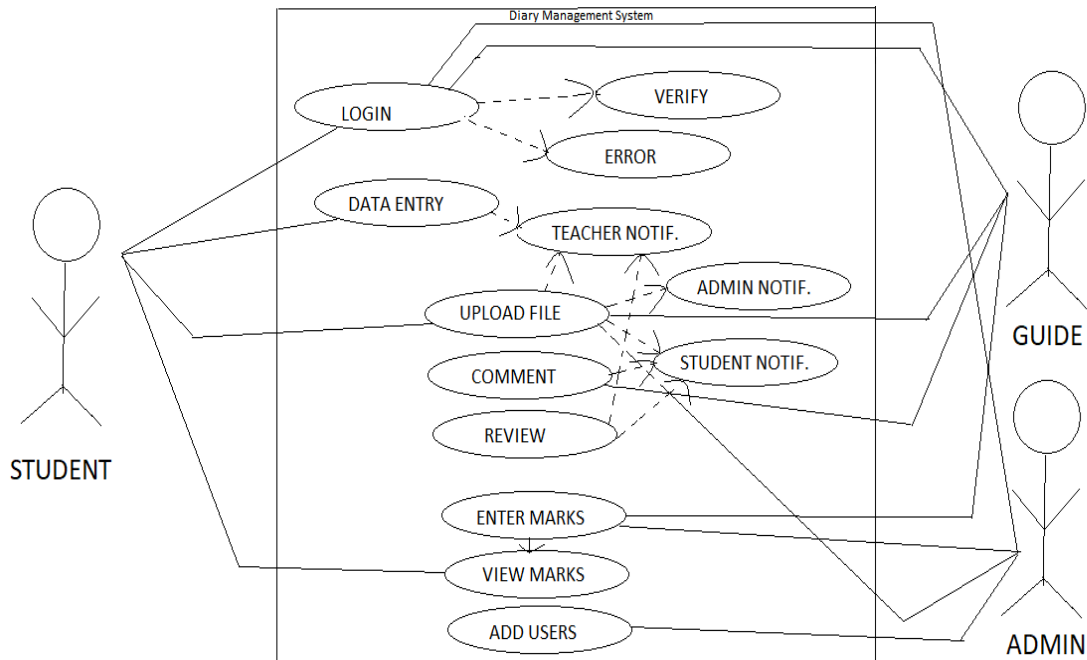


Figure 1 - System Environment

The Dairy Managemet App has three main active Actors. The Student, Guide and the Admin. The Student basically logs into the application and uploads a file about his weekly activity on his particular project at the end of each week the Guide can log into the application and review this file uploaded by the student and comment or enter marks on the same accordingly.

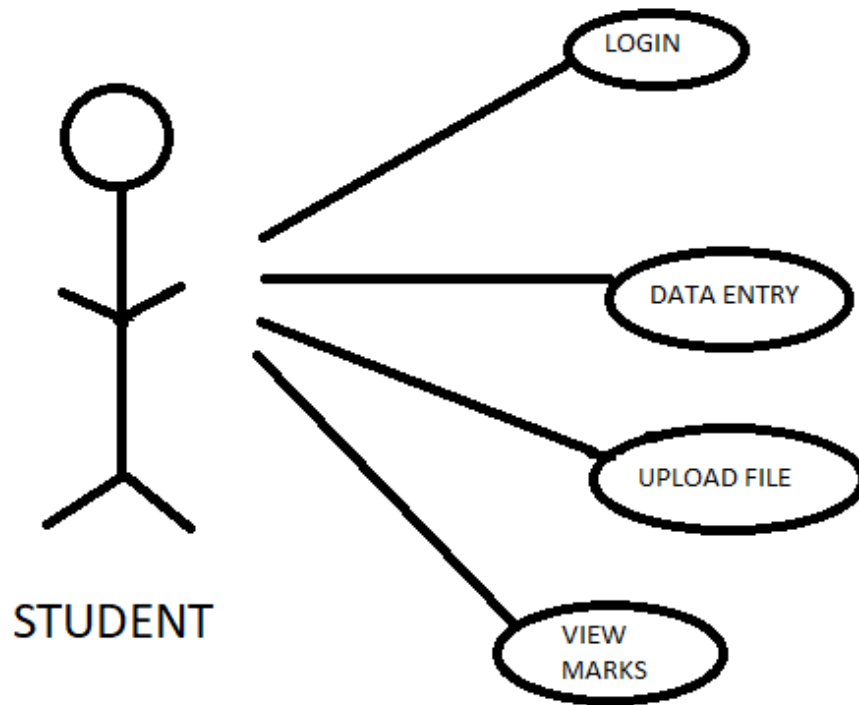
2.2. Functional Requirements Specification

This section outlines the use cases for each of the active Student separately. The student, the Guide and the Admin have only one use case apiece while the Student is main actor in this system.

2.2.1. Student Use Case

Use case: Student

Diagram:



Brief Description

The Student accesses his records(dairy) by logging into the application by entering his credentials. He then enters data at least once a week . He can view the files uploaded by the Admin and Teacher . He can also view his marks .

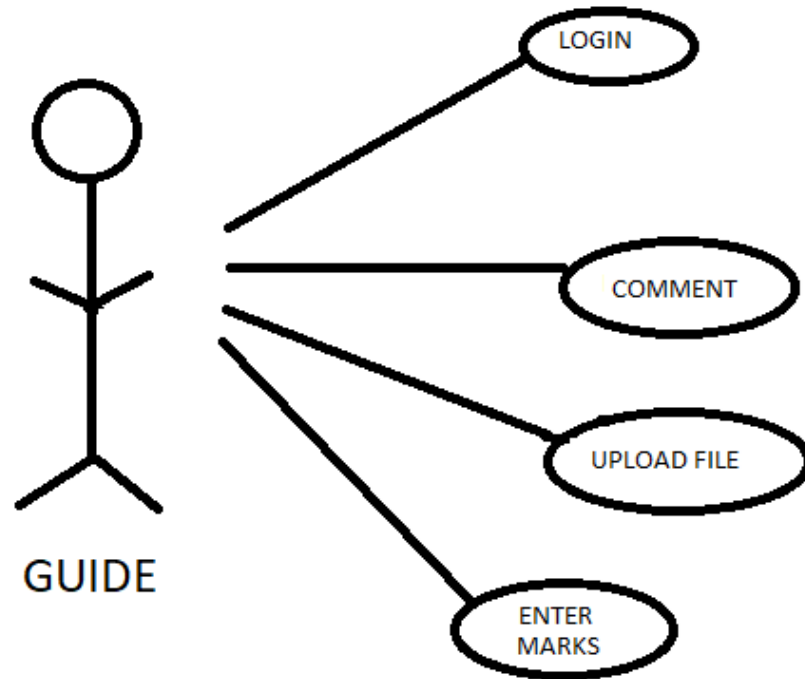
Initial Step-By-Step Description

The Student will log into the application by entering his credentials.
The application will verify or will show error if the entered credentials are incorrect.
Enters data i.e. weekly status of his project
Uploads his file at the end of each week.
He can also view his marks.

2.2.2. Guide Use Case

Use case: Guide

Diagram:



Brief Description

The Guide accesses the student's records(dairy) by logging into the application by entering his credentials. He then reads the students entry and enter comments least once a week. He can view the files uploaded by the Admin and Student and also upload files . He also uploads marks.

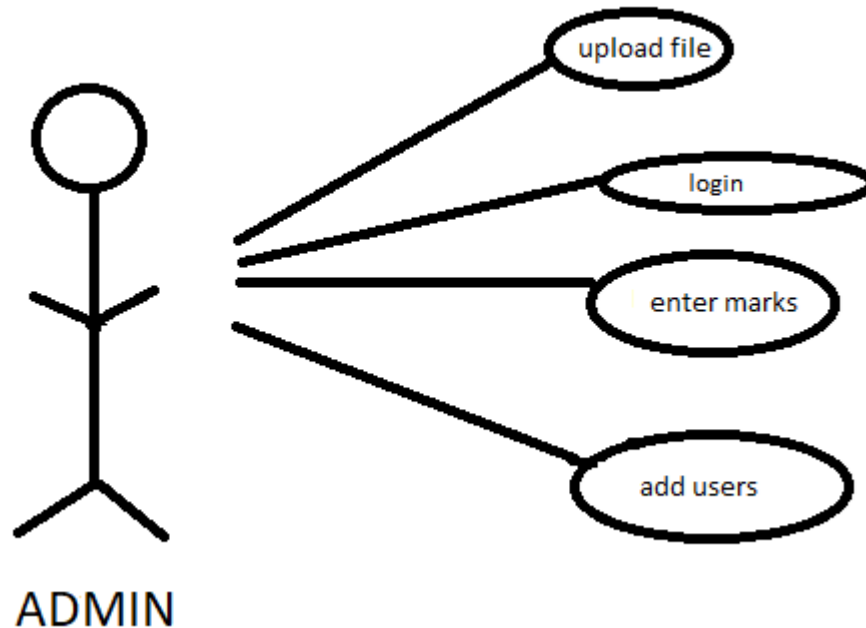
Initial Step-By-Step Description

The Guide can log into the application by entering his credentials.
He can upload files for student reference purpose.
He can review the file uploaded by the student and comment.
He can enter marks on the same file which was uploaded by the student.

2.2.3. Admin Use Case

Use case: Admin

Diagram:



Brief Description

The Admin accesses the system by logging into the application by entering his credentials. He then reads the students entry and Reviews it at least once a week . He can view the files uploaded by the Teacher and Student and also upload files . He also uploads marks . The Admin can add new users who can access the system .

Initial Step-By-Step Description

The Admin will login into the system by entering his credentials.
The Admin will add users who will be able to access software .
The Admin can upload files .
The Admin will review the files and mark the students accordingly .

2.3. User Characteristics

The Student is expected to be Computer literate and be able to use a Computer Application . The main screen of the Diary Management System will have a module to make the data entry .The Teacher are expected to be Computer literate and to be able to use Computer Applications . The main screen will have a module to access the data entered my the student .The Admin is expected to be Computer literate and should be able to use button, pull-down menus, and similar tools.

The detailed look of these pages is discussed in section 3.2 below.

2.4. Non-Functional Requirements

The Diary Management System will be a standalone application and requires a computer. The software developed here assumes the use of various tool for connection between the Application and the database. The working of the system will depend on the hardware used rather than characteristics of this system.

3.0. Requirements Specification

3.1. External Interface Requirements

The only link to an external system is the link to the Database to verify the Actor. The Editor believes that a society member is much more likely to be an effective reviewer and has imposed a membership requirement for a Reviewer. The HS Database fields of interest to the Web Publishing Systems are member's name, membership (ID) number, and email address (an optional field for the HS Database).

The Assign Reviewer use case sends the Reviewer ID to the HS Database and a Boolean is returned denoting membership status. The Update Reviewer use case requests a list of member names, membership numbers and (optional) email addresses when adding a new Reviewer. It returns a Boolean for membership status when updating a Reviewer.

3.2 Functional Requirements

The Logical Structure of the Data is contained in Section 3.3.1.

3.2.1. Login:

Use Case Name	Login
XRef	Section 2.2.1, Login
Trigger	All Actors assesses the Diary Management System
Precondition	The Application is already installed in the system and the Actor is added in the database by the Admin.
Basic Path	The Actors open the system and entire their credentials . If the Admin has added the actor as a user he will be able to access the system otherwise an error will be shown .
Alternative Paths	No alternate path
Postcondition	The Actor is logged into the system .
Exception Paths	The Actor may leave the login page anytime .
Other	None

3.2.2. Data Entry:

Use Case Name	Data Entry
XRef	Section 2.2.1 Student Data Entry;
Trigger	The student enters data to be viewed by the teacher.
Precondition	The student has to be logged into the system
Basic Path	The student log's into the system The student enters the data The student submits the data
Alternative Paths	None
Postcondition	A notification is sent to the Teacher
Exception Paths	The student may not submit the document .
Other	None

3.2.3. Upload File:

Use Case Name	Upload File
XRef	Section 2.2.1, Upload File; Section 2.2.2, Upload File
Trigger	Any Actor can upload a file .
Precondition	The Actor must be Logged into the system .
Basic Path	The Actor log's into the system The Actor uploads the file The Actor submits the file
Alternative Paths	None
Post condition	A notification is sent to all the Actors .
Exception Paths	The Actor may abandon the operation at any time.
Other	None

3.2.4. Comment:

Use Case Name	Comment
XRef	Section 2.2.2. Guide Comment
Trigger	The Teacher on reading the document writes a comment.
Precondition	The Teacher has to be logged into the system and there should be a data entry by the student .
Basic Path	The Teacher log's into the system The Teacher reads the students file and comments on it . The Teacher submits the Comment
Alternative Paths	none
Postcondition	A notification is sent to the Student
Exception Paths	The Teacher may abandon the operation at any time.
Other	none

3.2.5 Review:

Use Case Name	Review
XRef	Sec 2.2.3 Guide Review; Sec 2.2.3 Admin Review
Trigger	The Admin on reading the document , reviews it .
Precondition	The Admin has to be logged into the system and a data should have been entered by the Student and the Teacher should have commented on it .
Basic Path	The Admin log's into the system The Admin reads the students file and reviews it . The Admin submits the Review.
Alternative Paths	none
Postcondition	A notification is sent to the student and the teacher .
Exception Paths	The Admin may abandon the operation at any time.
Other	None

3.2.6 Enter Marks:

Use Case Name	Enter Marks
XRef	Section 2.2.2, Update marks; Section 2.2.3, Update marks
Trigger	The Teacher and the Admin reads the data uploaded by the student and give marks accordingly .
Precondition	The Admin and the Teacher has to be logged into the system and a data should have been entered by the Student and the Teacher should have commented on it which should have also been reviewed by the Admin .
Basic Path	The Admin/Teacher log's into the system The Admin/Teacher reads the students file and marks it . The Admin/Teacher submits the marks .
Alternative Paths	none
Postcondition	The marks can be viewed by the student .
Exception Paths	The Admin/Teacher may abandon the operation at any time.

3.2.7. View Marks:

Use Case Name	View Marks
XRef	Section 2.2.2 View marks
Trigger	The Student opens this module to view his marks .
Precondition	The student has to be logged into the system and the marks has to be uploaded by the Admin / Teacher
Basic Path	The student log's into the system The student Views his marks
Alternative Paths	None
Postcondition	none
Exception Paths	none
Other	none

3.2.8. Add User:

Use Case Name	Add User
XRef	Section 2.2.3, Admin Reviewer
Trigger	The Admin wants to add a new user
Precondition	The Admin has to be logged into the system .
Basic Path	The Admins log's into the system The Admin enters details of the New User
Alternative Paths	None.
Postcondition	A new user has been added to the database and he/she can access the application
Exception Paths	The Admin may abandon the operation at any time.
Other	none

3.3. Detailed Non-Functional Requirements

3.3.1. Logical Structure of the Data Entry

The logical structure of the data to be stored in the internal Diary Manager database is given below.

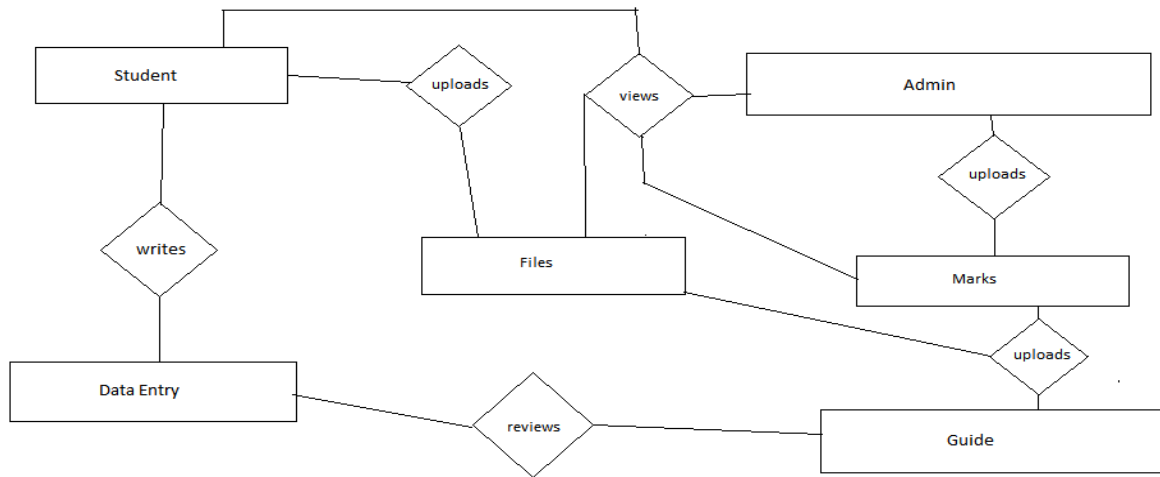


Figure 4 - Logical Structure of the Diary Manager Data

The data descriptions of each of these data entities is as follows:

Student Data Entity:

Data Item	Type	Description	Comment
Name	Text	Name of Student	
Registration no	Integer	Registration no of student	
Data Entry	Text	Progress entity	At least Once a week
Files	Text	Report, study material can be submitted	

Guide Data Entity:

Data Item	Type	Description	Comment
Name	Text	Name of Guide	
ID	Integer	ID number of Guide	Used as key in marks
Review	Text	Comment on the data entry by student	May be several
Enter marks	Integer	Marks on basis of progress	

Admin Data Entity:

Data Item	Type	Description	Comment
Name	Text	Name of admin	
ID	Integer	ID of admin	
Enter Marks	Integer	Marks obtained	
View Files	Text	Report given by student	

The Logical Structure of the data to be stored in the Online Diary database on the server is as follows:

Data Entry Data Entity:

Data Item	Type	Description	Comment
Name	Text	Name of Entry	
Student ID	Integer	Registration no of student	Key in student
Guide ID	Integer	Registration No of Guide	Key in Guide
Date	Date	Date of entry made	
Review	Text	Comments by Guide	Contains paragraph.
Admin ID	Integer	ID of admin	Key in Admin
Submit	Boolean	Entry has been accepted	

Marks Data Entity:

Data Item	Type	Description	Comment
Marks by Guide	Integer	Marks given by guide	
Student ID	Integer	Registration No of Student	To view marks
Guide ID	Integer	Registration No of Guide	Key in Guide
Admin ID	Integer	Registration No of Guide	
Marks By Admin	Integer	Marks Given By Admin	

Files Data Entity:

Data Item	Type	Description	Comment
Student file	Text	File uploaded by student	
Guide file	Text	File uploaded by Guide	
Marks By Admin	Integer	Marks Given By Admin	

3.3.2. Security

The server on which Online Diary resides will have its own security to prevent unauthorized write/delete access. There is no restriction on read access.

Only the Admin will have physical access to the machine and the program on it. There is no special protection built into this system other than to provide the Admin with write access to the Online Diary to upload marks and also by the guide For diary data entries.