Final Project

Submitted By: **Muhammad Awais Khan (12860)**

Subject : Software Quality Engineering

Section : 1

Submitted To: **Sir Rizwan Faiz**

Table of Contents

HOSTEL MANAGEMENT SYSTEM

Functional Requirements:

FR 1 The System will allow admin to login

FR 2 The system will allow Admin to register students

FR 3 The system should allow Admin to change password using two factor authentication.

FR 4 System shall allow the user to access the system by entering the username of 5 to 10 characters

FR 5 System shall allow the user to access the system by entering the password of 8 to 15 characters

FR 6 System shall allow admin to add student details. Student name must be in capital alphabets.

FR 7 The System will allow Admin to edit students

FR 8 The System will allow Admin to delete students

FR 9 The System will allow Admin to search students

FR 10 Admin will be able to update student records

FR 11 Admin can see student's fee details by entering sap_id

Defect Prevention

Defect Prevention upon SRS:

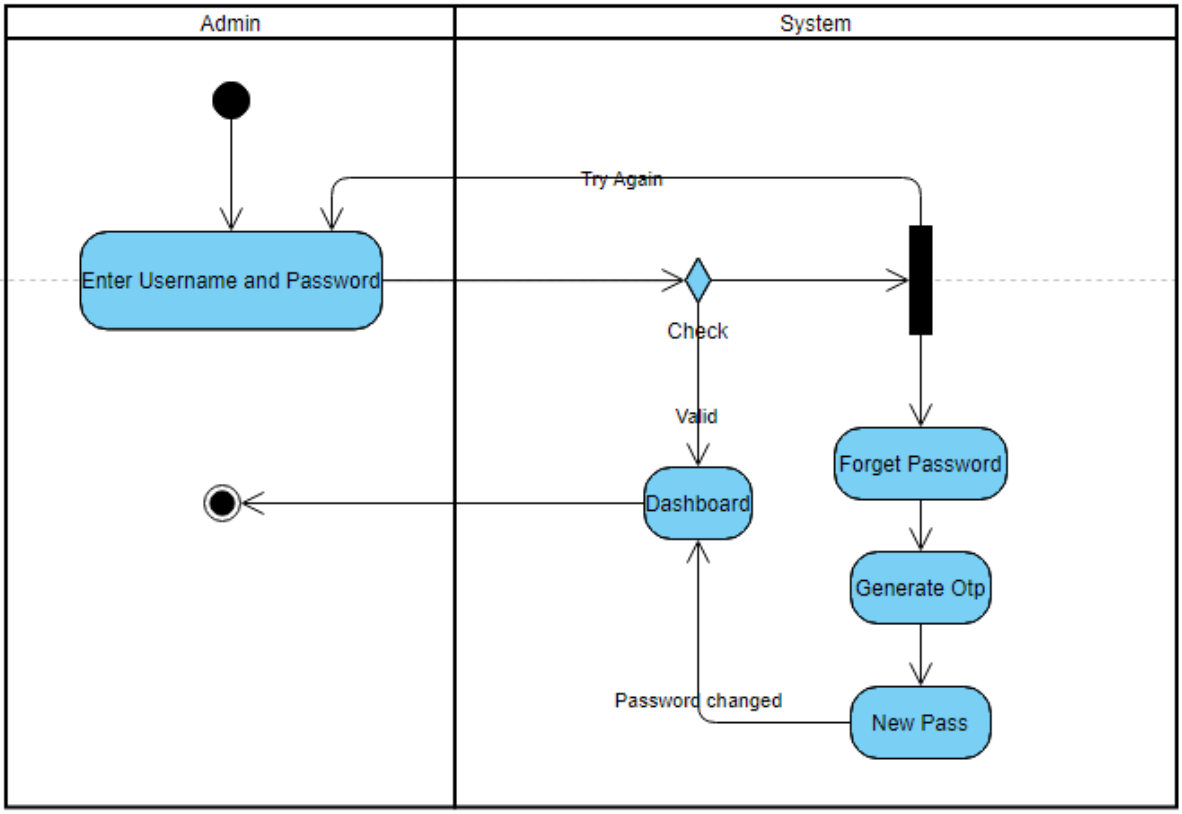
Goals	Sub Goals	Requirement	Defect	Defect Prevention Technique
Security	Authenticity	The System will allow admin to login	Denial of Unauthorized Access	Forcing Function
Security	Authenticity	The system	Duplication of	Forcing

		will allow Admin to register students	data	Function
Security	Authenticity	The system should allow Admin to change password using two factor authentication.	Denial of Unauthorized Access	Redundancy
Security	Integrity	The System will allow Admin to edit students	Denial of Access to data modification	Forcing Function
Security	Integrity	The System will allow Admin to delete students	Denial of Access to data modification	Forcing Function
Functionality	Accuracy	The System will allow Admin to search students	Student does not exists	Automation and Computerization
Functionality	Accuracy	Admin can see student's fee details by entering sap_id	Student does not exists	Automation and Computerization

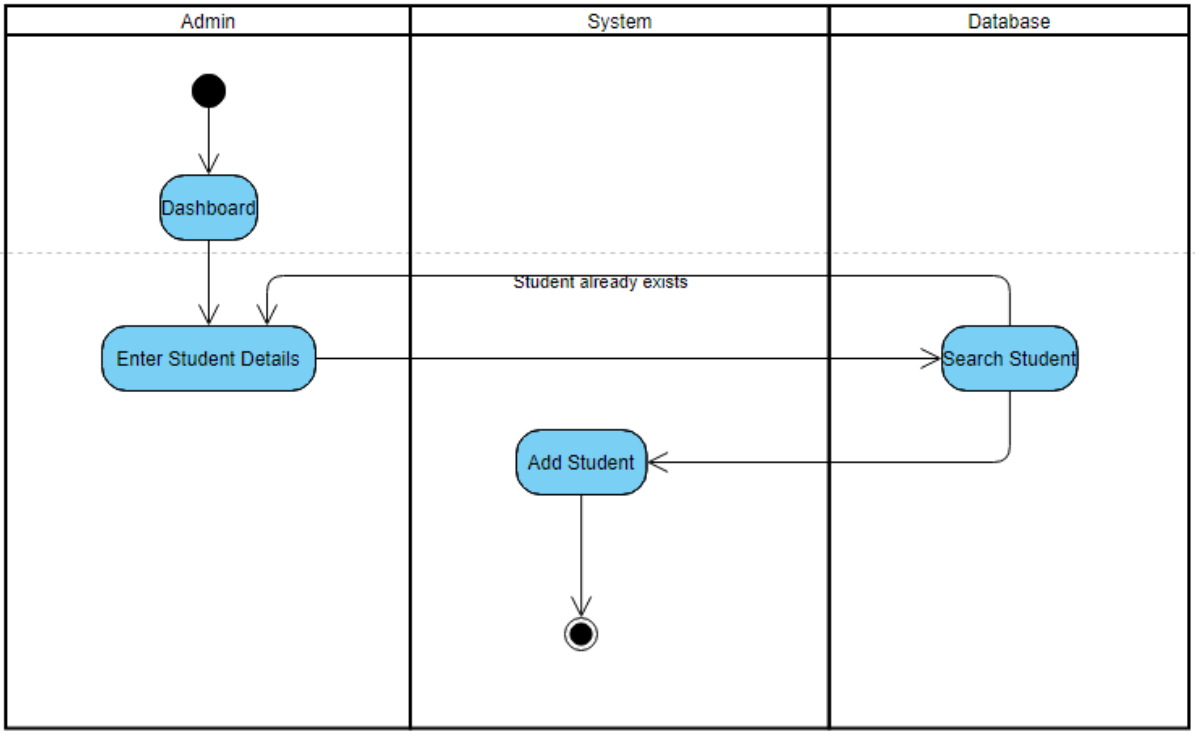
Defect Prevention upon Use Case:

FR 1 The System will allow admin to login

FR 3 The system should allow Admin to change password using two factor authentication.



FR 2 The system will allow Admin to register students



Defect Prevention upon Source Code:

FR 1 The System will allow admin to login

```
login_times = 3;

    while(login_times > 0){

      cout << "\n\n";

      cout << "Login"<<endl<<"_____";

      cout << "\n";

      cout << "login: ";

      cin >> login;

      cout << "Password: ";

      cin >> password_login;

      if(login == user && password_login == password){
```

```
 cout << "loged in Successfully!"<<endl;
 cout << "Welcome"<<user<<"!";
 break;
}
else if(login != user && password_login == password){
 cout << "username is incorrect!"<<endl;
 login_times--;
}
else if(login == user && password_login != password){
 cout << "password is incorrect!"<<endl;
 login_times--;
}
else{
 cout << "Everything is incorrect!"<<endl;
 login_times--;
}
```

FR 3 The system should allow Admin to change password using two factor authentication.

```
while(true)
    {
        auto t = time(NULL);
        auto left = time_step - (t % time_step);
        char otp[digits + 1] = {};
        result = oath_totp_generate(
            secret,
```

```
                sizeof(secret),
                t,
                time_step,
                OATH_TOTP_DEFAULT_START_TIME,
                digits,
                otp);
        if(result != OATH_OK)
        {
                cerr << oath_strerror(result) << endl;
                return -1;
        }
        cout << "OTP: " << otp << " (" << left << ") \r";
        cout.flush();
        this_thread::sleep_for(1s);
}
```

Defect Detection

Defect Detection Through Checklist Upon SRS:

Requirements	Checklist	Attribute	Defect
The system will allow Admin to register students	Are all the inputs to the system specified including their source, accuracy, range of values, and frequency?	Completeness	Admin will enter complete student details like Name, Father name, Roll number, department e.t.c
The System will	Do all the	Consistency	Add, update and

allow admin to manage(update, delete) student's record	requirements avoid conflicts with other requirements?		delete are separate requirements.
The admin should be provided user friendly interface	Are the requirements testable?	Testability	How to measure user friendliness? Is there any tool to check whether a system is user friendly or not?
The password should not be longer than 15 characters.	Are all requirements clearly understandable?	Unambiguous	Not clear what the system is supposed to do when admin enters password longer than 15 characters
The System shall allow admin to edit student's record The system shall allow admin to update student's record	Is each requirement unique?	Consistency	Both requirements i.e edit and update have same meaning.
The system should print challan form in acceptable time frame	Are all requirements clearly understandable?	Unambiguous	What is acceptable time frame?

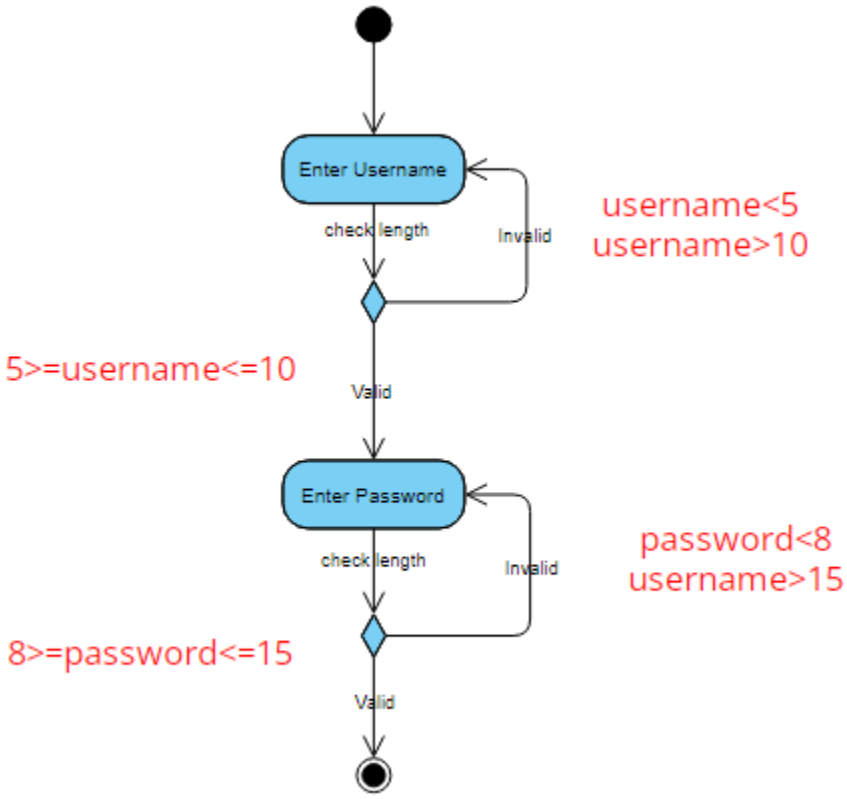
Defect Detection Through Test Case Upon SRS:

FR 4 System shall allow the user to access the system by entering the username of 5 to 10 characters

FR 5 System shall allow the user to access the system by entering the password of 8 to 15 characters

Use Case Name	User's Login
Primary Actor	Admin, Student
Description	In order to login the user should enter valid username and password length.
Pre-Condition	The user must be registered on the system
Post-Condition	User will be redirected to main dashboard
Basic Flow	1. User enters username 2. User enters password 3. User presses login button
Alternate Flow	4. Username length not valid, try again message is displayed 5. Password length not valid, try again try again message is displayed

Defect Detection Through Test Case Upon Use Case:



Scenerio 1: Valid Username → Test Case Id 1

Scenerio 2: Invalid Username → Test Case Id 2,3

Scenerio 3: Valid password → Test Case Id 4

Scenerio 4: Invalid password → Test Case Id 5,6

Test Data:

Input: Username

Data Type: Characters

Valid Class:

5<=Username<=10

Invalid Class:

Username<5

Username>10

T.C ID	Input	ECP	Expected Output
1	Username=6	5<=Username<=10	Valid
2	Username=3	Username<5	Invalid
3	Username=11	Username>10	Invalid

Test Data:

Input: Password

Data Type: Characters and numbers

Valid Class:

8<=password<=15

Invalid Class:

password <8

password >15

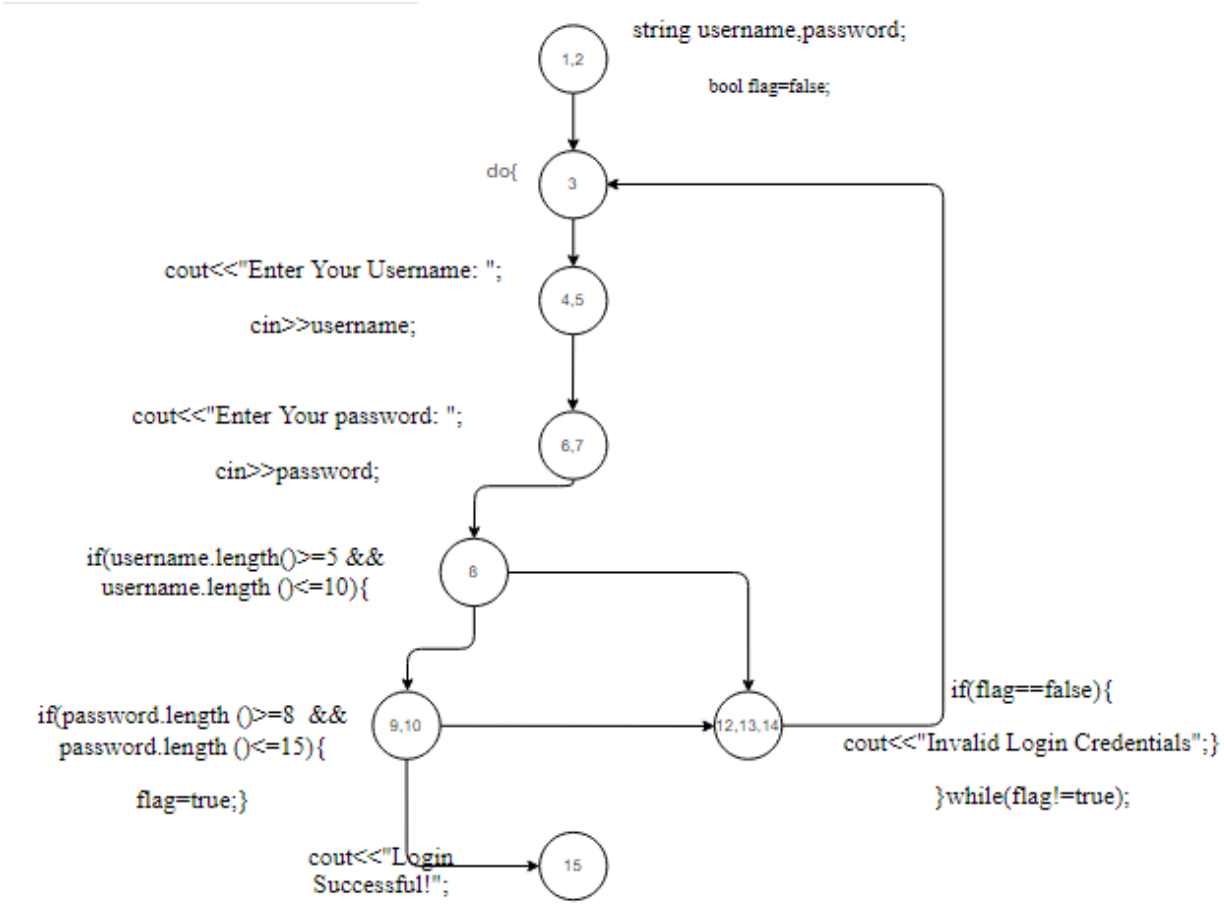
T.C ID	Input	ECP	Expected Output
4	password=9	8<=password<=15	Valid
5	password=7	password <8	Invalid
6	password=16	password >15	Invalid

Defect Detection Through Test Case Upon Code:

Code

```
1. string username,password;
2. bool flag=false;
3. do{
4. cout<<"Enter Your Username: ";
5. cin>>username;
6. cout<<"Enter Your password: ";
7. cin>>password;
8. if(username.length()>=5 && username.length ()<=10){
9. if(password.length ()>=8  && password.length ()<=15){
10.flag=true;}
11.}
12.if(flag==false){
13.cout<<"Invalid Login Credentials";
14.}
15.}while(flag!=true);
16.cout<<"Login Successful!";
```

Control Flow Diagram:



Test Data:

Input: Username,password

Data Type: Characters

Valid Class:

5<=Username<=10

8<=password<=15

Invalid Class:

Username<5

Username>10

password <8

password >15

T.C ID	Input	ECP	Actual Output
1	Username=6	5<=Username<=10	Login Successful!
2	Username=3	Username<5	Invalid Login Credentials
3	Username=11	Username>10	Invalid Login Credentials

T.C ID	Input	ECP	Actual Output
4	password=9	8<=password<=15	Login Successful!
5	password=7	password <8	Invalid Login Credentials
6	password=16	password >15	Invalid Login Credentials

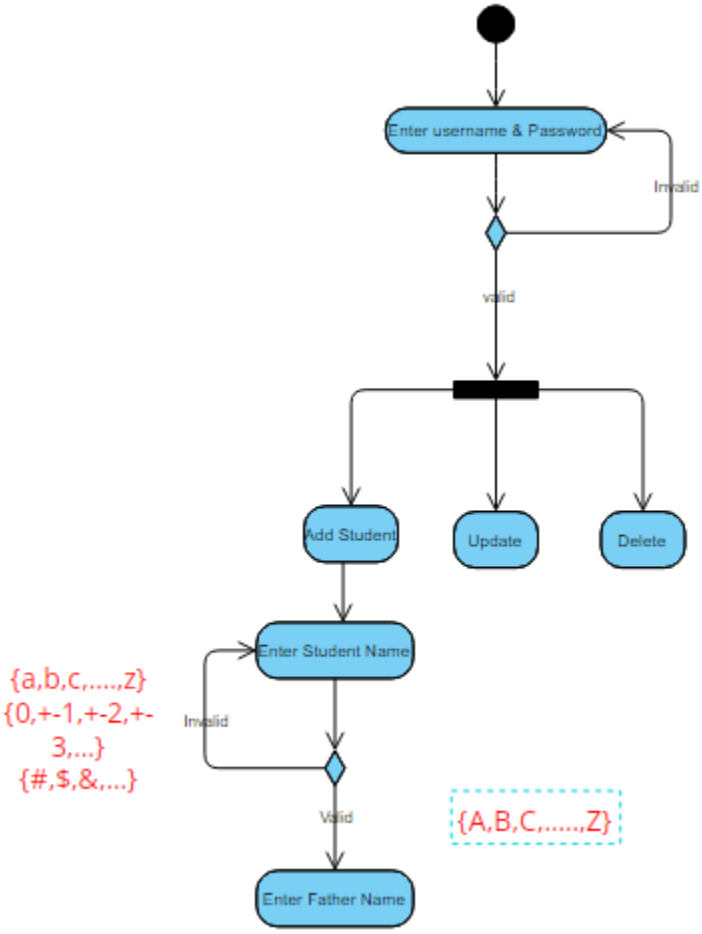
FR 8 System shall allow admin to add student details. Student name must be in capital alphabets.

SRS

Use Case Name	Add student detail
Primary Actor	Admin
Description	Admin can enter name of student in capital letters only.
Pre-Condition	The Admin must be logged in
Post-Condition	New student will be added on the system
Basic Flow	1. Admin logins 2. Admin opens add student page 3. Admin enters student name in capital letters
Alternate Flow	4. Admin enters digits in student name field, system displays error message

	5. Admin enters small letters in student name field, system displays error message 6. Admin enters special characters in student name field, system displays error message

UC



Test Data:

Input: StName

Data Type: Characters

Valid Class:

{A,B,C,.....,Z}

Invalid Class:

{a,b,c,.....z}

{0,+-1,+-2,..}

{!,#,$,.....}

T.C ID	Input	ECP	Expected Output
7	AWAIS	{A,B,C,.....,Z}	Valid
8	awais	{a,b,c,.....z}	Invalid
9	12345	{0,+-1,+-2,..}	Invalid
10	@i	{!,#,$,.....}	Invalid

Code

```
do{
cin>>stName;
if(isUpper(stName)){
cout<<"Student Name is Valid";
}
else{
cout<<"Error! Only Capital letters are allowed."
}
}while(!isUpper(stName))
```

Test Data:

Input: StName

Data Type: Characters

Valid Class:

{A,B,C,.....,Z}

Invalid Class:

{a,b,c,.....z}

{0,+-1,+-2,..}

{!,#,$,.....}

T.C ID	Input	ECP	Actual Output
7	AWAIS	{A,B,C,.....,Z}	Student Name is Valid
8	awais	{a,b,c,.....z}	Error! Only Capital letters are allowed.
9	12345	{0,+-1,+-2,..}	Error! Only Capital letters are allowed.
10	@i	{!,#,$,.....}	Error! Only Capital letters are allowed.

Error Logging

(Comparison between White-box & Black-box)

FR 4 System shall allow the user to access the system by entering the username of 5 to 10 characters

T.C ID	Expected Output	Actual Output
1	Valid	Login Successful!
2	Invalid	Invalid Login Credentials
3	Invalid	Invalid Login Credentials

FR 5 System shall allow the user to access the system by entering the password of 8 to 15 characters

T.C ID	Expected Output	Actual Output
4	Valid	Login Successful!
5	Invalid	Invalid Login Credentials
6	Invalid	Invalid Login Credentials

FR 6 System shall allow admin to add student details. Student name must be in capital alphabets.

T.C ID	Expected Output	Actual Output
7	Valid	Student Name is Valid
8	Invalid	Error! Only Capital letters are allowed.
9	Invalid	Error! Only Capital letters are allowed.
10	Invalid	Error! Only Capital letters are allowed.

Error Analysis

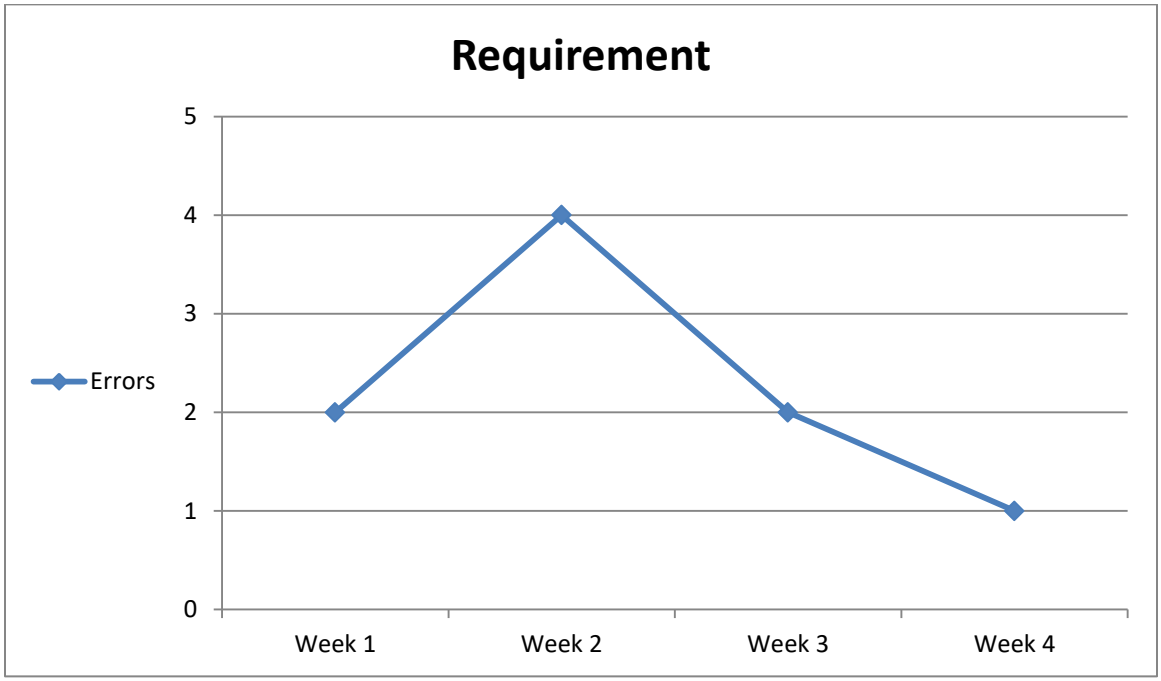
Goal:

I have tested 20 Test cases of 3 Software Artifacts (i.e Requirements, Use Case and Code). This process of testing continued for four week. The ultimate goal is to identify the number of errors in different artifact of software in given amount of time.

	Requirements	Use Case	Source Code
Test Cases Passed	11	7	8
Test Cases Failed	9	13	12

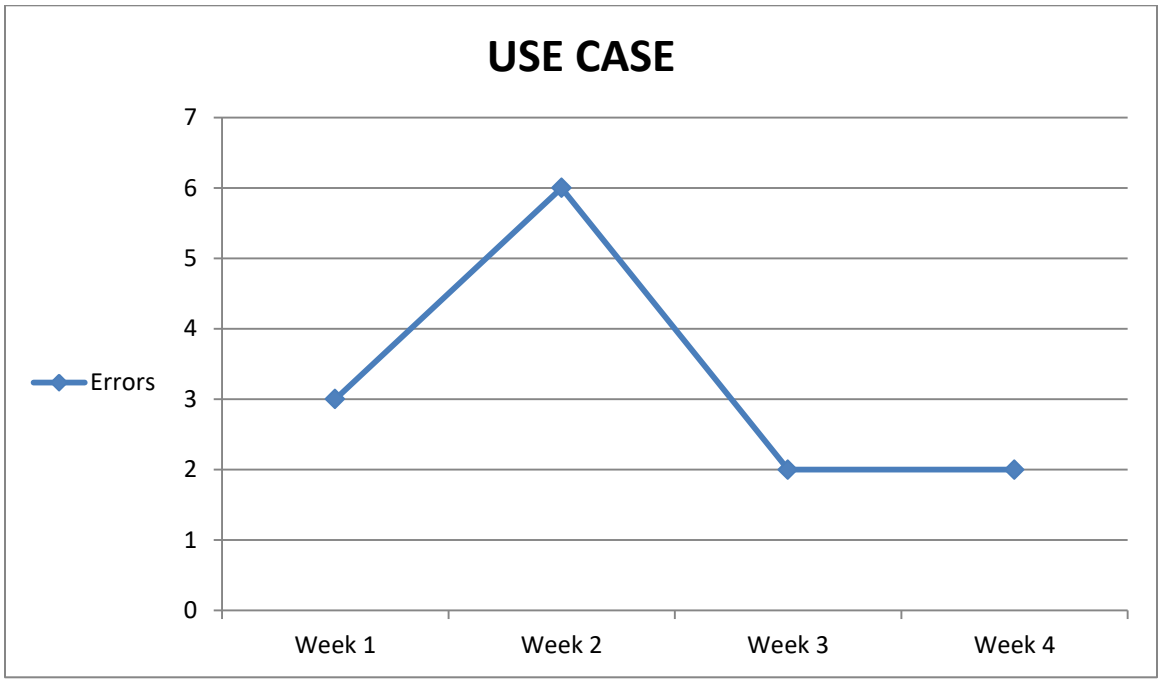
Requirement:

Metric	Errors
Week 1	2
Week 2	4
Week 3	2
Week 4	1



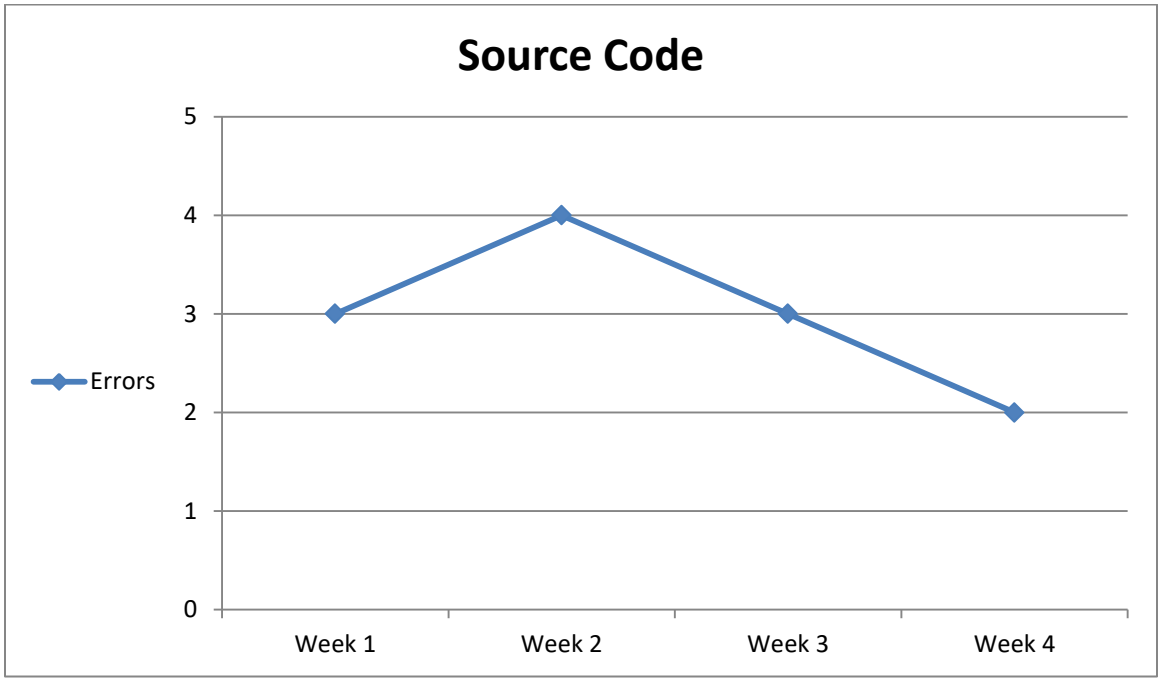
Use Case:

Metric	Errors
Week 1	3
Week 2	6
Week 3	2
Week 4	2



Source Code:

Metric	Errors
Week 1	3
Week 2	4
Week 3	3
Week 4	2


Source Code
5
4
3
2
1
0
Errors
Week 1
Week 2
Week 3
Week 4

Error Containment

Scenario:

The Scenario is for hostel management system in which the admin Edit student details. There are many possible errors which could occur.

Event Tree Analysis: