



**ATMA RAM**  
**DHARMA**



**SANATAN**  
**COLLEGE**

**UNIVERSITY OF DELHI**

## **SOFTWARE ENGINEERING PROJECT REPORT**

Paper Code : 32341401

SUBMITTED BY

Name	Roll No.
Harsh Meena	20/88028

Tanmaya Tiwari	20/88041
Divya Deept Pandey	20/88040

Bsc. (Honours) Computer Science

## **ACKNOWLEDGEMENT**

On the successful completion of project entitled "**Student Career Consultation and Exam Detail System**", I would like to express my sincere thanks of gratitude to our supervisor **Mrs. Uma Ojha** for her kind supervision, valuable guidance and constant encouragement towards fulfilment of this project. The attention and ease with which she has followed us through to this work can never be aptly repaid.

Thanking You

# **Student Career Consultation and Exam Detail System**

## **INTRODUCTION**

The Student Career Consultation and Exam Detail  
System

provides students information about all the examinations they should be appearing for based on their interests and their future goals like its registration dates, eligibility criteria, previous year papers, syllabus and all the preparatory resources on a single platform. It also provides an environment in which a student can ask career related questions from the experts. Not only that, a confused student can schedule an online meeting from a counsellor expert so that he doesn't end up taking the wrong decision

## INDEX

Sr. No.	Topic	Page No.
1	Problem Statement	6
2	Process Model	7-8

3	Use Case Diagram	9
4	Use Cases	10-13
5	Data Flow Diagram	14-18
6	Data Dictionary and Sequence Diagram	19-20
7	Software System Specification	21-27

### Product Metrics

8	Function Point	28-30
9	COCOMO II	31-34
10	Gantt Chart	35
11	Structure Chart	36
12	Black Box Testing	37-38
13	White Box Testing	39-42
14	Screenshots	43-44

## PROBLEM STATEMENT

In India nearly 85% of higher education candidates are not equipped to take career decisions according to India Skills Report 2018. India is perceived as counselling-deficit sector where over 93% schools do not have a professional counsellor on board.

Apart from the usual pressure of competition , the increasing complexity of career choices and disruptive technologies in the volatile job market have made career choices a major challenge for both students and parents.

A Mindler survey on career options awareness revealed that 93% of the students aged 14 to 21 were aware of just seven career options though there are more than 250 different types of job options available in India. Every hour one student commits suicide in India.

Many students after their high school are unaware of what examinations to give for the right career. Multiple of them unfortunately miss the examination due to lack of knowledge about the registration dates of the same. Also some find it difficult to find the resources needed for the preparation of the respective examination, wasting precious time in clicking different website links only to end up on an advertisement.

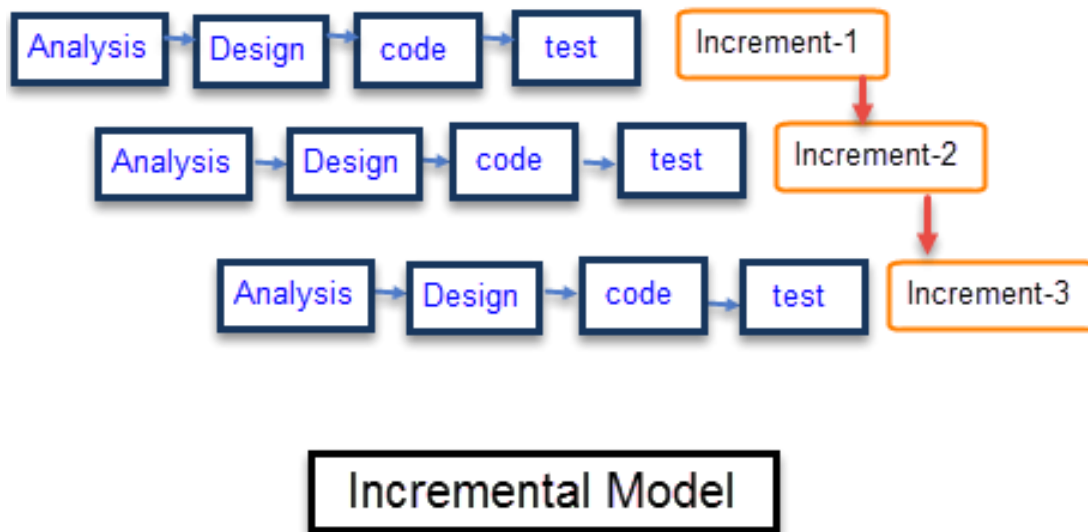
The Student Career Consultation and Exam Detail System provides students information about all the examinations they should be appearing for based on their interests and their future goals like its registration

dates, eligibility criteria, previous year papers, syllabus and all the preparatory resources on a single platform. It also provides an environment in which a student can ask career related questions from the experts. Not only that, a confused student can schedule an online meeting from a counsellor expert so that he doesn't end up taking the wrong decision

## PROCESS MODEL

Model used in our application is **Incremental Model**. Incremental Model is a process of software development where requirements divided into multiple standalone modules of the software development cycle. In this model, each module goes through the requirements, design, implementation and testing phases. Every subsequent release of the module adds function to the previous release. The process continues until the complete system achieved.





The various phases of incremental model are as follows:

**1. Requirement analysis:** In the first phase of the incremental model, the product analysis expertise identifies the requirements. And the system functional requirements are understood by the requirement analysis team. To develop the software under the incremental model, this phase performs a crucial role.

**2. Design & Development:** In this phase of the Incremental model of SDLC, the design of the system functionality and the development method are finished with success. When software develops new practicality, the incremental model uses style and development phase.

**3. Testing:** In the incremental model, the testing phase checks the performance of each existing function as well as additional functionality. In the testing phase, the various methods are used to test the behaviour of each task.

**4. Implementation:** Implementation phase enables the coding phase of the development system. It involves the final coding that design in the designing and development phase and tests the functionality in the testing phase. After completion of this phase, the number of the product working is enhanced and upgraded up to the final system product.

### Advantage of Incremental Model -

- Errors are easy to be recognized.
- Easier to test and debug

- More flexible.
- Simple to manage risk because it handled during its iteration.
- The Client gets important functionality early.

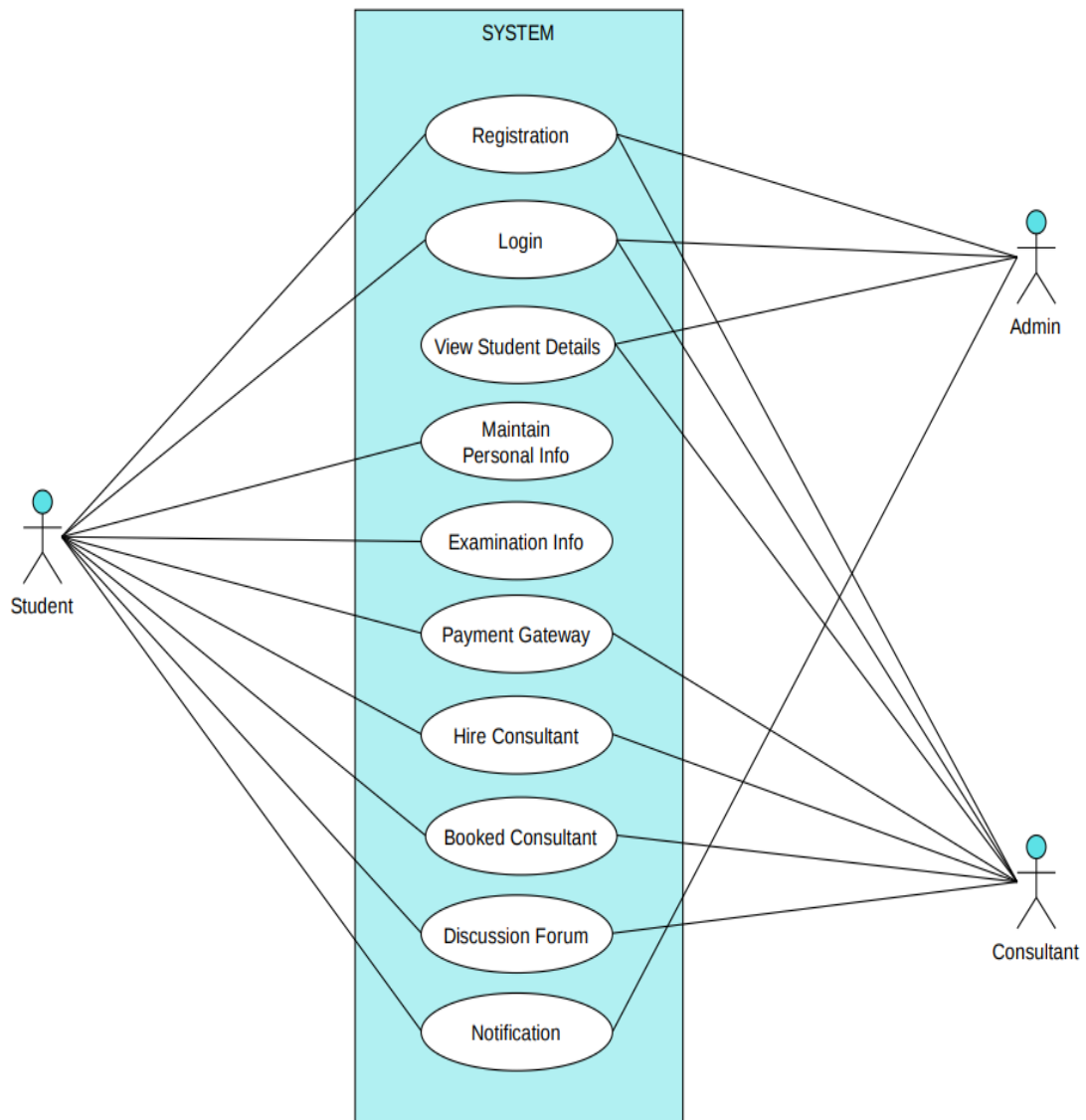
### **Disadvantage of Incremental Model -**

- Need for good planning
- Total Cost is high.
- Well defined module interfaces are needed.

### **Why we choose the Incremental Model -**

- Requirements of the system are clearly understood
- System development is broken down into many mini development projects
- Partial systems are successively built to produce a final total system
- The highest priority requirement is tackled first
- Once the requirement is developed, a requirement for that increment is frozen.
- It is flexible and less expensive to change requirements and scope
- Throughout the development, stages changes can be done
- This process model is simple to manage risk because it handled it during iteration.

## USE CASE



## USE CASES

## Registration:

- **Brief Description:** This use case describes how a user register into the system.
- **Actors:** The following actors take part or interact with the system:
  - Admin
  - Consultant
  - Student

- **Flow of Events:**

Basic Flow:

- Any actor registering the system will be asked for username, E-mail ID, Password, role and other required data with respect to their role as an actor.
- The data is stored into a database.
- Then, it shows a successful registration if not then an error otherwise.

Alternative Flow:

In case of unsuccessful registration, there would have been an error raised and asked user to enter correct information.

- **Special Requirements:** None
- **Pre-conditions:** None
- **Post-conditions:** After successful registration, the actor is registered into the system and its data got saved into database.

## Login:

- **Brief Description:** The case describes the how any of the registered user can login the system.
- **Actors:** The following actors take part or interact with the system:
  - Admin
  - Consultant
  - Student

- **Flow of Events:**

Basic Flow:

- The actor login system with their credentials.
- Then data is verified with the database for valid actor.
- If the actor is already registered then they can move ahead.

Alternative flow:

In case of unsuccessful login, the user prompted with an error and asked to re-enter the credentials correctly or else to register them.

- **Special Requirements:** None
- **Pre-conditions:** The actor must be registered before.
- **Post-conditions:** If login done successfully then actor would be directed to the next page.

## Maintain Personal info:

- **Brief Description:** This use case allows the actor to maintain personal information. This includes adding and updating the information from the system.
- **Actors:** The following actors take part or interact with the system:
  - Student
- **Flow of Events:**

Basic Flow:

- This case starts when the actor wishes to update or add information from the system.
- The actor then updates or add the required information.

Alternative flow:

In case of unsuccessful attempt, the user prompted with an error and asked to re-enter the info correctly.

- **Special Requirements:** None
- **Pre-conditions:** The actor must be logged in through the registered id.
- **Post-conditions:** If the use case was successful the actor has updated or added information from the system.

## View Student Details:

- **Brief Description:** This use case allows the actors to view student information. This includes viewing or updating the information from the system.
- **Actors:** The following actors take part or interact with the system:
  - Admin
  - Consultant

- **Flow of Events:**

Basic Flow:

- **Admin:**  
This case starts when the actor wishes to update or view information from the system.  
The actor then updates or view the required information.
- **Consultant:**  
This actor can only view the information from the system.

Alternative flow:

If the Actor enters invalid info while updating, then the user is prompted with an error and asked to re-enter the info correctly.

- **Special Requirements:** None
- **Pre-conditions:** The actor must be logged in through the registered id.
- **Post-conditions:** If the use case was successful the actor has updated or viewed information from the system.

## Payment Gateway:

- **Brief Description:** This use case acts as a payment portal for actor. This includes seeing how and where the cash is flowing in or out.
- **Actors:** The following actors take part or interact with the system:
  - Student
  - Consultant
- **Flow of Events:**

#### Basic Flow:

- This case starts when the actor wishes to pay, review or generate payment information reports.
- The system asks the actor to specify the function that they'd like to perform, i.e., pay, review, or generate payment information reports.
- The actor then fills in the required information.
- The system then executes one of the sub-flows based on what the actor chose.
- If the actor chose "review payment information", the review payment information sub-flow is executed.
- If the actor chose "Pay option", then the payment sub-flow is executed.
- If the actor chose "generate payment report", the generate payment report sub-flow is executed.

#### Alternative flow:

If the Actor decides not to update the payment info, then the update is cancelled and they are returned to the beginning of the Basic flow.

- **Special Requirements:** None
- **Pre-conditions:** The actor must be logged in through the registered id.
- **Post-conditions:** If the use case was successful the actor has paid, viewed or generated payment reports from the system.

## Hire Consultant:

- **Brief Description:** This use case allows actor to book an appointment with the desired consultant which actor selects from the options given to him/her.
- **Actors:** The following actors take part or interact with the system:
  - Student
  - Consultant
- **Flow of Events:**

#### Basic Flow:

- Student - This use case starts when the actor wishes to book an appointment with the consultant. The actor is given a list of consultants to choose from. After selecting the desired consultant, the actor is then directed to the payment gateway.

- Consultant – The actor can view the appointment request made by Student and accept or reject the appointment after checking the request.

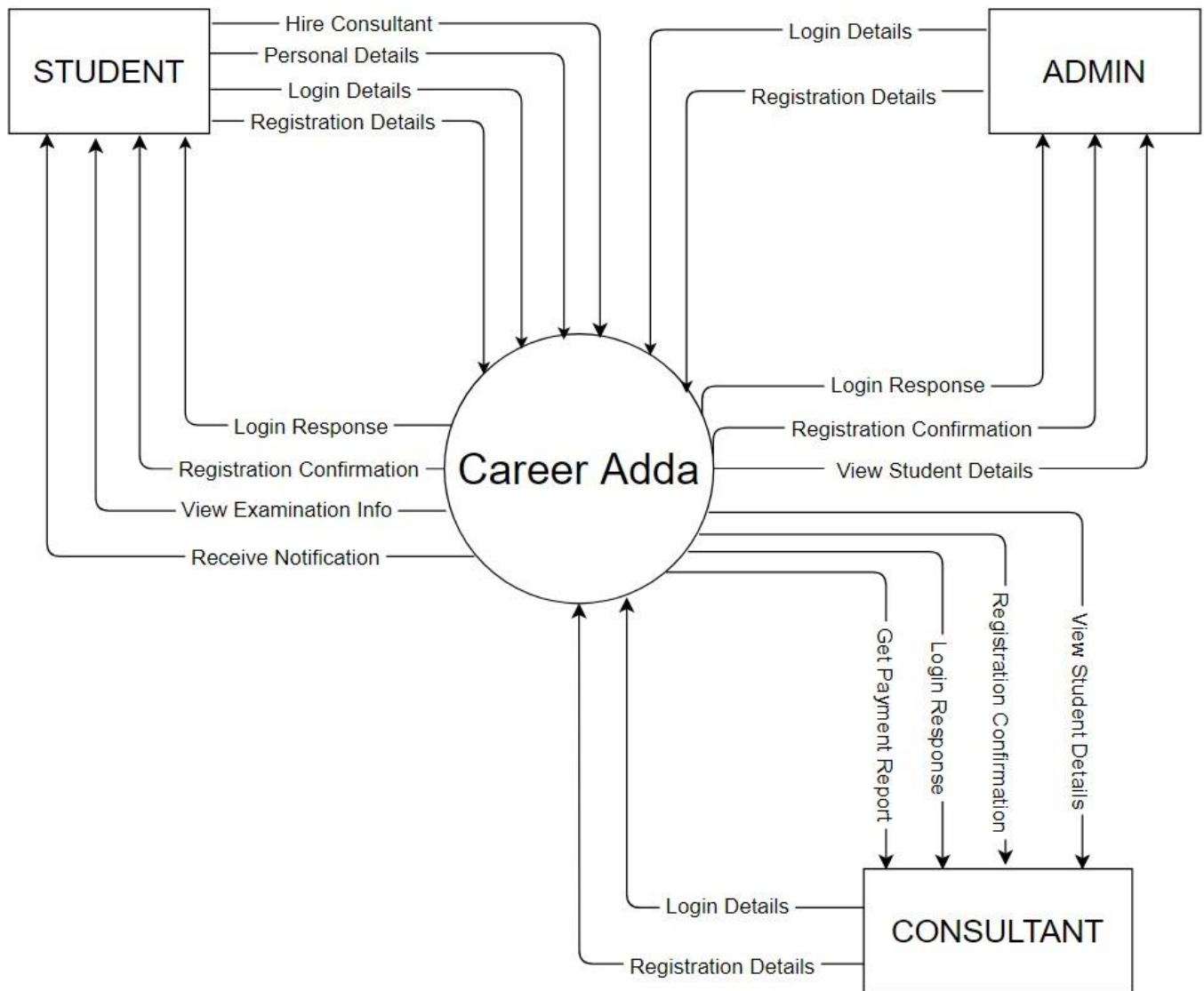
Alternative flow:

If the Actor decides to reject the appointment, then the student is returned to the beginning of the Basic flow.

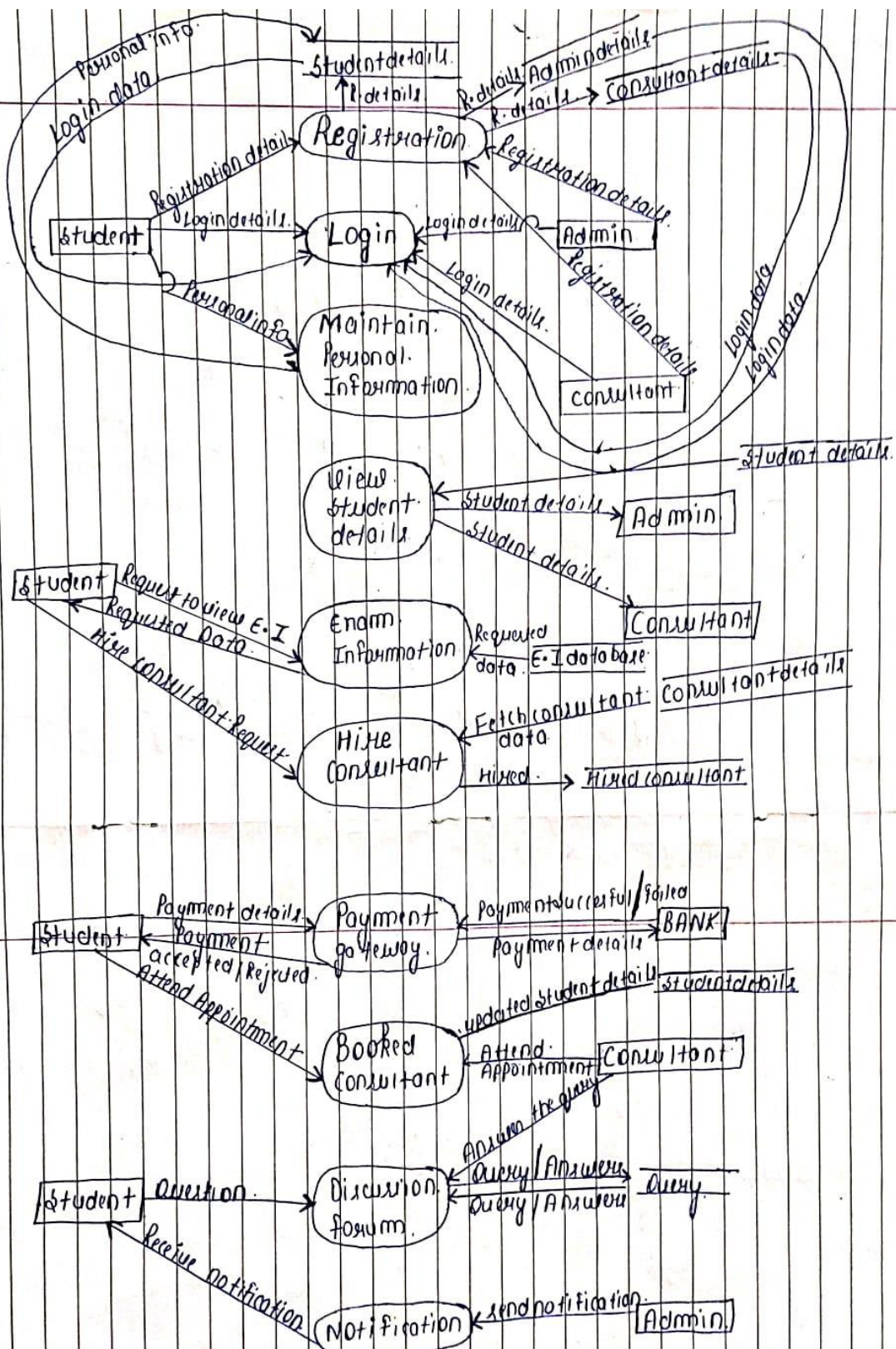
- **Special Requirements:** None
- **Pre-conditions:** The actor must be logged in through the registered id.
- **Post-conditions:** If the use case was successful the actor has booked an appointment, then they have access to the booked consultant option.



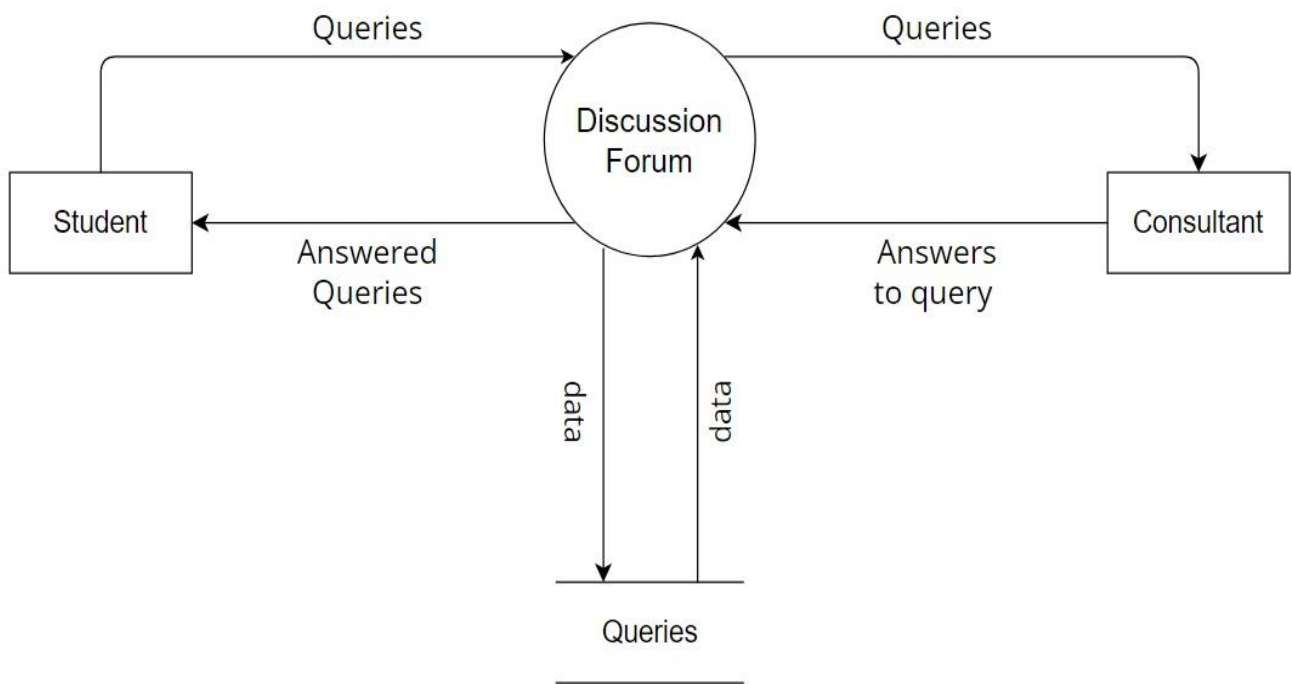
# DATA FLOW DIAGRAM



Level 0

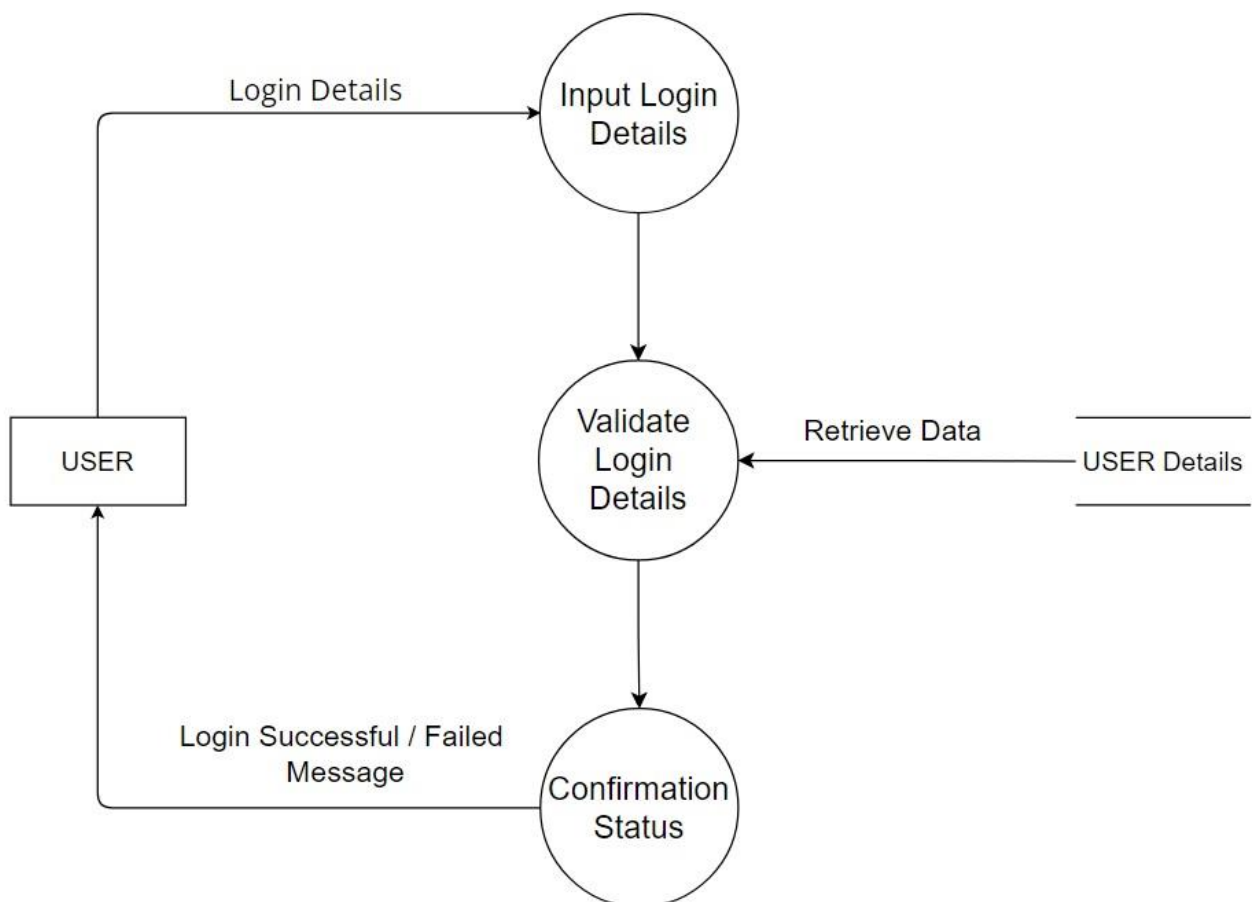


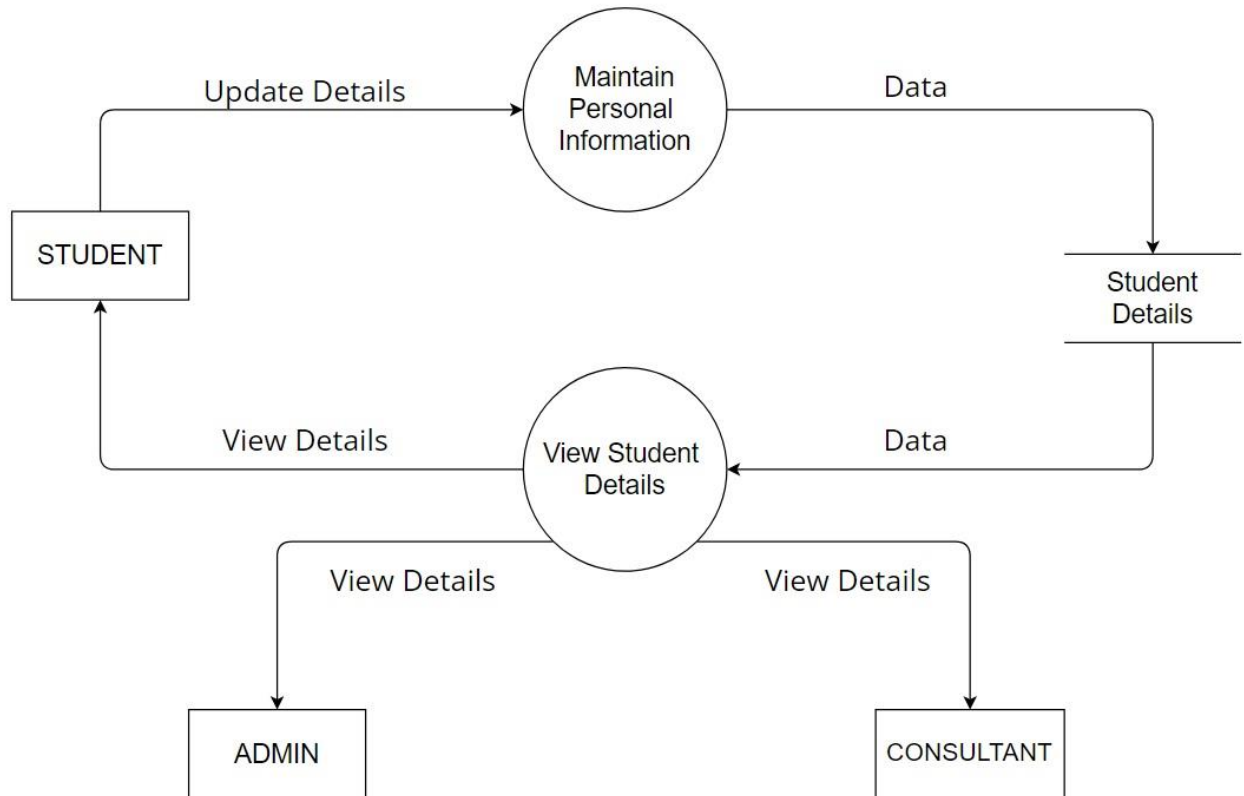
Level 1



## Level 2 Discussion Forum

## Level 2 Login





## Level 2 Student Details

# DATA DICTIONARY AND SEQUENCE DIAGRAM

The data dictionary provides an **organized** approach for representing the characteristics of each data object and control item. It has been proposed for describing the content of objects defined during structured analysis. A data dictionary is very **important** in the software development process because:

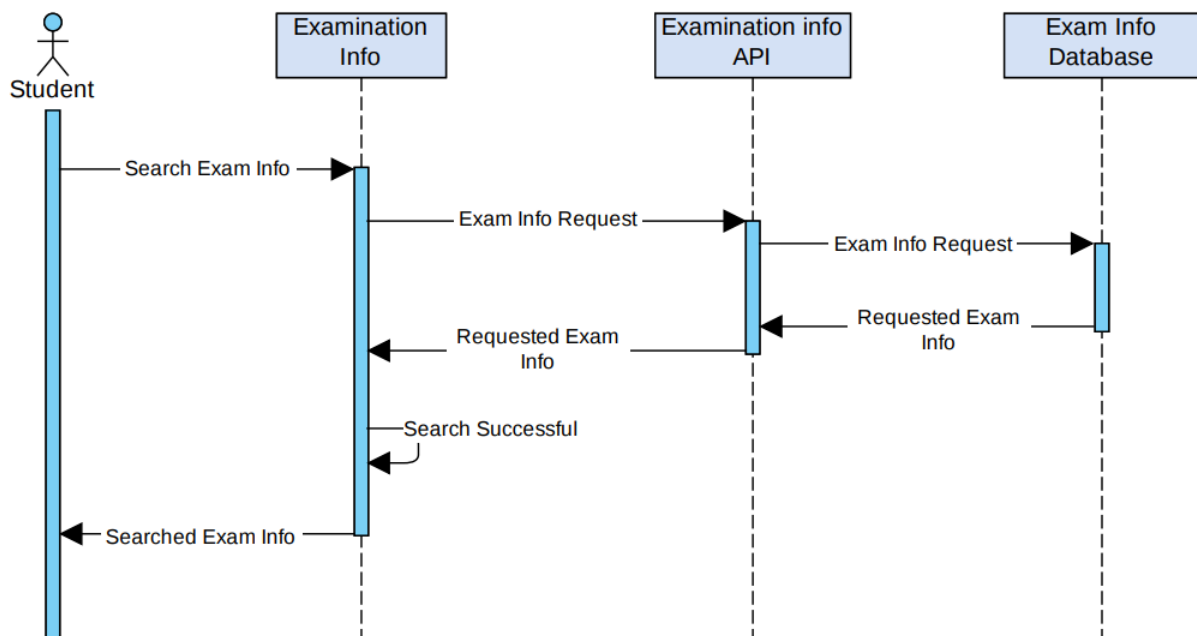
- A data dictionary lists standard terminology for use by an engineer working on a project.
- The dictionary provides the analyst with means to determine the definition of different **data structures** in the terms of their component elements.

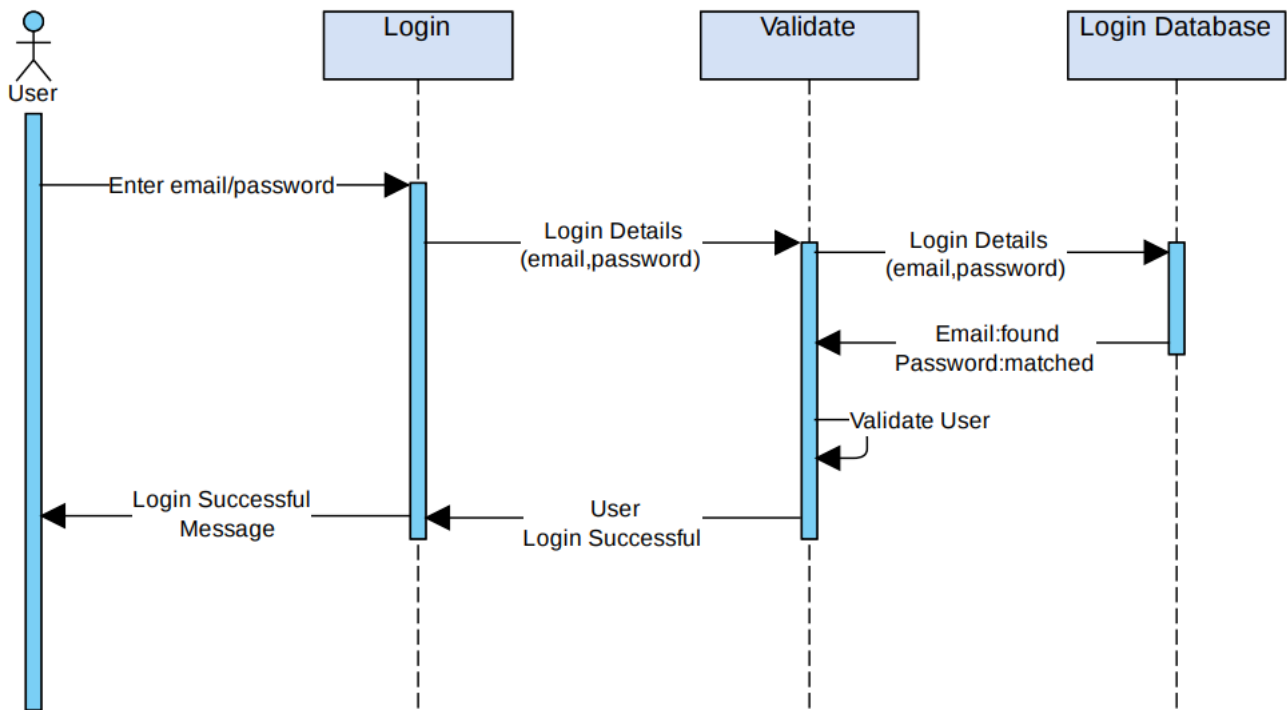
## The Format of Data Dictionary:

- Name the **primary name** of the data or control item, the data store or an external entity.
  - Aliases-other names used for first entity.
  - **Description** a notion for representing content.
  - **Type** of the data.

<b><u>FIELD NAME</u></b>	<b><u>DESCRIPTION</u></b>
Registration	Name + Email + Password + PhoneNO + Role(Admin, Consultant, Student)
Login	Email + Password
Personal Info	Result
Examination Info	Eligibility + Tentative Dates + Study Material
Payment Gateway	Name + Bank Name + CVV + Amount

## SEQUENCE DIAGRAM





# SOFTWARE REQUIREMENT SPECIFICATION

## 1. Introduction

The purpose of this SRS document is to provide a detailed overview of the Student Career Consultation and Exam Detail System along with its parameters and goals. This describes the projects audience and its user interface, hardware and software requirements. It defines how the user sees the product and interacts with its functionality. Nonetheless, it helps any designer and developer to assist in Software Development Life Cycle (SDLC) processes.

### 1.1 Purpose:

The purpose of this SRS document is to present a detailed description of the ....., a website/app. It will explain the purpose and features of the system, interfaces of the



system, what the system will do, the constraints under which it must operate and how the system will react to external activities.

## **1.2 Scope:**

The system mainly focuses on giving information to a student about all the examinations they should be appearing for based on their interests and their future goals like its registration dates, eligibility criteria, previous year papers, syllabus and all the preparatory resources on a single platform. It also provides an environment in which a student can ask career related questions from the experts and can even schedule an appointment with the consultant

## **1.3 Definitions, Acronym, Abbreviation**

Term	Definition
User	Any person or company logged on the system
Database	Collection of all the information monitored by the system.
DFD	Data flow Diagram
Software Requirements Specification	A document that completely describes all the functions of a system.

## **1.4. References:**

- Pressman, R. S., & Maxim, B. R. (2015). Software Engineering: A Practitioner's Approach 8th edition, McGraw-Hill.
- Aggarwal, K. K., & Singh, Y. (2007). Software Engineering. 3rd edition.
- IEEE recommended practice for Software Requirement Specifications. ISO/IEC/IEEE 29148:2011

## **1.5 Overview:**

The remaining section of this document gives the general description, including the functionality of this project. It provides the general factors that affect the product and its specific requirements.

## **2. Overall Description:**

This Software project allows user to search examination related info and contact to consultant regarding their future. There is also an ADMIN which has the role of maintaining all others accounts and s/he also creates a key through which all other users can register to the system.

### **2.1 Product Perspective:**

The application will be a windows-based application service in order to search and book the service providers to get their work done. User will get a quotation for their job and then one can choose the accurate service provider. The history is stored and a user can access this from their login details.

- **System Interfaces:**

The system may interface with SQL.

- **User Interfaces:**

The application has user-friendly and menu-based interface as follows:

- **Register Screen:** The screen displays the columns where users are asked to enter the relevant details and get themselves register with the system. The screen demands the user's name (full name + last name), username, password, phone number, email address and a key generated by the admin for a successful registration with the system.

- **Login Screen:** The users shall use their username and password to login the system and access the feature according to their role.
- **User Profile Screen:** The user through this screen can view or edit the information in their profile.
- **View Examination Info Screen:** The screen displays the details about the various exams. The student can get all details regarding the desired exam they are willing to study for.
- **Payment Gateway Screen:** The client would be prompted with payment module after choosing for online payment. The screen displays the columns and asked to fill the details related to the payment.
- **Student Detail Screen:** The Admin and Consultant through this screen can view the student details stored in the database.
- **Hire Consultant Screen:** The screen displays the list of consultants according to student's choices. And student can book an appointment with the suitable consultant.
- **Booked Consultant Screen:** The screen displays the details of student's personal appointment with the hired consultant.
- **Discussion Forum Screen:** The screen displays the message window, through which client and service provider can communicate with each other regarding the service.
- **Notification:** The small notification screen is visible whenever any services got booked or cancelled.

## 2.2 **Product Functions**

The product functions generally focus on the providing the examination info and best consultant support to a student.

### **2.3 User Characteristics:**

User should be familiar with the terms like login, register, exam info etc.

- **Technical Expertise:** User should be comfortable using general purpose application on a computer.
- **Language:** The user should at least understand English language to easily function through the system.
- **Knowledge:** The user must have basic knowledge regarding the career they are planning to pursue in their future.

### **2.4 Principal Actors:**

Principle Actors are Student and Consultant.

### **2.5 Constraints:**

- Validate for registered user via email id password.
- Be robust enough so that users do not get delayed.
- Be able to handle multiple users at the same time.

### **2.5 Hardware Interface:**

- The system must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. E.g., modem, WAN, LAN, Ethernet.

## **2.6 Software Interface:**

Windows operating system windows 7/8/10.

- Web browser
- WAMP Server.
- MySQL Database Management System.

## **3. Specific Requirements:**

This section contains all the software requirements at a level of detail sufficient to enable designers to design or system to satisfy these requirements, and testers to test that the system satisfies those requirements.

### **3.1 External Interface Requirement:**

- **User Interfaces:**

The following screens will be provided:

- **Registration Screen:**

This screen will be the first screen to be displayed. It will allow users to get them registered or to access the system. Various fields available on this screen will be:

- User ID: Alphanumeric of length up to 20 characters.
- E-mail/Phone no.

- Password: Alphanumeric.
- Role: Have Student, Consultant and ADMIN.

- **Login Screen:**

User need to enter the email id and Password and if user enters incorrect password, then they will be prompted with an error. After entering the correct details, user can access platform.

- **Examination Information:**

This screen is visible to students willing to view all exam related information.

Students are provided with detailed information about each exam which includes general information about the exam, eligibility criteria, important dates, syllabus and much more based on their search.

- **Hire Consultant:**

Based on student's choices they are provided with a list of experienced Consultant from which student need to select a suitable consultant to book an appointment with.

- **Booked Consultant:**

After the student has booked an appointment, they are then eligible for accessing the live discussion session with their selected consultant.

- **Discussion Forum:**

The option acts as a discussion ground for every student using the app, as they able to express their doubts and ask questions to the consultant regarding their studies and examinations.

- **Payment Gateway:**

- **State:** Display the payment tabs and columns.
- **Input:** Client need to enter the payment related details and make the payment.
- **Output:** Display the payment receipt.

### **3.2 Performances Requirements:**

- It must be able to handle multiple users.
- High data transfer rate.
- 24/7 data connection should be available.

### **3.3. Logical Database Requirements:**

- The application communicates with the internal database.
- The database is used for registration, login, to set and get student details.
- The database is used for maintaining the discussion forum.

### **3.4. Software System Attributes:**

- **Availability:** The system should have a good internet connection to receive the notifications timely. The system should be backed up timely for easy recovery in any dysfunction scenarios.
- **Security:** The application is secured by a username and password.
- **Maintainability:** The system will be designed in a maintainable manner. It will be easy to incorporate new requirements in the individual.
- **Portability:** The software will be portable and can be used on any windows-based electronics.

# PRODUCT METRICS

Project metrics and the indicators derived from them are used by the project manager and a software team to adapt project work flow and technical activities. The intent of project metrics is twofold. First, there metrics are used to minimize the development schedule by making the adjustment necessary to avoid delays and mitigate potential problems and risks. Second, project metrics are used to assess product quality on an ongoing basis and, when necessary, modify the technical approach to improve quality.

## FUNCTION POINT:

External Input (EI): 4

- Student, Admin, consultant registration details
- Student enters academic details
- Student enters payment details
- Student, Consultant enter queries and answers discussion forum respectively

External Output (EO): 5

- Student, Admin, consultant registration confirmation/rejection
- Student, Admin, consultant login confirmation/rejection
- View notification
- View list of consultants
- Payment rejection/acceptance

External Inquiries (EQ): 4

- Login
- Get student details
- Get Exam info
- Get consultant details

Internal Logical Files (ILF): 2

- Student record
- Consultant record

External Interface Files (EIF): 1

- Examination info from API



Information Domain Value	Count	Weighting factor				
		Simple	Average	Complex		
External Inputs (EIs)	<input type="text" value="3"/>	3	4	6	=	<input type="text" value="18"/>
External Outputs (EOs)	<input type="text" value="3"/>	4	5	7	=	<input type="text" value="21"/>
External Inquiries (EQs)	<input type="text" value="3"/>	3	4	6	=	<input type="text" value="18"/>
Internal Logical Files (ILFs)	<input type="text" value="3"/>	7	10	15	=	<input type="text" value="45"/>
External Interface Files (EIFs)	<input type="text" value="3"/>	5	7	10	=	<input type="text" value="30"/>
Count total						<input type="text" value="132"/>

Information domain value	Count		Weighting Factor	
External inputs	4	X	4 (average)	= 16
External outputs	5	X	5 (average)	= 25
External inquiries	4	X	6 (complex)	= 24
Internal logical files	2	X	10 (average)	= 20
External interface files	1	X	5 (simple)	= 5
Count total				= 90

Value Adjustment Factor Calculation Table –

Factor (Fi)	Value (0-5)
1. Backup and recovery	4
2. Data Communication	2
3. Distributed Processing	0
4. Performance Critical	3
5. Existing Operating Environment	3
6. Online Data Entry	5
7. Input Transaction over Multiple Screens	3

8. ILFs updated online	5
9. Information Domain values Complex	2
10. Internal Processing Complex	4
11. Code Designed for Reuse	4
12. Conversion/Installation in Design	3
13. Multiple Installations	2
14. Application Designed for Change	4
$\sum(F_i)$	44
Value Adjusting Factor $[0.65 + 0.01 \times \sum(F_i)]$	1.09

$$FP = COUNT\ TOTAL \times [0.65 + 0.01 \times \sum(F_i)]$$

$$FP = COUNT\ TOTAL \times VALUE\ ADJUSTING\ FACTOR$$

$$FP = 90 \times 1.09$$

$$\text{Thus, } FP = 98.1$$

$$\approx 98$$

**Assumption:** The organizational average productivity for systems of this type is 6.5 FP/pm (here pm is person- month) based on a burdened labour rate of \$8000 per month.

- **COST PER FP = Labor Rate per month/Value of FP per pm**

$$= 8,000 / 6.5$$

$$= \$1,230.769 = \$1,230.8$$

$$\approx \$1,231$$

- **ESTIMATED OR TOTAL PROJECT COST = Cost of one FP X Estimated FP**

$$= \$1,231 \times 98$$

$$= \$120,638$$

- **EFFORTS = Estimated FP/ FP per pm**

= 98/6.5

= 15.07 person- month **≈ 15 person-month**

## EFFORT ESTIMATION USING COCOMO II MODEL

COCOMO-II model, constructive cost model is one of the most widely used software estimation model that addresses the following areas:

- Application composition model
- Early design model
- Post architecture state model

Our project is based on application-based model as this model is used during early stages of software when prototyping of user interface, consideration of software, system interaction, assessment of performance and evaluation of technology maturity is paramount.

Object type	Complexity weight		
	Simple	Medium	Difficult
Screen	1	2	3
Report	2	5	8
3GL component			10

Object point is indirect software count measure i.e. compiled using counts of number of screens, reports and 3GL components. Each object instance is classified into one of the three complexity levels: simple, medium or complex.

The object count is determined by multiplying the total number of object instances by weighing factor. When component-based development or general software re-used is to be applied, the percent of re-use is estimated and object count is adjusted. In our software project, the **percent re-use is 10%**.

<b>Developer's experience/capability</b>	Very low	Low	Nominal	High	Very high
<b>Environment maturity/capability</b>	Very low	Low	Nominal	High	Very high
<b>PROD</b>	4	7	13	25	50

To calculate weight complexity for screen we will follow this table:

No. of views contain	Sources of data tables		
	Total < 4 ( < 2 servers < 3 clients )	Total < 8 ( 2 - 3 servers 3-5 clients )	Total 8 + ( > 3 servers > 5 clients )
< 3	Simple	Simple	Medium
3 - 7	Simple	Medium	Difficult
> 8	Medium	Difficult	Difficult

**For Screens**

To calculate weight complexity for report we will follow this table:

No. of section contain	Sources of data tables		
	Total < 4 ( < 2 servers < 3 clients )	Total < 8 ( 2 - 3 servers 3-5 clients )	Total 8 + ( > 3 servers > 5 clients )
0 - 1	Simple	Simple	Medium
2 - 3	Simple	Medium	Difficult
4 +	Medium	Difficult	Difficult

**For Reports**

➤ Number of Screens:

1. Registration: It need 3 data table and 3 view of screen.
2. Login: It need 3 data table and 3 view of screen.
3. Maintain Personal Info: It need 1 data table and 1 view of screen.
4. View Student Details: It need 1 data table and 2 view of screen.
5. Exam Info: It need 1 data table and 1 view of screen.
6. Payment Gateway: It need 1 data table and 2 view of screen.
7. Hire Consultant: It need 1 data table and 1 view of screen.
8. Booked Consultant: It need 2 data table and 2 view of screen.
9. Discussion Forum: It need 1 data table and 2 view of screen.

➤ Number of Reports:

1. Searching: It has 2 sections [exam info + list of consultants] and it uses 2 tables.
2. Booked Consultant/Customer Appointment: It has 3 sections

[student details + consultant details + appointment details] and it uses 3 tables.

3. Notification: It has 1 section and it uses 1 table.

➤ There are 10 3GL components.

Measuring Weight Complexity:

Sr. No.	Name	Object	Complexity	Weight
1	Registration	Screen	Simple	1
2	Login	Screen	Simple	1
3	Maintain Personal Info	Screen	Simple	1
4	View Student Details	Screen	Simple	1
5	Exam Info	Screen	Simple	1
6	Payment Gateway	Screen	Simple	1
7	Hire Consultant	Screen	Simple	1
8	Booked Consultant	Screen	Simple	1
9	Discussion Forum	Screen	Simple	1
10	Searching	Report	Simple	2
11	Booked Consultant/Customer Appointment	Report	Medium	5
12	Notification	Report	Simple	2
3GL Components		10 x 10		100
Count Total				118

∴ Object Points = 118

$$\text{New Object Point (NOP)} = (\text{object points}) \times [(100 - \% \text{reuse}) / 100]$$

Since, we are using no previously used code or any other component, the reuse% ought to be 10.

$$\therefore \text{NOP} = 106.2$$



Here, Developer's Experience is Low

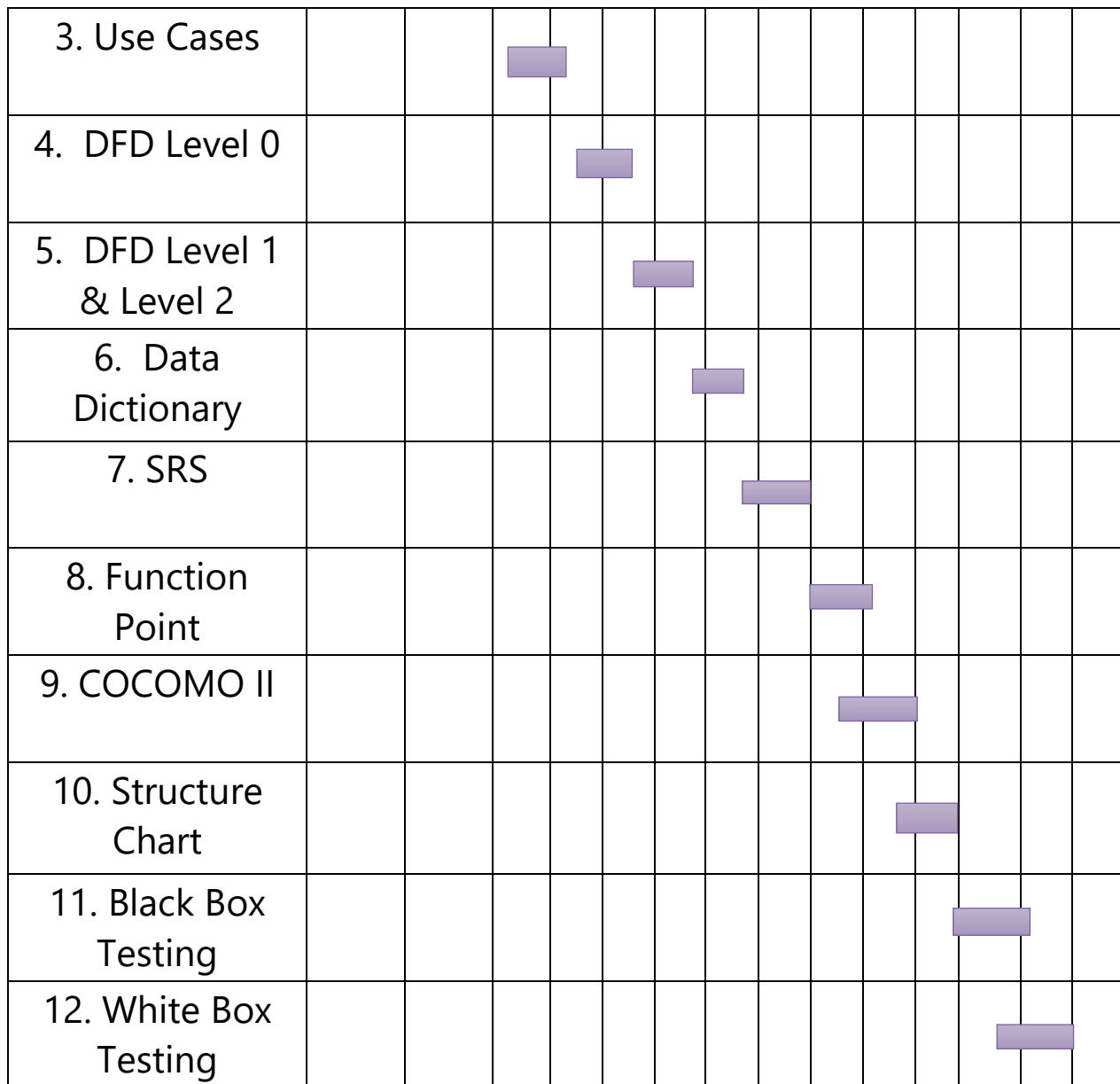
$$\therefore \text{PROD} = 7$$

Now, Estimated Effort = NOP / PROD

$$= 106.2 / 7$$

$$= 15.17 \text{ person-month} \approx 15 \text{ person-month}$$

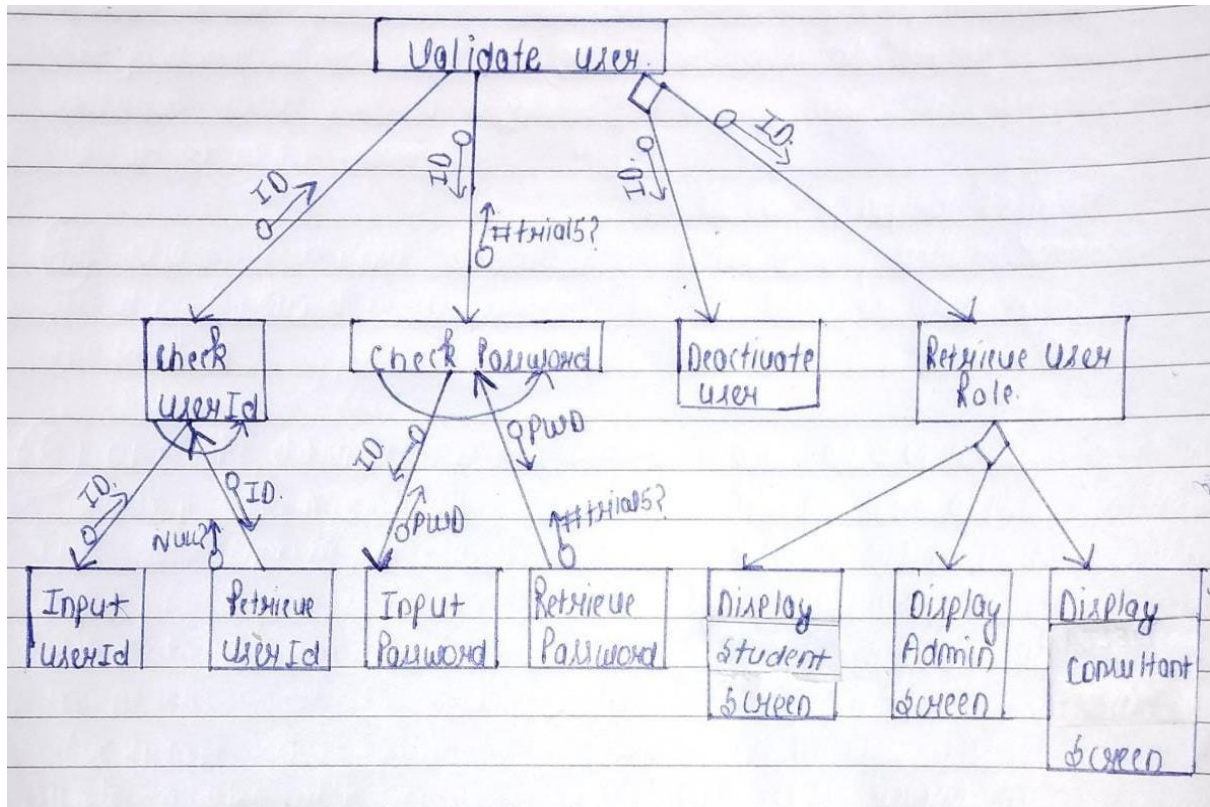
	JANUARY		FEBRARY				MARCH				APRIL			
WEEK	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1. Problem Statement														
2. Software Model														



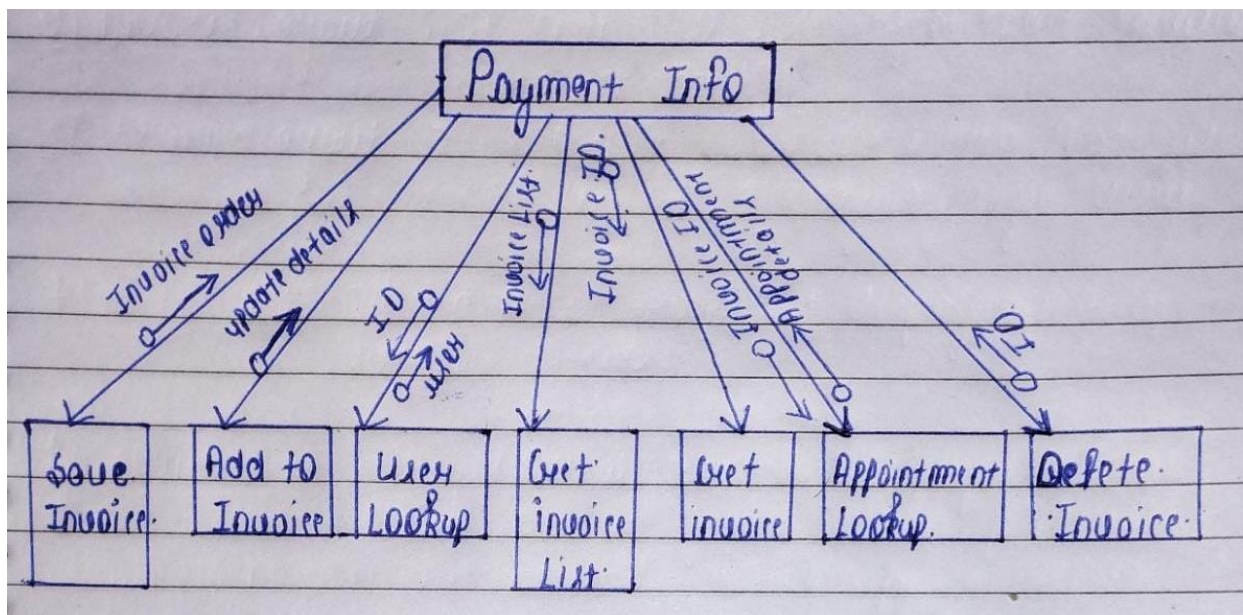
## GANTT CHART

## STRUCTURE CHART





### Payment Gateway –



## BLACK BOX TESTING

- **Test cases for Boundary value analysis:**

Experience shows that test cases that are close to boundary conditions have a higher chance of detecting an error. Here, boundary conditions mean, an input value may be on the boundary, just below the boundary (upper side) or just above the boundary (lower side). We, here, perform Black Box Testing on LOGIN module. The range for Email ID and password is 6-25 and 6-10 characters.

Both can be alphanumeric. And the combination of Email ID and password must match from the values in database.

- **Best Case:**

As we know, with the single fault assumption theory,  $4n + 1$  test cases can be designed and which are here in this case equal to 9. The boundary value test cases are:

Test Cases	Email ID	Password	Expected Output
1	Mpiotr12@aol.com	Qwe24	Invalid
2	Mosses67@yahoo.com	Prod1234	valid
3	teverett@yahoo.ca	Looptest123	Invalid
4	Baveja45@hotmail.com	Baj60	Invalid
5	evilopie@yahoo.com	Shank23	valid
6	Mfleming83@mac.com	Vamos321	valid
7	Papathan90@yahoo.ca	Papapapa001	Invalid
8	Chaikin12@mac.com	Chika	Invalid
9	Sabren16@sbcglobal.net	Top123abc123	Invalid

- **Worst Case:**

If we reject the "single fault" assumption theory of reliability and may like to see what happens when more than one variable has an extreme value. Worst case testing for a function of  $n$  variables generates  $5^n$  test cases. Our login module takes 2 variables input and will have  $5^2 = 25$  test cases and are given below:

Test Cases	Email ID	Password	Expected Output
1	Mpiotr12@aol.com	Qwe2478	valid
2	Mosses67@yahoo.com	Prod1234689	Invalid
3	teverett@yahoo.ca	Looptest123	Invalid
4	Baveja45@hotmail.com	Baj60	Invalid
5	evilopie@yahoo.com	Shank23	valid
6	Mfleming83@mac.com	Vamos321	valid
7	Papathan90@yahoo.ca	Papapapa001	Invalid
8	Chaikin12@mac.com	Chika34	valid
9	Sabren16@sbcglobal.net	Top123abc123	Invalid
10	Mpiotr12@aol.com	Qwe24	Invalid
11	Mosses67@yahoo.com	Prod1234	valid
12	teverett@yahoo.ca	Looptest12389	Invalid
13	Baveja45@hotmail.com	Baj6080	valid

<b>14</b>	evilopie@yahoo.com	Shank	invalid
<b>15</b>	Mfleming83@mac.com	Vamos@45678	Invalid
<b>16</b>	Papathan90@yahoo.ca	Papa1234	valid
<b>17</b>	Chaikin12@mac.com	Chika	Invalid
<b>18</b>	Sabren16@sbcglobal.net	Top123ab	valid
<b>19</b>	Mpiotr12@aol.com	Qwe24	Invalid
<b>20</b>	Mosses67@yahoo.com	Prod1	Invalid
<b>21</b>	teverett@yahoo.ca	Looptest12	valid
<b>22</b>	Baveja45@hotmail.com	Baja1	Invalid
<b>23</b>	evilopie@yahoo.com	Shank236784	Invalid
<b>24</b>	Mfleming83@mac.com	Vamos3213577	Invalid
<b>25</b>	Mpiotr12@aol.com	Qwe24	Invalid

- **Robustness Testing:**

It is nothing but the extension to boundary value analysis. Here, we would like to see, what happens when the extreme values are exceeded with a value slightly greater than the maximum and value slightly less than minimum. It means, we want to go outside the legitimate boundary of input domain. There are four additional test cases which are outside the legitimate input domains. Hence, total test cases in robustness testing are  $6n + 1$ , where  $n$  is the number of input variables. So, here 13 cases are as follows:

<b>Test Cases</b>	<b>Email ID</b>	<b>Password</b>	<b>Expected Output</b>
<b>1</b>	Lukaku23@aol.com	Lakaka12	valid
<b>2</b>	Mosses67@yahoo.com	Prod1234689	Invalid
<b>3</b>	teverett@yahoo.ca	Looptest123	Invalid
<b>4</b>	Baveja45@hotmail.com	Baj60	Invalid
<b>5</b>	evilopie@yahoo.com	Shank23	Invalid
<b>6</b>	Ronald61@mac.com	Abcd456	valid
<b>7</b>	Papathan90@yahoo.ca	Papapapa001	Invalid
<b>8</b>	Chaikin12@mac.com	Chika34	Invalid
<b>9</b>	Sabren16@sbcglobal.net	Top123abc123	Invalid
<b>10</b>	Mpiotr12@aol.com	Qwe24	Invalid
<b>11</b>	Mosses67@yahoo.com	Prod1234	Invalid
<b>12</b>	teverett@yahoo.ca	Looptest12389	Invalid
<b>13</b>	Baveja45@hotmail.com	Baj6080	valid

# WHITE BOX TESTING

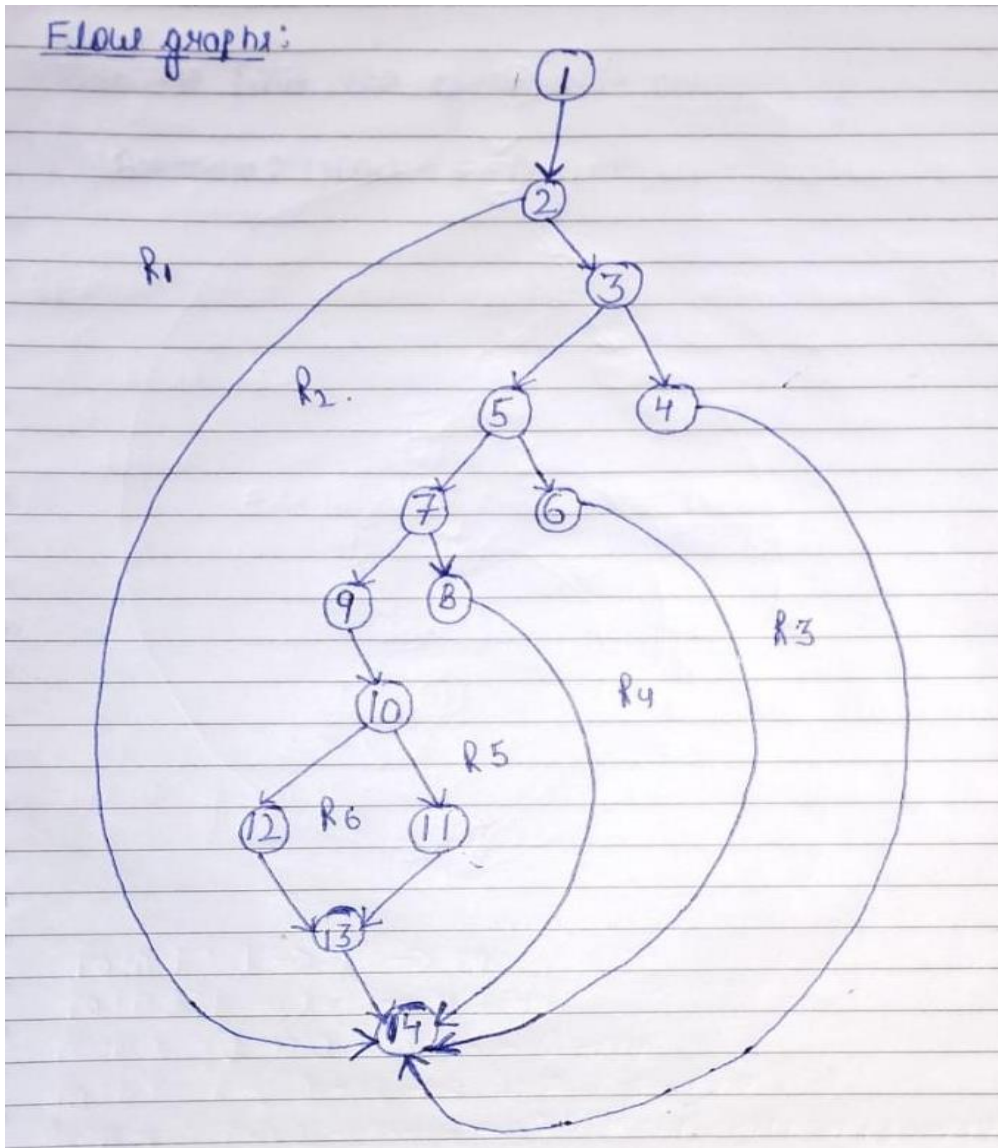
We are performing this white box testing for **register screen**:

<?php

1.     require 'database.php';
2.     if(isset(\$\_POST["submit"]))  
       {  
           \$pass = \$\_POST["txt4"];  
           \$cpass = \$\_POST["txt5"];  
           \$mail = \$\_POST["txt6"];  
3.     if (empty(\$mail) empty(\$pass) empty(\$cpass))  
4.     { echo "FILL IN ALL THE FIELDS";  
       }  
5.     else if(!filter\_var(\$mail, FILTER\_VALIDATE\_EMAIL))  
       {  
6.     echo "wrong email";  
       }  
7.     else if(\$pass!=\$cpass)  
       {  
8.     echo "wrong password";  
       }  
9.     Else  
       {  
           \$sql = " Insert Into users(email, passwd)

```
        values('$mail', '$pass');"
10.    if(mysqli_query($conn, $sql))
        {
11.        echo "Registered successfully";
        }
12.    else echo "Connection error";
13.    }
14.    }
?>
```

FLOW GRAPH –



**CYCLOMATIC COMPLEXITY:** Cyclomatic complexity of a code section is the quantitative measure of the number of linearly independent paths in it. It is a software metric used to indicate the complexity of a program. It is computed using the Control Flow Graph of the program. The nodes in the graph indicate the smallest group of commands of a program, and a directed edge in it connects the two nodes i.e., if second command might immediately follow the first command.

There are three techniques to calculate the cyclomatic complexity which are as follows:

1.  $V(G) = E - N + 2 * P$

where, E= number of edges in the flowgraph

N= number of nodes in the flowgraph

P= number of connected components

So, according to our flowgraph:

$$E = 18, N = 14, P = 1$$

$$\text{Thus, } V(G) = E - N + 2 * P$$

$$= 18 - 14 + 2$$

$$= 6$$

2.  $V(G) = P + 1$  where,  $P$  = number of predicate nodes

So, according to our code:

$$P = 5$$

$$\text{Thus, } V(G) = P + 1$$

$$= 5 + 1$$

$$= 6$$

3.  $V(G)$  = Total number of regions

Hence, according to our flowgraph:

$$\text{No. of regions} = 6$$

$$\text{i.e., } R_1, R_2, R_3, R_4, R_5, R_6$$

$$\text{Thus, } V(G) = 6$$

SO, THE CYCLOMATIC COMPLEXITY,  $V(G) = 6$

## INDEPENDENT PATHS

An independent path is any path through the program that introduces at least one new set of processing statements or a new condition. When stated in terms of a flow graph, an independent path must move along at least one edge that has not been traversed before the path is defined.

From the above Flow Graph of Code, the INDEPENDENT PATHS are:

Path 1:  $1 \rightarrow 2 \rightarrow 14$

Path 2:  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 14$

Path 3:  $1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 6 \rightarrow 14$

Path 4:  $1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 7 \rightarrow 8 \rightarrow 14$

Path 5:  $1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 7 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 13 \rightarrow 14$

Path 6:  $1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 7 \rightarrow 9 \rightarrow 10 \rightarrow 12 \rightarrow 13 \rightarrow 14$

## SCREENSHOTS



Name

Email

Phone No.

Password

REGISTER

## Register Module

Email

Password

LOGIN

## Login Module