Google Meet Extension for Class Management

Software Engineering
Project Report
(C-IX)

Submitted by:
Aaditya Tripathi (19059570008)
Ausaaf Nabi (19059570025)
Vaibhav Maheshwari (19059570012)
B.Sc (H) Computer Science, Semester IV

Supervisor:
Ms. Mona Adlakha

2021

Department of Computer Science
Aryabhatta College
University of Delhi

Contents

Acknowledgement

We wish to express our profound and sincere gratitude to **Mona Adlakha Ma'am**,Computer science Department, Aryabhatta College (University of Delhi) who guided us into the intricacies of this project.We are highly grateful to team members who evinced keen interest and invaluable support in the progress and successful completion of our project work.We are indebted to Our family for their constant encouragement, co-operation and help.Words of gratitude are not enough to describe the accommodation and fortitude which they have shown throughout our endeavour.

Certificate

This is to certify that **Aaditya Tripathi, Ausaaf Nabi** and **Vaibhav Maheshwari** successfully carried out the completion of the project entitled **"Google Meet Extension for Class Management"** under my supervision. The Project has been submitted as per the requirement of the Lab based on **Software Engineering** of B.Sc. (H) Computer Science, IV Semester.

Aaditya Tripathi
19059570008

Ausaaf Nabi
19059570025

Vaibhav Maheshwari
19059570012

Ms. Mona Adlakha
(Supervisor)

Problem Statement

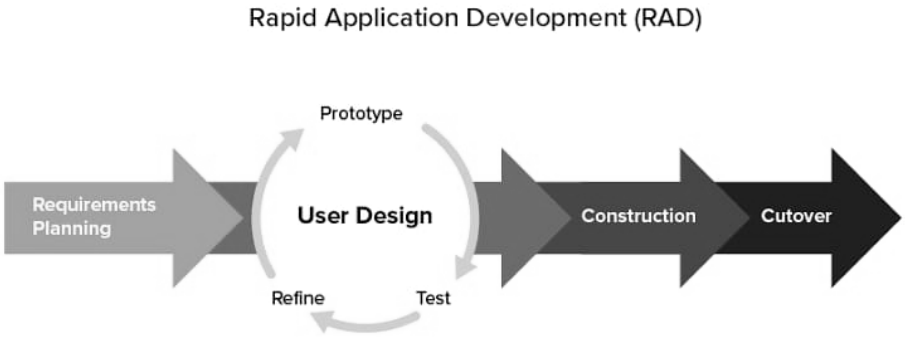
As a large traffic on **Google Meet** is growing day-by-day whether it's from online classes or everyday meetings, everything nowadays is concentrated majorly on Google Meet. There are still some improvements which can be added to make the things easier for the organizer as well as the attendees. Major problems faced during online meetings include record management of attendees/students/participants(Attendance Record) and sharing resources.

The goal of the project **'Google Meet Extension for Class Management'** is to ease the user experience on Google Meet application, and to fulfil the upcoming demands of users from 'Google Meet'.In this, we majorly focus on the problems faced during day-to-day online classes. Major extensions include: **Attendance Record Management System**, we focused on a smart automatic time calculation system, through which we will be able to mark the participant (student) Present/ or Absent. Whereas, **Resource Management System**, sharing of resources through Google Meet can be enforced using this extension. It can be majorly benefitted for teachers/organizers as well as students/attendees attending the online lectures. Users here can be anyone using these extensions of Google Meet.

Process Model

The process model chosen by us is the **RAD Model(Rapid Application Development Model) [Refer Appendix A]**. We have chosen this model because of the following features:

- **Time factor -** Time is a major constraint in our software project. Since, in this COVID-19 pandemic situation there has been a lot of traffic on that app, with a lot of increasing demand of users day-by-day. Thus, the extensions must be modelled and delivered to the users as early as possible. The RAD model works well in this stipulated time frame.
- **To gain competitive advantage -** A lot of other products are also flushing into the market, and therefore the RAD model is more advantageous in this aspect as it ensures faster development. Hence, can create a **minimal viable product(MVP)** to gain market lead.
- **Better market feedback -** RAD model ensures and encourages customer feedback and later on helps in project improvement and make additional changes in the future without disturbing the previous extensions.



1. Software Requirement Specification

1.1 Overall Description

1.1.1 Product Functions

- Get Transcript
- Get Summary
- Calculate Attendance
- Get Resources
- Post Notes
- Get Notes

1.1.2 User Characteristics

There will be majorly two types of Users -

- **Teacher(or Meeting Organizer) -** The person who organizes the meeting on the Google Meet platform. He/She must have the knowledge of using the platform and utilize these extensions to the fullest.
- **Students(or Meeting Participants) -** The person who will take part in the Google Meet meetings and must have knowledge about how to access the resources provided by the organizer.

1.1.3 General Constraints

The internet connection and knowledge of Google Meet platform are the constraints to this application. Both the organizer and the participants must have access to decent internet connection and a system(PC or Mobile) to view and upload the contents.

1.1.4 Assumptions and Dependencies

We assumed that -

- the organizer as well as participants are good enough to adapt to the new extensions and utilize those extensions.
- both the organizer and participants must be trained with basic computer skills (like creating/ joining the meeting, access to the resources, validate the attendance record management system)

1.2 External Interface Requirements

1.2.1 User Interfaces

The user must interact with software either through a web application. Users must have a unique Gmail ID.The user will use the same Gmail ID to login the application and then the user will be prompted for the extension.

1.2.2 Hardware Interfaces

Below are the Hardware requirements needed for the software -

CPU: Intel or AMD processor with 64-bit support; *Recommended:* 2.8 GHz or faster processor

Disk Storage: 128 MB of free disk space *Recommended:* 256 MB

Monitor Resolution: 1280x800; *Recommended:* 1920x1080

Internet: Internet connection over 200kbps is required for software usage(WiFi recommended) .

1.2.3 Software Interfaces

To use this software, users must have a Desktop computer system running any operating system(Windows 7 or above is recommended) with good internet connectivity.To be specific any OS 32 bit with network capabilities(*Recommended: any* OS 64 bit).

A browser is needed for interaction. The software can be downloaded as a PWA extension on popular extension libraries(Google store or Mozilla store). Users must have a Google ID and Google Drive logged in with the same ID.

1.3 Functional Requirements

1.3.1 FR 1

Get Transcript: The application records the transcript constantly spoken by the speaker from the captions. The Organizer sends the http request to get the transcript as soon the session ends.

Input- Organizer sends the HTTP Request to get the transcript (the information(data) a lecturer gives during the entire lecture) through the application.

Process- The transcripts saved from the Transcript daemon get stored in Db. On input, the requested Transcript is searched in Db if found then it is sent to the user .

Output- Transcript is shared as a single sharable file of type txt.

1.3.2 FR 2

Get Summary: Students (or participants) request the application to get access to the summary of the lecture.

Input- Http request containing summary id is sent to fetch the required summary from summary database.

Process- Extracts the id from http request. Then, lookup into the summary database, if found the required summary is being fetched else if only transcript is found then, the summary would be generated from summarizer and will be processed further to be sent to the participant.

Output- summary of the required lecture will be shared among the participants.

1.3.3 FR 3

Calculate Attendance: Meeting platform checks for the amount of time a person joined the meeting and using this it estimates the total time a participant attended the meeting and calculates the total time and thereby, marks them as present/absent.

Input- Time fetched from the system's date and time and participant's id being matched.

Process- each participant's time spent during the meeting is being recorded after a particular time interval and using the data fetched, an estimated time is being calculated for each person, if the estimated time is greater than the threshold time then the participant is marked present for the meeting, else absent.

Output- attendance record is being stored in the attendance database.

1.3.4 FR 4

Get Resources: Meeting Organizers share some resources(PDF files, drafts, handwritten notes) during the meeting and want to store it for later use.

Input- Http request containing Resource ID (optional) and meet ID.

Process- Search for all the resources shared in that meet id, organize the resources in the resource database, and store them systematically. Only if resource id exists and matches then send specific resources .

Output- list of all matched resources are shared.

1.3.5 FR 5

Post Notes: Whenever the teacher(or organizer) wants to upload the notes in his/her Google drive, using the post notes function, can store the notes directly into the notes storage database.

Input- input notes from the teacher

Process- convert the individual pages into a single file and store them into a notes database.

Output- file is stored in the database and can be retrieved further.

1.3.6 FR 6

Get Notes: Whenever the teacher(or organizer) wants to retrieve the notes from his/her database, using the get notes function, can retrieve the notes directly from the notes storage database.

Input- input the notes which the teacher wants to retrieve.

Process- search for the required notes in the database, and display it back to the teacher(or organizer).

Output- notes get displayed on the teacher's screen.

1.4 Performance Requirement

Availability:- The software will be available 24/7. Anyone can access it anytime with a good network connection when initiating the Google Meet application. Organizers can opt to not use the extensions by not initializing them within the meeting platform.

Scalability:- The software is highly scalable and can remain stable in terms of performance and efficiency even in higher(200 users/meeting) user traffic conditions. On increasing user demands, the software remains stable while adapting to changes, upgrades and overhauls.

Security:- The software is highly secured with NoSQL database, which provides a highly secure record management system. Data security is a major constraint in every software project, and therefore, safeguarding the user's data(login details, notes and varied resources) through OAuth v2 Authentication. which validates the user on every successful login.

Reliability:- Every function in the software project seeks permission from the organizer. The probability of system failure is minimal(or below 20%).

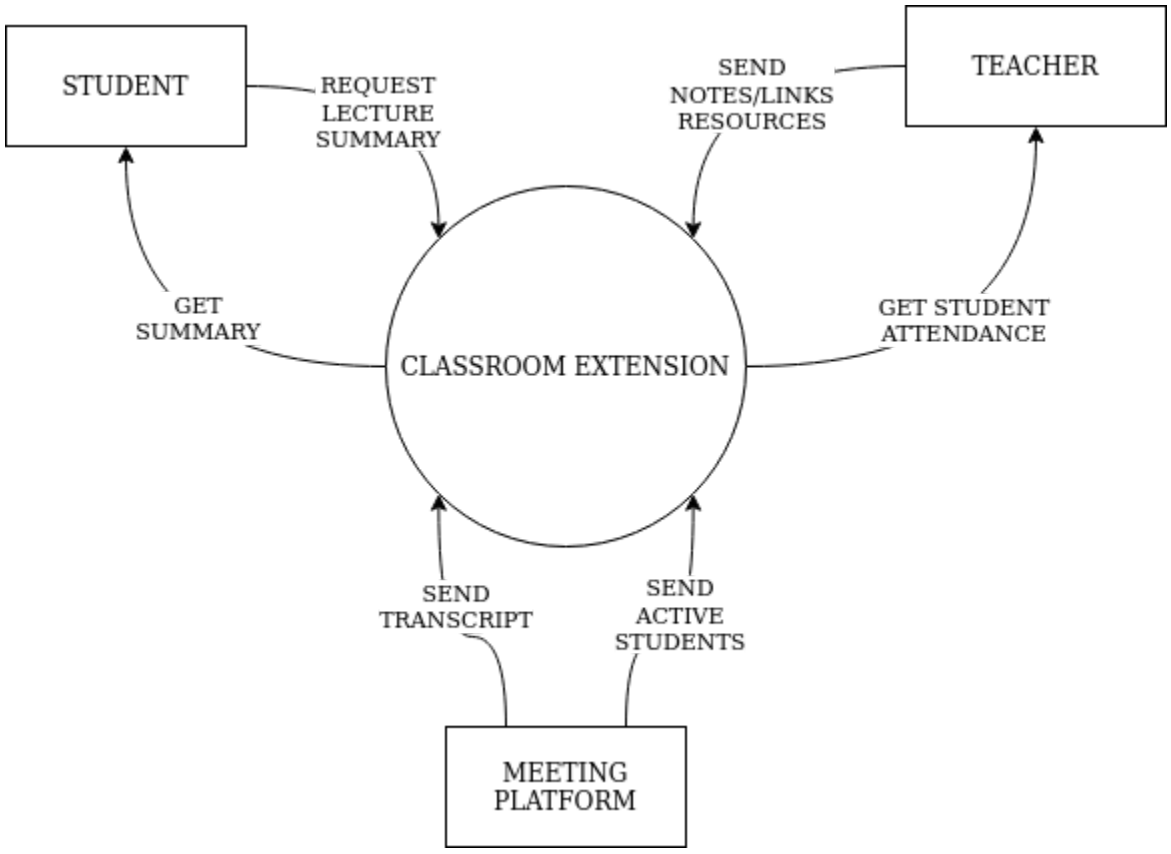
1.5 Design Constraints

Hardware Limitations: It is designed just for Desktop Systems , having a good internet connectivity and with an updated Web Browser.

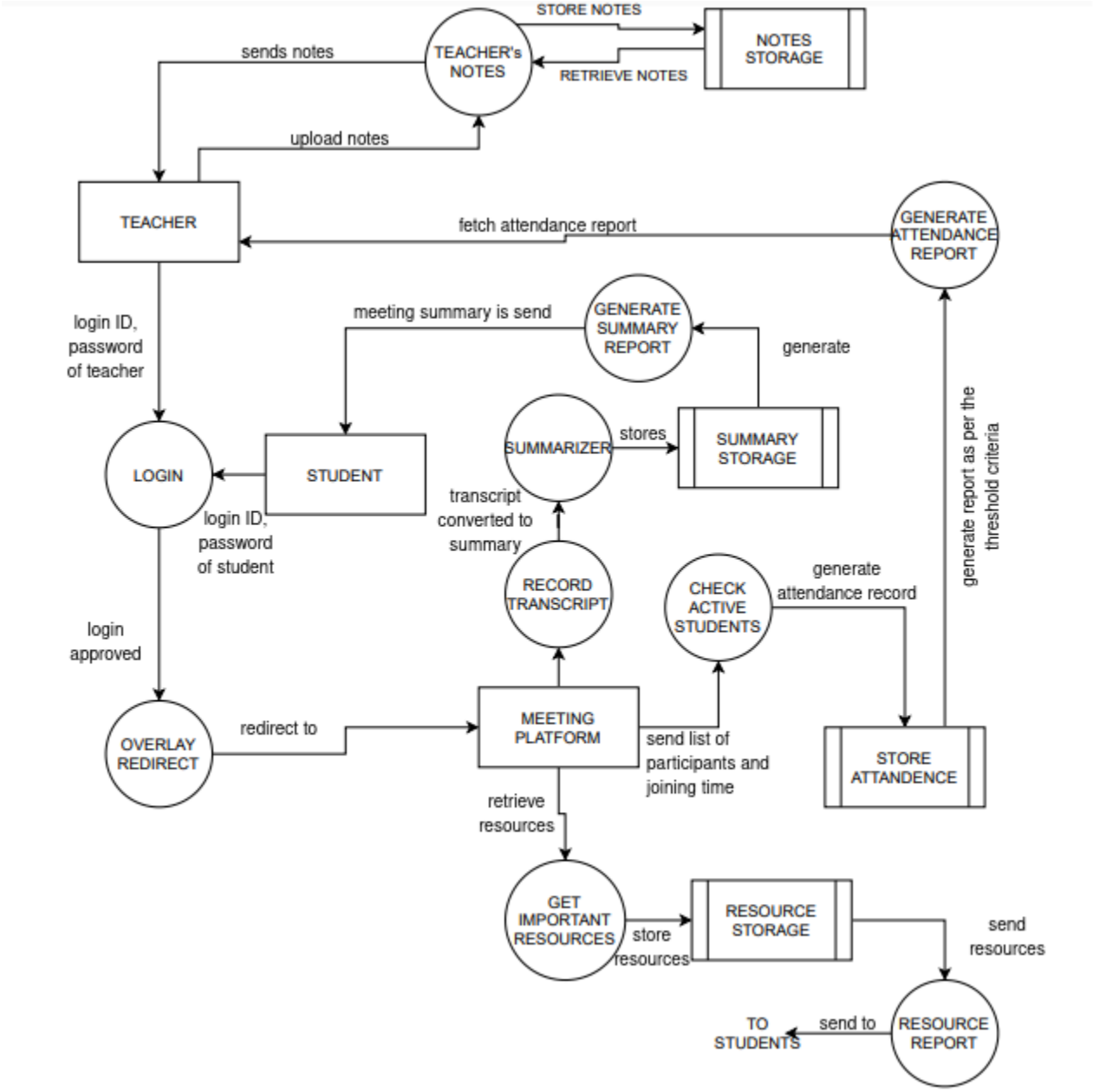
Reliability: Recovery requirements are often an integral part here, detailing what the system should do if some failure occurs to ensure certain properties. Rehability requirements are very important for critical applications.

1.6 Data Flow Diagram

1.6.1 Context Level DFD

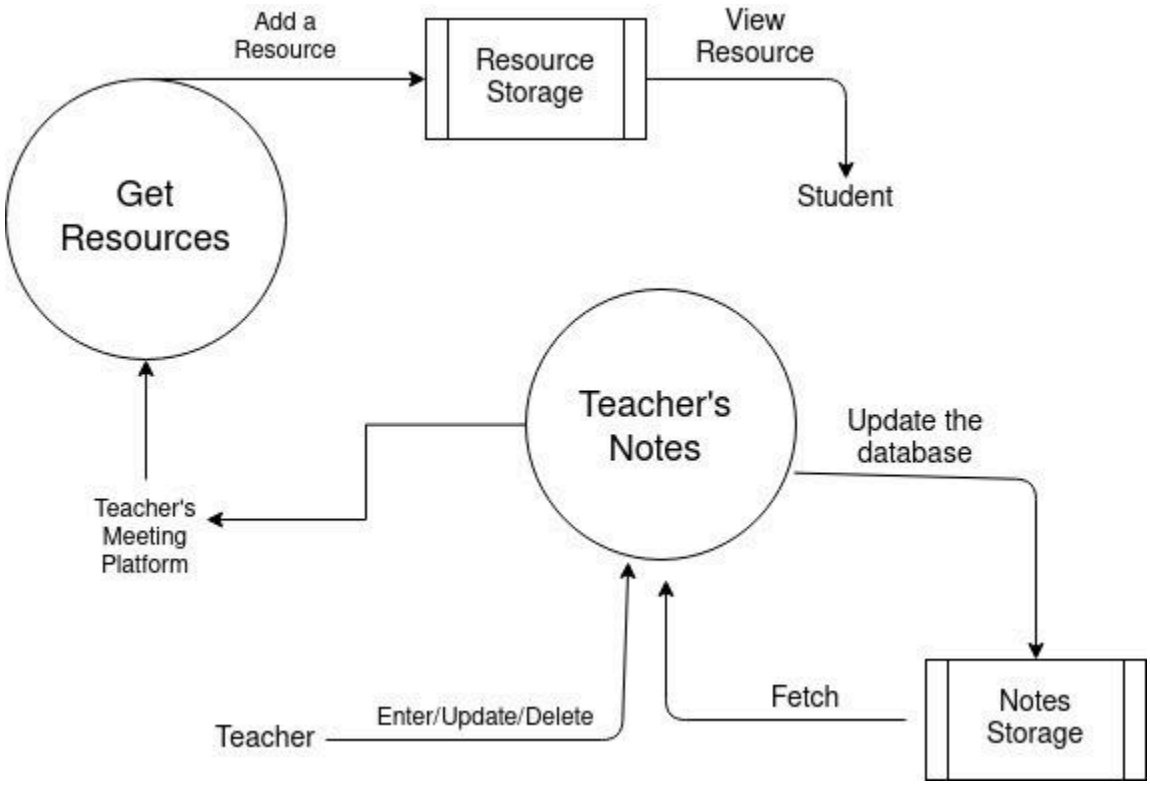


1.6.2 Level 1 DFD

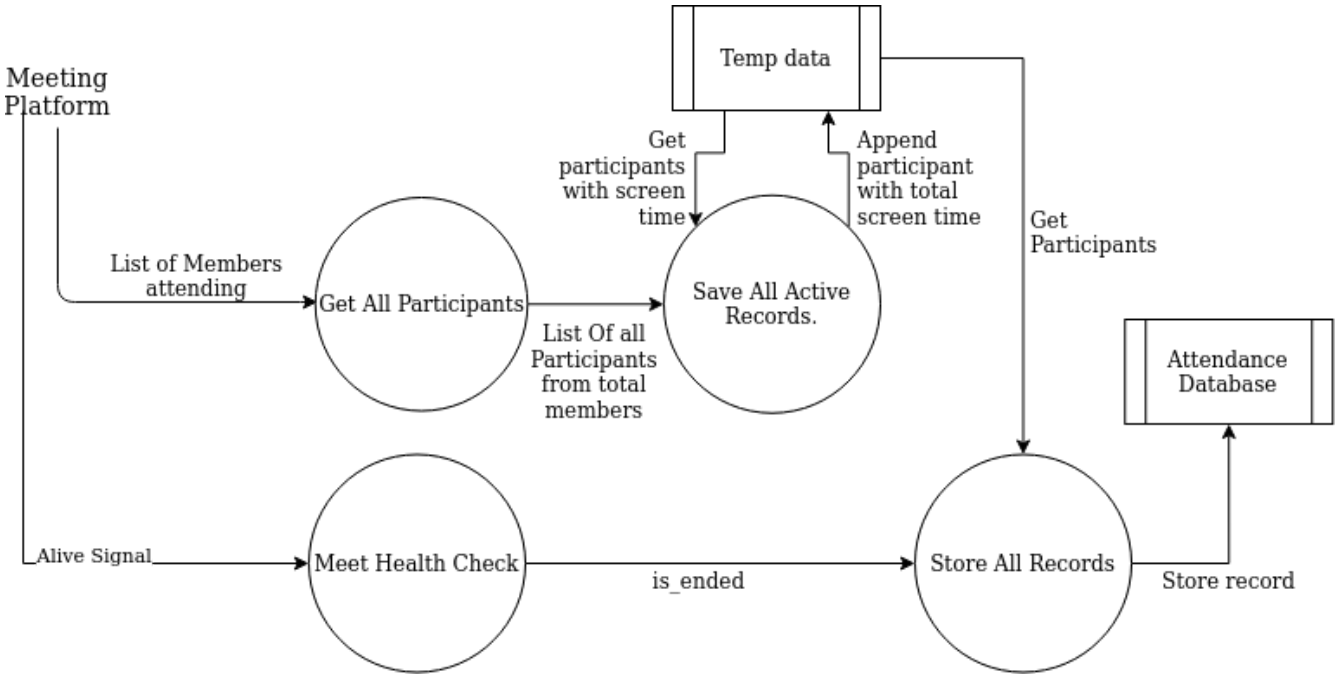


1.6.3 Level 2 DFD

1. **Teacher's Notes**



2. **Attendance Manager**



1.7 Data Dictionary

Student_Status = available(1) | unavailable(0) Is_Teacher_Device = 1[True] | 0 [False] Teacher_Devices = 0 + is_teacher_device $\forall$ device Total_Students = (Σ members) - teacher_device

1.8 Use Cases and Use Case Diagrams

There are basically 3 used cases :-

a. Attendance Calculator
b. Summarizer
c. Resource and Note Management

Use Case 1 : Attendance Calculator

Introduction : It will calculate the attendance record of any person joining the meeting as per the time a person spent in the meeting and calculated result is delivered to the meeting organizer.

Actors : Participants and Meeting organizers are the actors. Participants' records will be fetched and the calculated output is delivered to the organizer.

Pre-condition : Participants must join the meeting, so that his/her time spent in the meeting could be calculated.

Post-condition : Organizer must be there in the meeting, so that the calculated result can be delivered.

Flow of Events : Organizer starts a valid meeting on the platform and starts the attendance calculator event before any participants join the meeting and later on participants join for the meeting. If the participants join the meeting for a specific amount of time, he/she will be marked as present or absent.

Use Case 2 : Summarizer

Introduction : It will summarize the meeting into a specific format, which includes text and/or images, so that anyone can access it, after the meeting ends.

Actors : Participants and Meeting organizers are the actors. Participants and organizers can access a summary of the meeting.

Pre-condition : Organizer must start the summary event before the meeting starts.

Post-condition : Anyone registered for the meeting can request the organizer for the summary and it will be directly sent to the participant’s gmail.

Flow of Events : Organizer starts a valid meeting on the platform and starts the summary event and as soon as the organizer ends the same, a sharable summary is recorded and can be shared with whoever requests it.

Use Case 3 : Resource and Note Management

Introduction : Teachers(organizer) upload the resources(or notes) during the meeting and participants can access those resources later. Participants requests for the resources and resource storage send those resources to the participant.

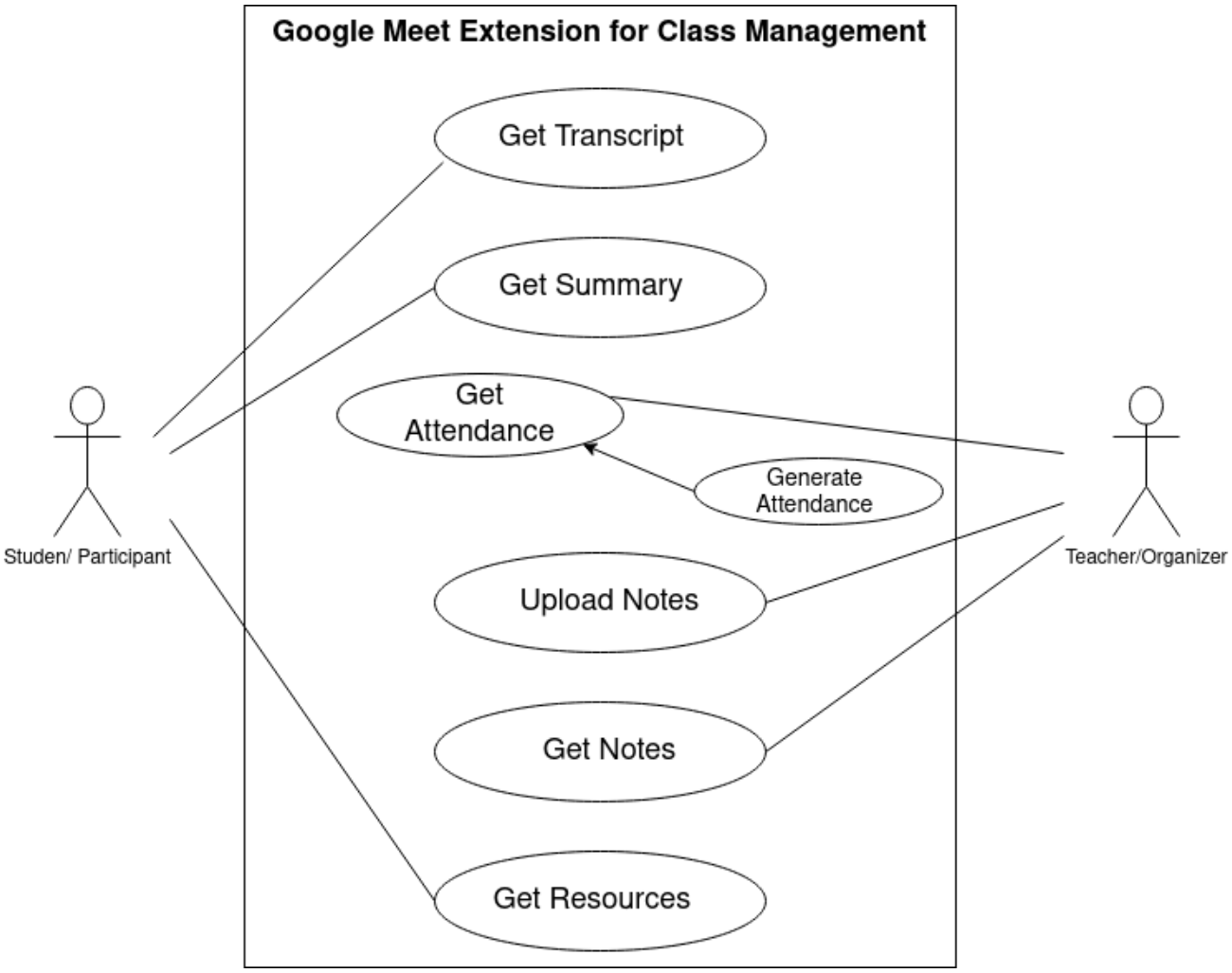
Actors : Participants and Meeting organizers are the actors. Meeting organizers upload the resources and participants can access those resources.

Pre-condition : Organizers must upload the specific resource in the meeting.

Post-condition : Participants accessing a resource must be a valid resource.

Flow of Events : Organizer starts a valid meeting on the platform and uploads the resources and later on, if any participant needs those resources, can request those resources and the resource is checked for validity and if valid, is sent to the participant.

Used Case Diagram :


Google Meet Extension for Class Management
Get Transcript
Get Summary
Get Attendance
Generate Attendance
Upload Notes
Get Notes
Get Resources
Studen/ Participant
Teacher/Organizer

2. Estimations

EIs: External Input (EI) is a transaction function in which Data goes "into" the application from outside the boundary to inside. This data is coming external to the application. Data may come from a data input screen or another application. An EI is how an application gets information.

EOs: External Output (EO) is a transaction function in which data comes "out" of the system. Additionally, an EO may update an ILF. The data creates reports or output files sent to other applications.

EQs: External Inquiry (EQ) is a transaction function with both input and output components that result in data retrieval.

ILFs: Internal Logical File (ILF) is a user identifiable group of logically related data or control information that resides entirely within the application boundary. The primary intent of an ILF is to hold data maintained through one or more elementary processes of the application being counted.

EIFs: External Interface File (EIF) is a user identifiable group of logically related data or control information that is used by the application for reference purposes only. The data resides entirely outside the application boundary and is maintained in an ILF by another application.

We have taken the average values for our Estimations as the overall Complexity of our Software is Average.

2.1 Function Points

Measurement parameters	Count	Weighing factor (Average)	FP Count = Count x Weighing factor
Number of User Inputs	13	4	13x4=52
Number of User Outputs	9	5	9x5=45
Number of User Inquires	3	4	3x4=12
Number of Files	4	10	4x10=40
Number of External Interfaces	2	7	2x7=14
Count Total or **Unadjusted Function Point Count (UFP)**			52+45+12+40+14 = **163**

Thus **UFP = 163**

The **value adjustment factor (VAF)** (also called **Complexity adjustment factor**) is calculated as follows:

$$\mathbf{VAF = [0.65 + 0.01 * \Sigma(F_i)]}$$

Functional Point(FP) Table

S.no	Questions	Value (F_i)
1	Does the system require reliable backup and recovery?	2
2	Are specialized data communications required to transfer information to or from the application?	4
3	Are there distributed processing functions?	1
4	Is performance critical?	4
5	Will the system run in an existing, heavily utilized operational environment?	2
6	Does the system require online data entry?	4
7	Does the online data entry require the input transaction to be built over multiple screens or operations?	2
8	Are the ILFs updated online?	1
9	Are the inputs, outputs, files, or inquiries complex?	2
10	Is the internal processing complex?	4
11	Is the code designed to be reusable?	4
12	Are conversion and installation included in the design?	2
13	Is the system designed for multiple installations in different organizations?	3
14	Is the application designed to facilitate change and ease of use by the user?	4
	$\Sigma(F_i)$	**38**

Given, **Complexity adjustment factor = Value Adjustment Factor (VAF)**

$= [0.65 + 0.01 * \Sigma(F_i)] = [0.65 + 0.01 * 38]$

$= 1.03$

Adjusted FP Count = Unadjusted FP Count × VAF

$= 163 \times 1.03 = 167.89$

Hence, Adjusted FP Count or $FP_{estimated}$ = 168 (approx.)

2.2 EFFORTS

$FP_{estimated}$ = 168

We assume that,

Average Productivity = 2 FP per person month

Labour Rate = ₹ 35,000 per month

From the above assumptions:

Estimated Effort = $FP_{estimated}$ / Average Productivity

= 168 FP / (2 FP per pm) = **84 pm**

Cost Per FP = Labour Rate / Average Productivity

= Rs 35, 000 pm / (2 FP per pm) = Rs 17,500 per FP

Total Estimated Project Cost = Estimated Effort * Labour Rate

= 84 pm * Rs 35,000 pm

= Rs. 29,40,000

Or

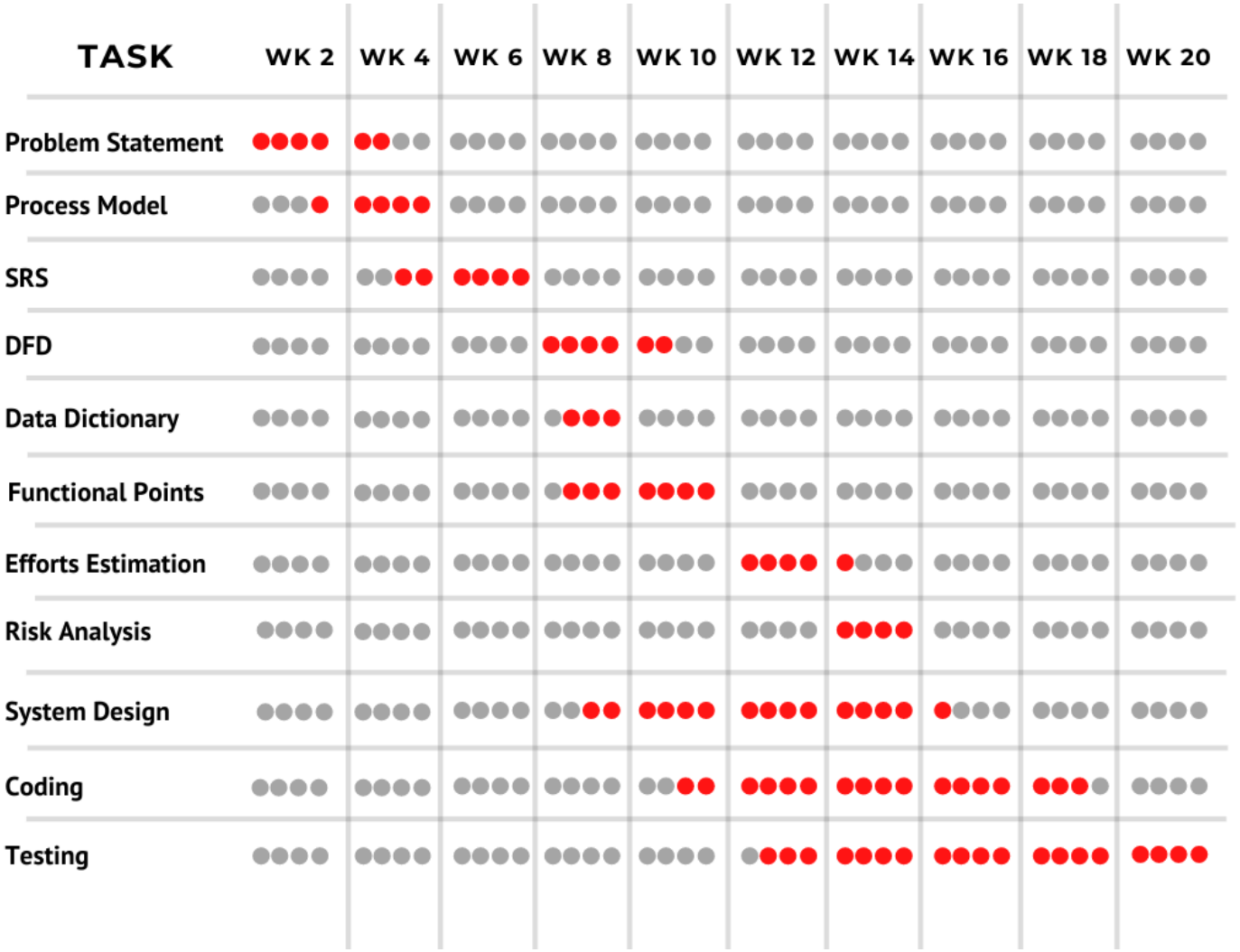
Total Estimated Project Cost = $FP_{estimated}$ * Cost Per FP

= 168 * Rs 17,500 per FP

= Rs. 29,40,000

Thus, the Total estimated project cost is Rs. 29,00,000 and the estimated effort is 84 person months.

3. Scheduling



4. Risk Management

4.1 Risk Table

After analysing our project, we found that there is a possibility that four categories of risk would affect the product in the upcoming days.
A risk table provides us with a simple technique for risk projection. It will show the probability of the occurrence of those risk and the impact it would have on our project .
The Risk Table for our project :

S. No	Risks	Category	Probability	Impact
1.	Inadequate Funding	CU	60%	**2**
2.	Loss of Database	PS	55%	**2**
3.	Larger Number of users than expected	PS	50%	**3**
4.	Delivery Deadline	BU	40%	**3**
5.	Technology will not expectations	TE	30%	**3**
6.	Lack of awareness about online tools	BU	20%	**4**

Abbreviations
PS – Product Size
BU – Business Risk
CU – Stakeholder Characteristics
TE – Development Environment

Impact Values
1 – Catastrophic
2 – Critical
3 – Marginal
4 – Negligible

4.2 RMMM (Risk Mitigation, Monitoring and Management)

Mitigation:- It is the strategy adopted by the software team for avoiding the risk.

Monitoring:- It is the strategy adopted by the product manager to monitor factors that may provide an indication of whether the risk is becoming more or less likely.

Management:- It comes into play after the future of the Mitigation strategy, when risks becomes the reality. It helps to reduce the impact of the risk on the project.

In our project there are 3 risks that lie above the cutoff line. The cutoff line for our project was 50% and above in terms of probability.

The table below shows the mitigation handling strategies for the risks above the cutoff line.

Risk Impact	Risks	Mitigation
1.	Inadequate Funding	Dividing the funds into several parts and by making a reserve fund to use it when we run out of funds.
2.	Loss of Database	Creating a backup for the previous/used databases.
3.	Larger Number of users than expected	Working more on its scalable design to serve more users.

4.3 Risk Exposure

The overall risk exposure RE is determined using the following relationship
RE= P * C ,
where P is the probability of occurrence for a risk, and C is the cost to the project if the risk occur.

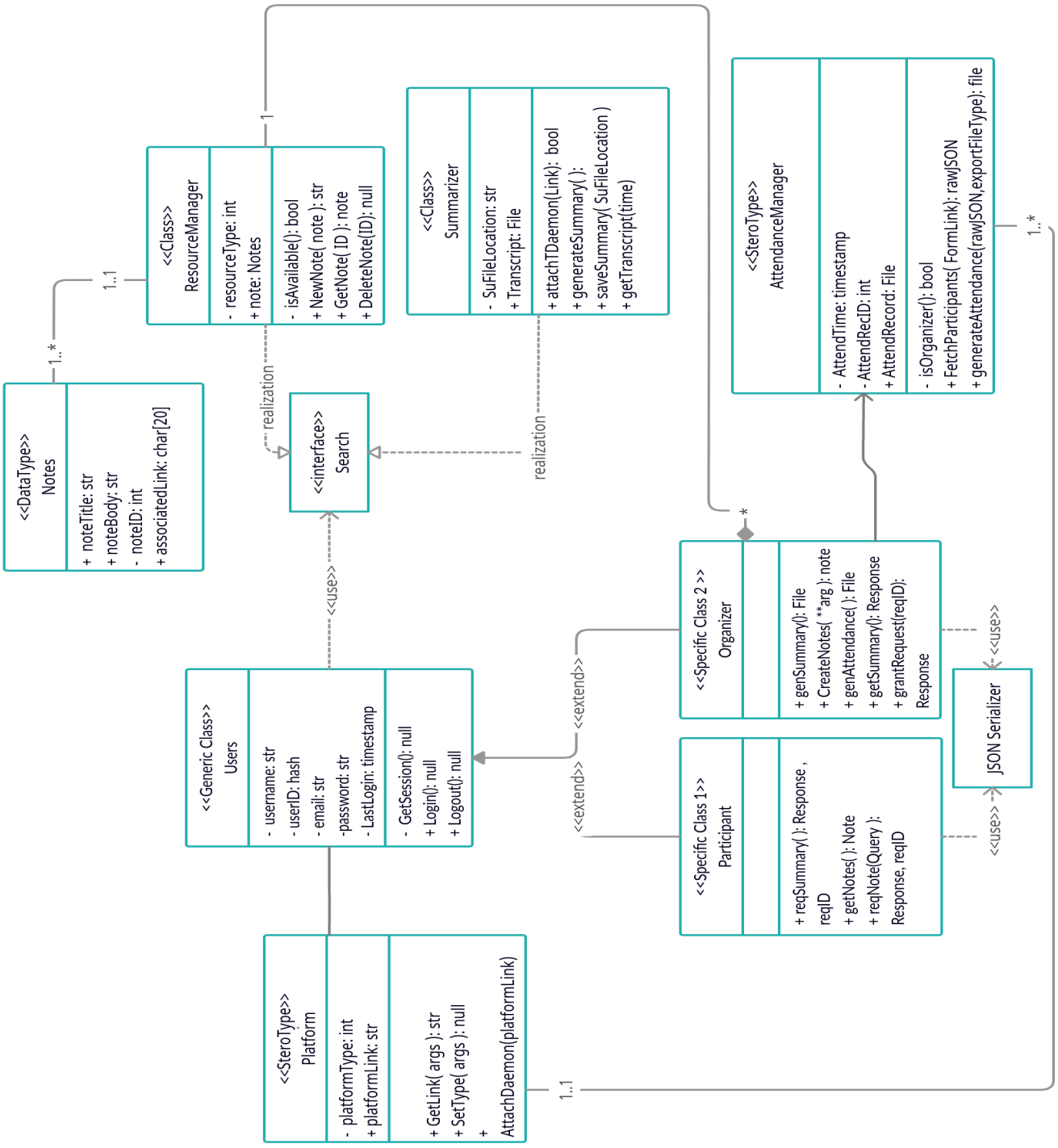
1. ***Risk Identification:*** There will be lack of funding for the development of this project.
2. ***Risk Probability:*** 60% (likely)
3. ***Risk Impact:*** In order to finish the project before the deadline, the Dev. Team will require Rs. 1,40,000.
4. ***Risk Exposure :***

   RE= P * C

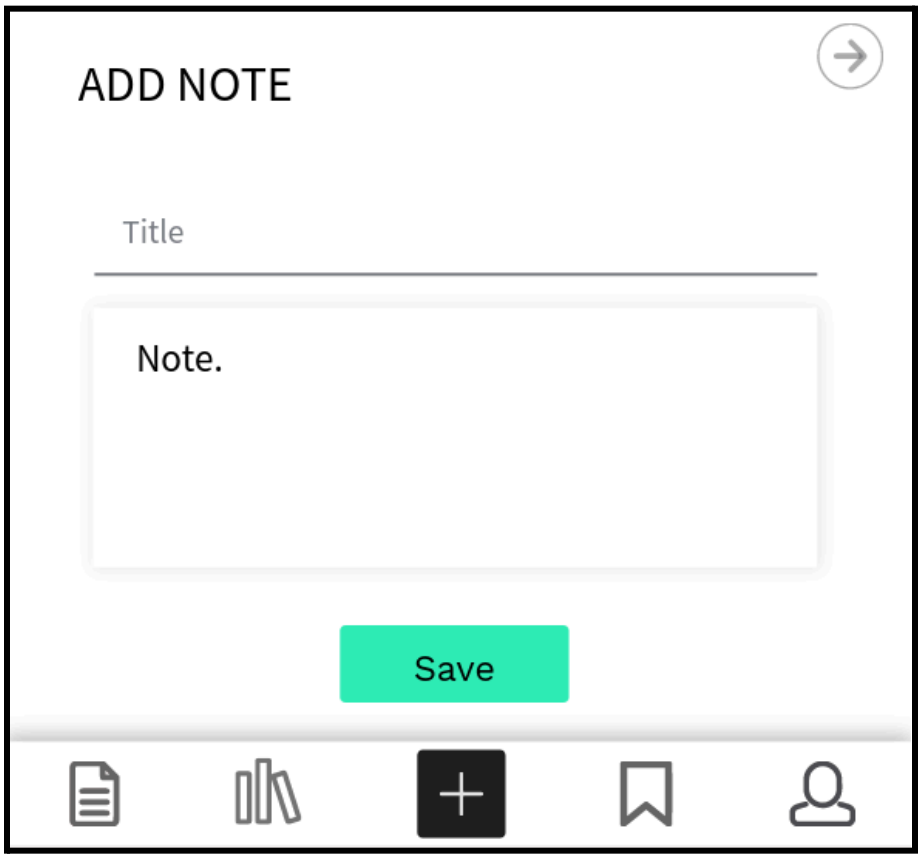
   RE= 0.60* 1,40,000 ≈ Rs. 84,000

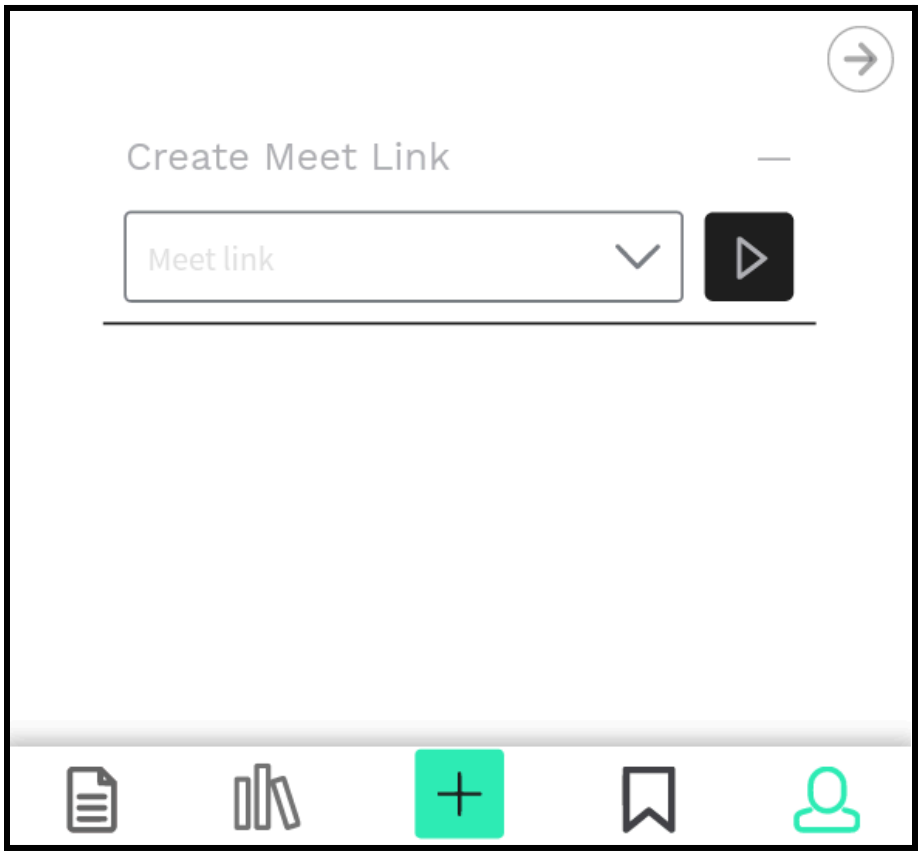
5. Design

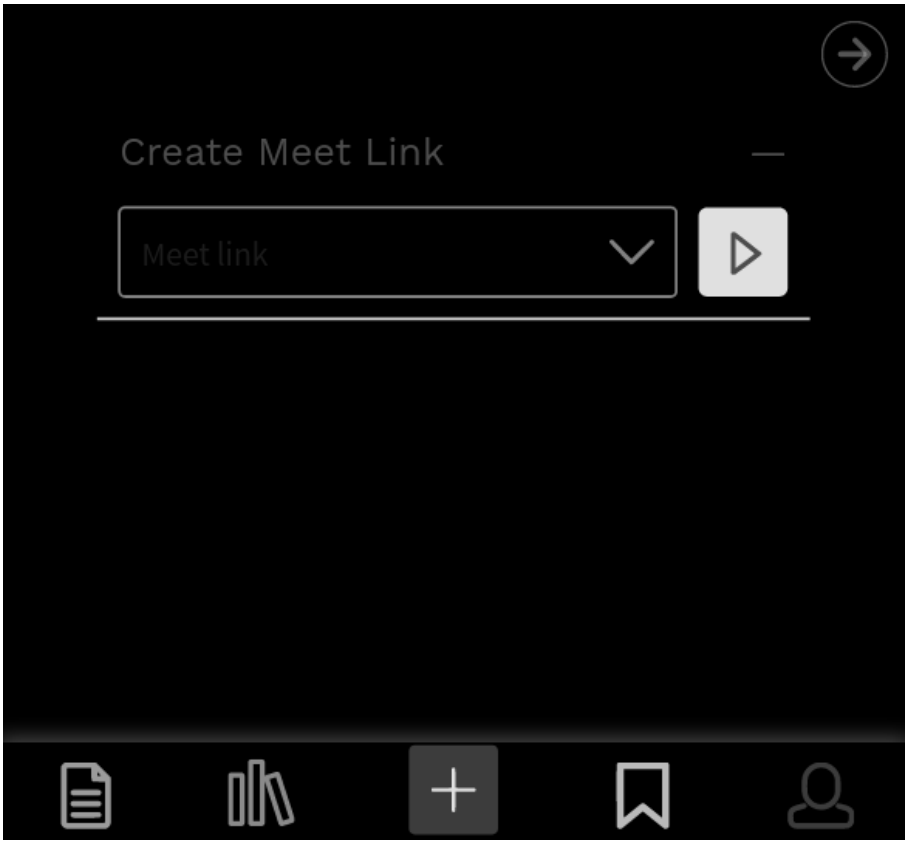
5.1 System Design

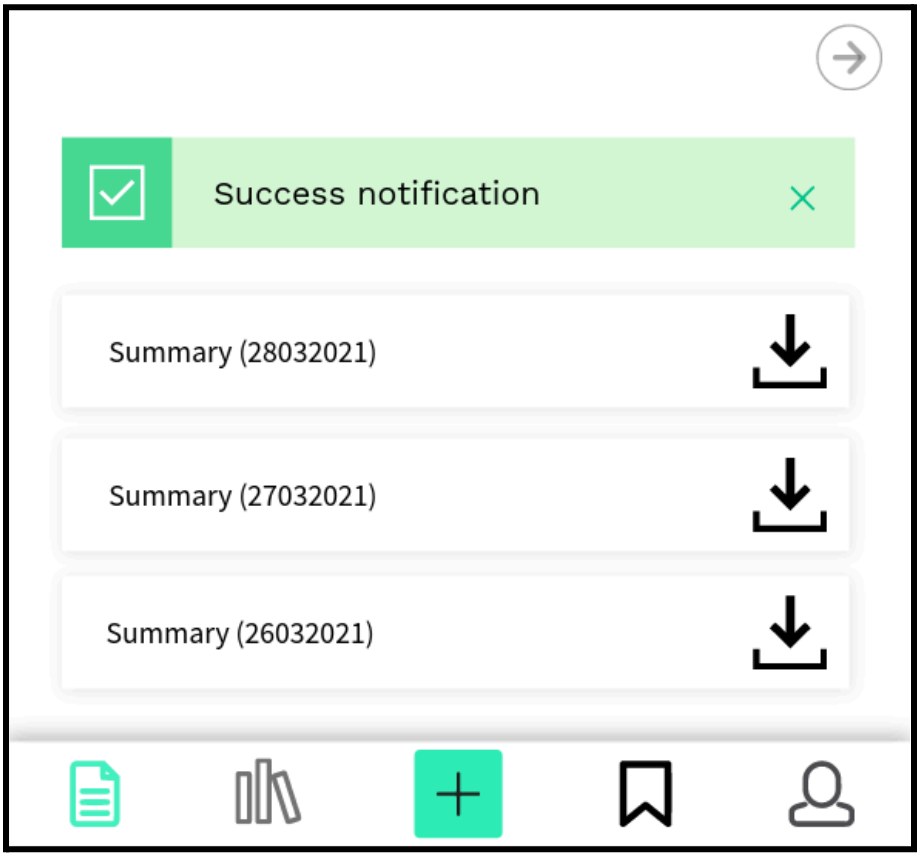


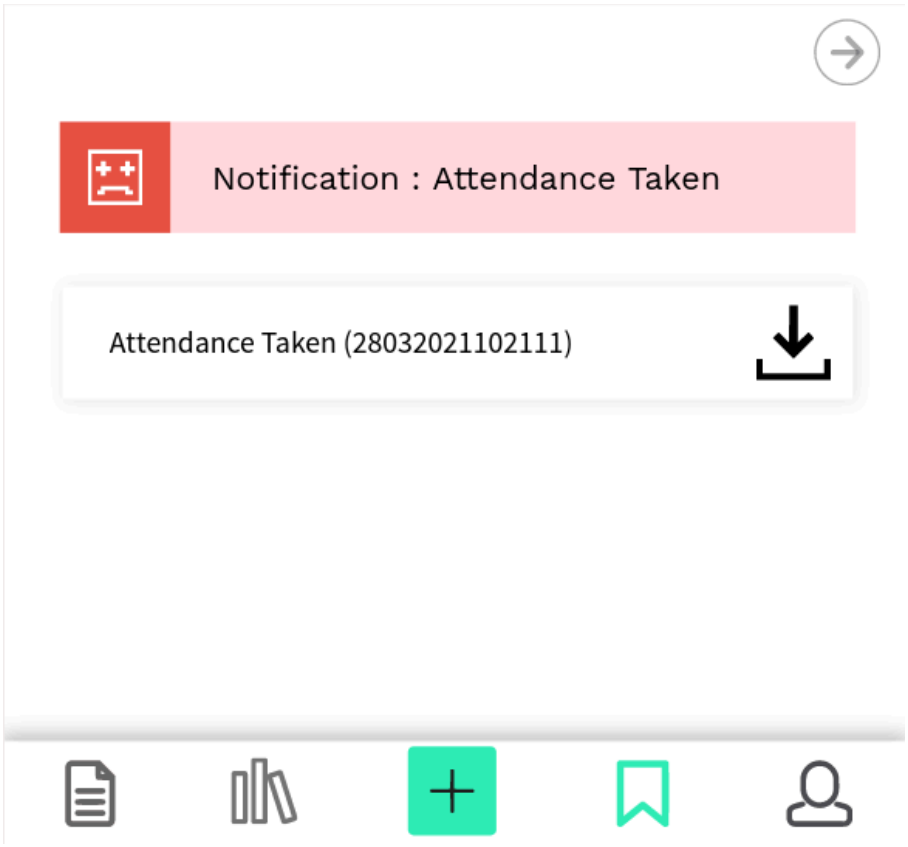
5.2 Screen Design

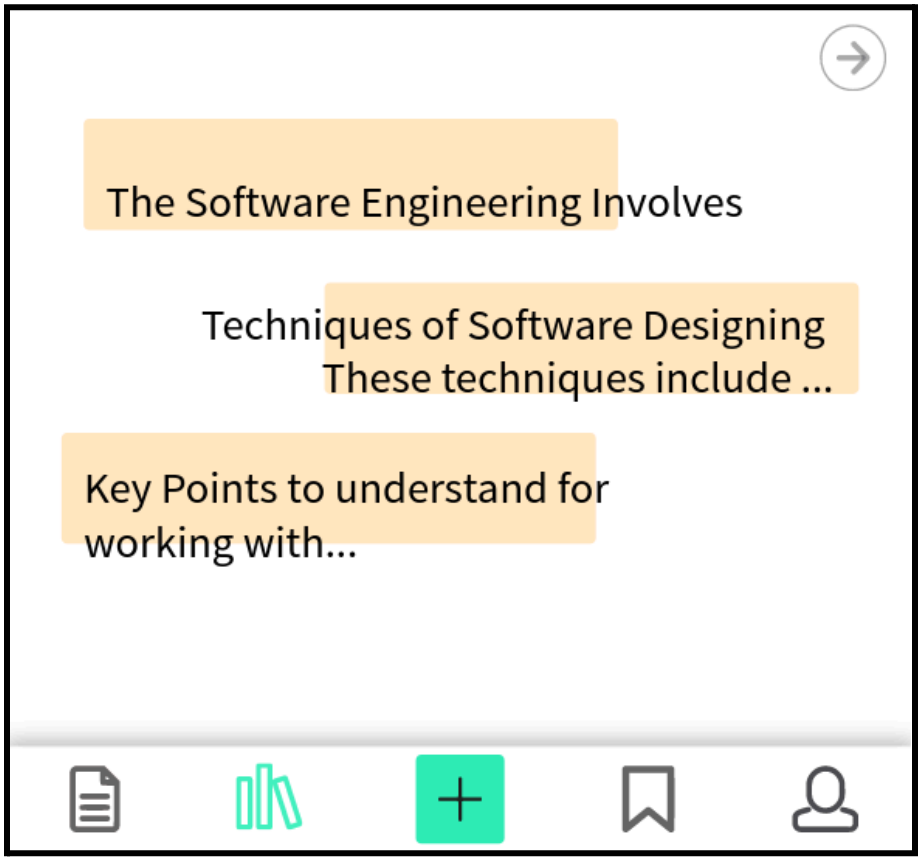




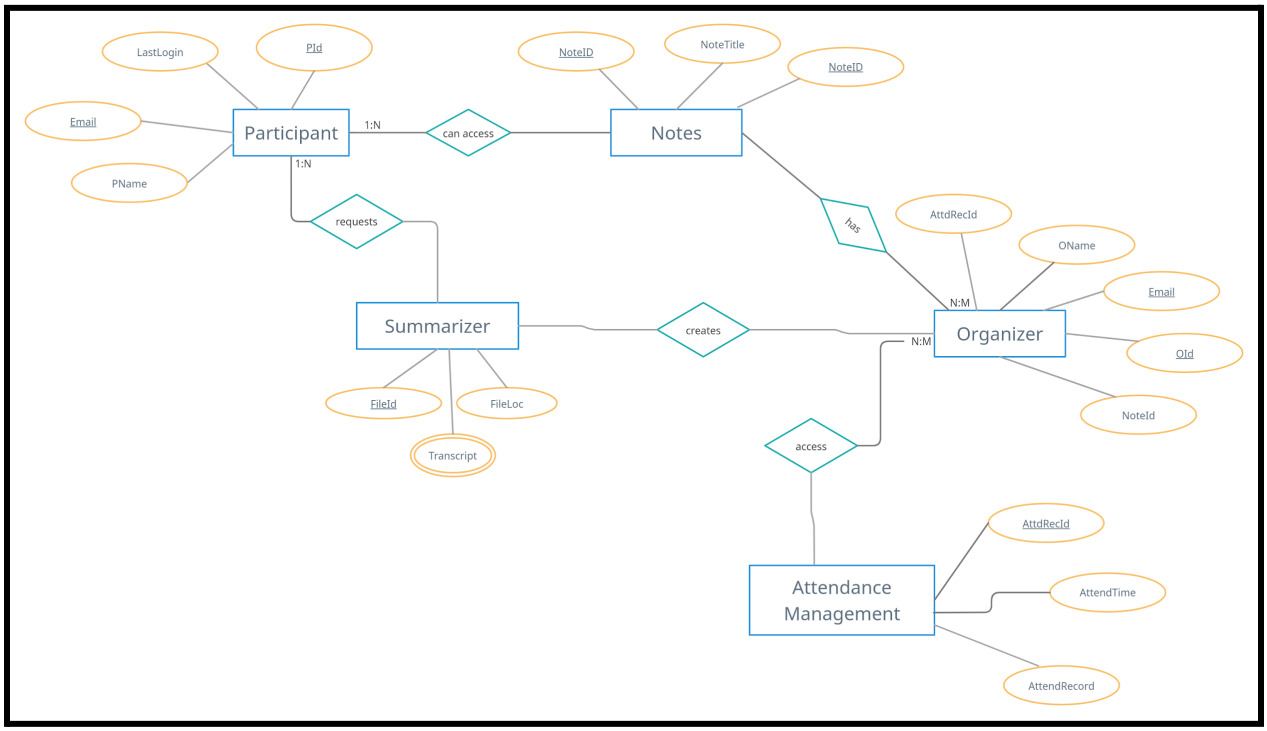

Create Meet Link
Meet link


Success notification
Summary (28032021)
Summary (27032021)
Summary (26032021)

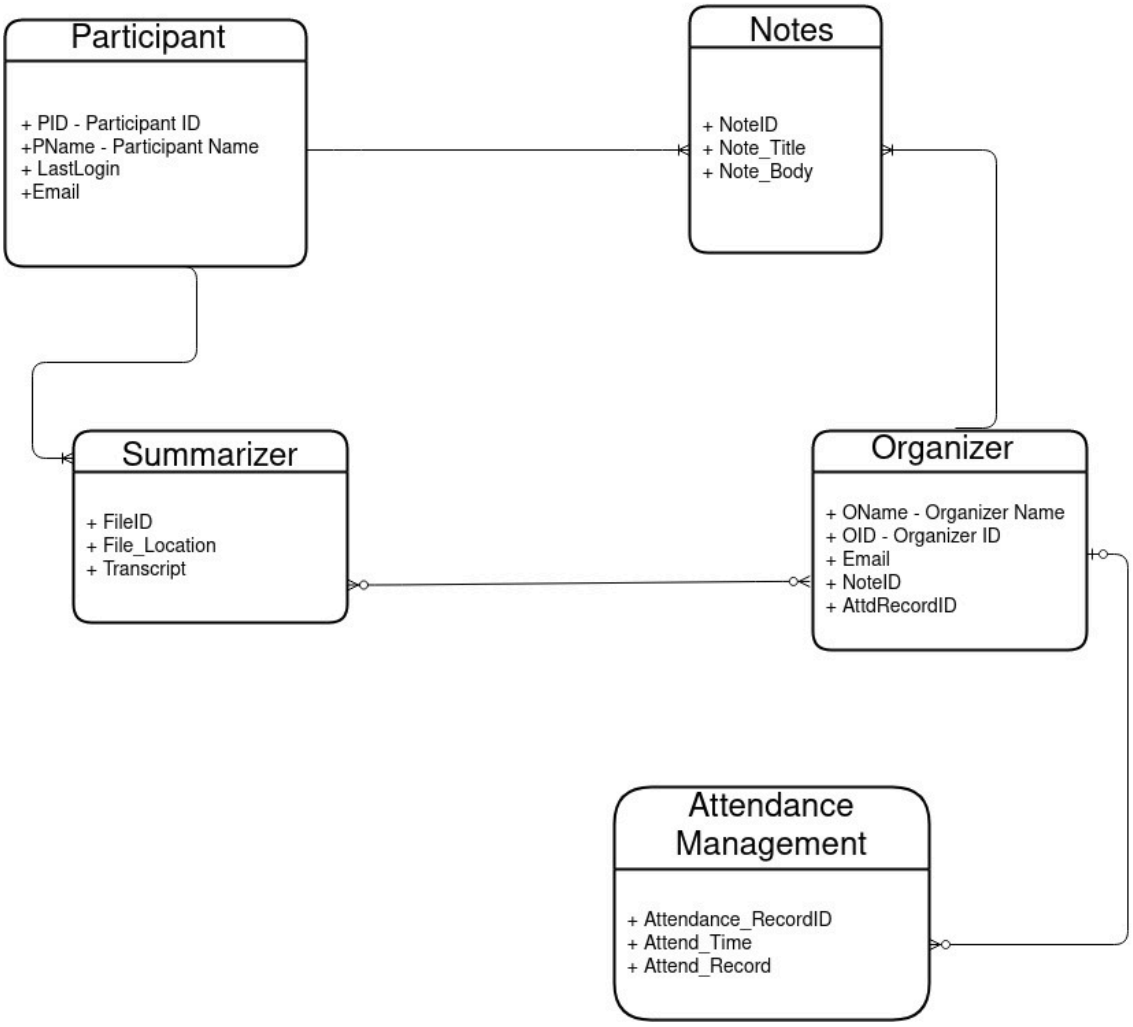

Notification : Attendance Taken
Attendance Taken (28032021102111)


The Software Engineering Involves
Techniques of Software Designing
These techniques include ...
Key Points to understand for
working with...

5.3 ER Diagram



5.4 Database Design



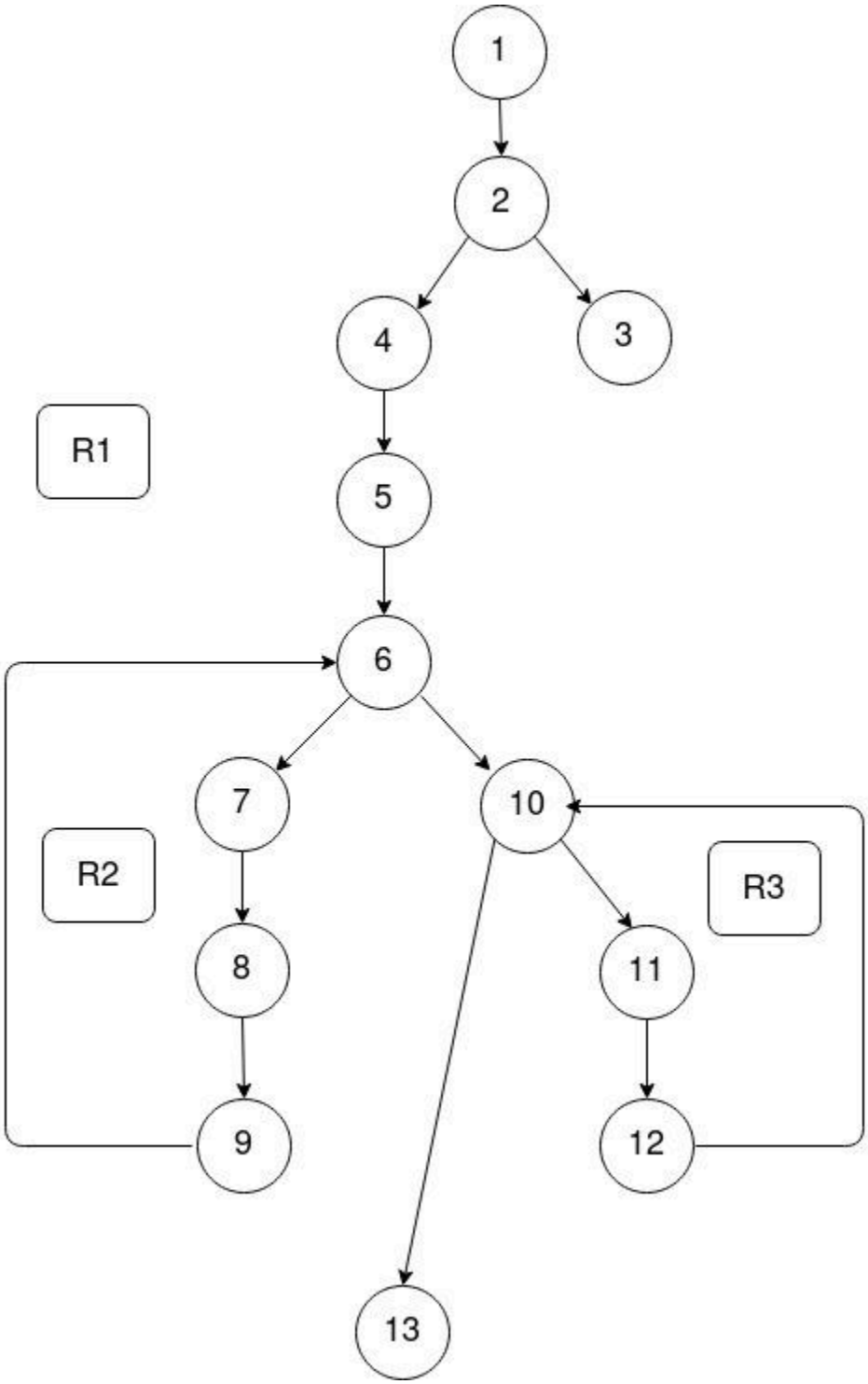
6. Coding

A sample code-segment for login, coded in Python -

```
[1]def login(name, password):

#getLoggedInUser() checks whether the user is already logged in or not

 [2]   if (getLoggedInUser() != NULL):
 [3]       raise Exception("Already Logged In")
 [4]  login = False
 [5]   user = User()
 [6]   for (user in getUser()):

    #checks whether the entered username and password matches with the record
    #if it matches, then login is granted!!

 [7]       if(user.name_password == (name+'.#.'+password)):
 [8]                 loginGranted = user
 [9]                 login = true

    #if login fails, it raises an exception

 [10]  if(!login):
 [11]     for (user in getUser()):
 [12]                 raise Exception("Incorrect User name or Password!! Retry.")
 [13]  return (login)
```

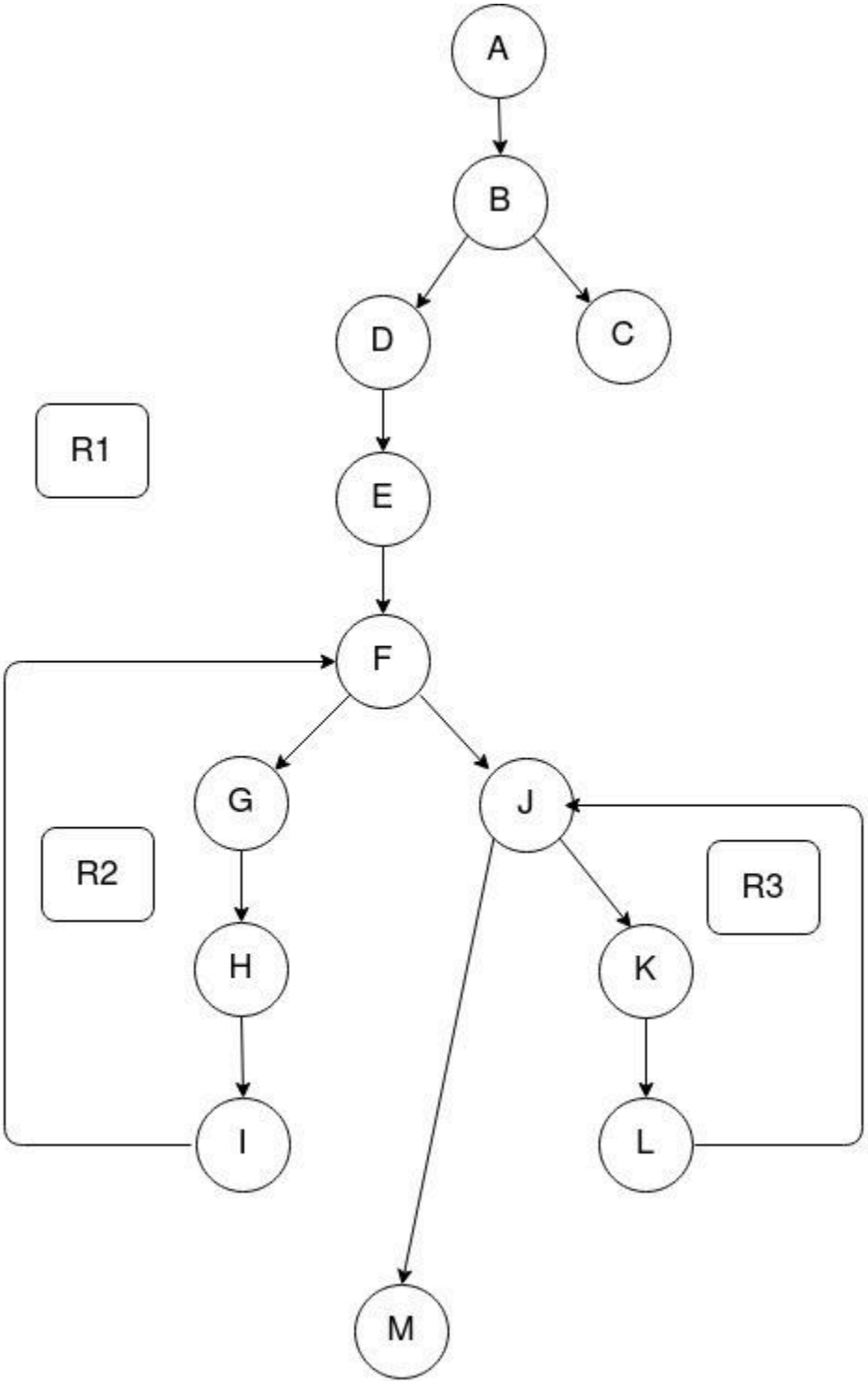
7. Testing

7.1 Flowgraph



7.2 DD Path Graph

Note - Since all lines of code in the flow graph(ie. 1 to 13) are individual nodes, therefore, 1 to 13 is depicted as ‘A’ to ‘M’ in the DD Path Graph. In other words, we can say that, 1 denote ‘A’, 2 denote ‘B’ and so on 13 denote ‘M’.



7.3 Cyclomatic Complexity

Cyclomatic complexity can be calculated using the formulas -

(a) E - N + 2*P

where, E denotes number of edges

N denotes number of nodes

P denotes number of disconnected graphs

(b) Total number of regions

Total regions = number of bounded region + number of unbounded region

(c) P + 1

where, P denotes number of predicates

Now, calculating Cyclomatic Complexity -

number of edges = 14

number of nodes = 13

number of disconnected graphs = 1

number of predicates = 2

total regions = 3

(a) E - N + 2*P = 14 - 13 + (2 * 1) = 14 -13 + 2 = 3

(b) Total regions = 3

(c) P + 1 = 2 + 1 = 3

Therefore, cyclomatic complexity = 3

7.3.1 Independent Paths

Various Independent paths possible -

(a) A → B → C

(b) A → B → D → E → F → G → H → I → F → J → M

(c) A → B → D → E → F → J → K → L → J → M

7.3.2 Regions

Total regions = 3

1 unbounded region(defined as R1 in the diagram) and 2 bounded regions

2 bounded regions are -

(a) F → G → H → I → F (defined as R2 in the diagram)

(b) J → K → L → J (defined as R3 in the diagram)

7.4 Test Cases

7.4.1 Test Case 1

Path : A → B → C

Input Value : Username and Password

Title : Sign In

Description : In this step, entered userID is checked whether it's already logged in the software or not. If the user is already logged in then the function ends, else it passes to line 4.

7.342 Test Case 2

Path : A → B → D → E → F → G → H → I → F → J → M

Input Value : Username and Password

Title : Sign-in approved

Description : In this step, entered userID and Password is matched with the existing userID and Password. All the existing userID and password are stored in a special format (ie. userID.#.password), when this string matches with the entered ID and password, login is approved and function ends.

7.4.3 Test Case 3

Path : A → B → D → E → F → J → K → L → J → M

Input Value : Username and Password

Title : Sign-in NOT approved

Description : In this step, entered userID and Password is matched with the existing userID and Password, when login failed due to incorrect userID and/or password, an exception is raised and function ends.

8. Future Scope

The following would be potential features that could be involved in the future releases.

1. Chat integration for real-time communication while screen sharing.
2. Interactive Quiz with polls for class interaction.
3. Class test monitoring capabilities.

9. References

[1] Pressman, Software Engineering: A Practitioner's Approach, 8th edition, 2012.
[2] Aggarwal, K. K., & Singh, Y. (2007). Software Engineering. 3rd edition.
[3] Airbreak, Rapid Application Development(RAD), 2016

10. Appendix

10.1 Rapid Application Development (RAD)

Definition: The Rapid Application Development (or RAD) model is based on prototyping and iterative models with no (or less) specific planning. In general, RAD approach to software development means putting less emphasis on planning tasks and more emphasis on development and coming up with a prototype. In disparity to the waterfall model, which emphasizes meticulous specification and planning, the RAD approach means building on continuously evolving requirements, as more and more learnings are drawn as the development progresses.

Description:

Below phases are in rapid application development (RAD) model

1. Business modeling: The information flow is identified between different business functions.

2. Data modeling: Information collected from business modeling is used to define data objects that are required for the business.

3. Process modeling: Data objects defined in data modeling are converted to establish the business information flow to achieve some specific business objective process descriptions for adding, deleting, modifying data objects that are given.

4. Application generation: The actual system is created and coding is done by using automation tools. This converts the overall concept, process and related information into actual desired output. This output is called a prototype as it's still half-baked.

5. Testing and turnover: The overall testing cycle time is reduced in the RAD model as the prototypes are independently tested during every cycle.

However, the overall flow of information, user interfaces and other program interfaces, and coaxials between these interfaces and the rest of data flow need to be tested as per the acceptance process. Since most of the programming components have already been tested, it reduces the risk of any critical issue.

10.2 Code for Login Section UI

```
from PyQt5 import QtCore, QtGui, QtWidgets

class Ui_Widget(object):
    def setupUi(self, Widget):
        Widget.setObjectName("Widget")
        Widget.resize(400, 300)
        self.Home = QtWidgets.QPushButton(Widget)
        self.Home.setGeometry(QtCore.QRect(180, 260, 31, 29))
```

```
        self.Home.setObjectName("Home")
        self.frame_2 = QtWidgets.QFrame(Widget)
        self.frame_2.setGeometry(QtCore.QRect(0, 249, 401, 51))
        self.frame_2.setFrameShape(QtWidgets.QFrame.StyledPanel)
        self.frame_2.setFrameShadow(QtWidgets.QFrame.Raised)
        self.frame_2.setObjectName("frame_2")
        self.back = QtWidgets.QCommandLinkButton(Widget)
        self.back.setGeometry(QtCore.QRect(370, 0, 31, 31))
        self.back.setObjectName("back")
        self.Summary = QtWidgets.QPushButton(Widget)
        self.Summary.setGeometry(QtCore.QRect(20, 260, 31, 29))
        self.Summary.setObjectName("Summary")
        self.Attendance = QtWidgets.QPushButton(Widget)
        self.Attendance.setGeometry(QtCore.QRect(260, 260, 31, 29))
        self.Attendance.setObjectName("Attendance")
        self.Settings = QtWidgets.QPushButton(Widget)
        self.Settings.setGeometry(QtCore.QRect(340, 260, 31, 29))
        self.Settings.setObjectName("Settings")
        self.Notes = QtWidgets.QPushButton(Widget)
        self.Notes.setGeometry(QtCore.QRect(100, 260, 31, 29))
        self.Notes.setObjectName("Notes")
        self.Login = QtWidgets.QPushButton(Widget)
        self.Login.setGeometry(QtCore.QRect(260, 190, 71, 29))
        self.Login.setObjectName("Login")
        self.PasswordField = QtWidgets.QLineEdit(Widget)
        self.PasswordField.setGeometry(QtCore.QRect(61, 140, 271, 30))
        self.PasswordField.setObjectName("PasswordField")
        self.username = QtWidgets.QLabel(Widget)
        self.username.setGeometry(QtCore.QRect(61, 40, 71, 22))
        self.username.setObjectName("username")
        self.UsernameField = QtWidgets.QLineEdit(Widget)
        self.UsernameField.setGeometry(QtCore.QRect(61, 60, 271, 30))
        self.UsernameField.setObjectName("UsernameField")
        self.Password = QtWidgets.QLabel(Widget)
        self.Password.setGeometry(QtCore.QRect(61, 110, 71, 22))
        self.Password.setObjectName("Password")

        self.retranslateUi(Widget)
        QtCore.QMetaObject.connectSlotsByName(Widget)

    def retranslateUi(self, Widget):
        _translate = QtCore.QCoreApplication.translate
        Widget.setWindowTitle(_translate("Widget", "Form"))
        self.Home.setText(_translate("Widget", "+"))
        self.back.setText(_translate("Widget", "CommandLinkButton"))
        self.Summary.setText(_translate("Widget", "H"))
        self.Attendance.setText(_translate("Widget", "A"))
        self.Settings.setText(_translate("Widget", "S"))
        self.Notes.setText(_translate("Widget", "N"))
        self.Login.setText(_translate("Widget", "Login"))
        self.username.setText(_translate("Widget", "Username"))
```

```
                self.Password.setText(_translate("Widget", "Password"))


if __name__ == "__main__":
        import sys
        app = QtWidgets.QApplication(sys.argv)
        app.setStyle('stylesMeetExtension')
        Widget = QtWidgets.QWidget()
        ui = Ui_Widget()
        ui.setupUi(Widget)
        Widget.show()
        sys.exit(app.exec_())
```