

B.Sc. in Computer Science and Engineering Thesis

## **Automated Meeting Scheduling and Document Management System**

Submitted by

Sirajum Munira  
201414010

Tasbiraha Athaya  
201414014

Afsana Zaman  
201414042

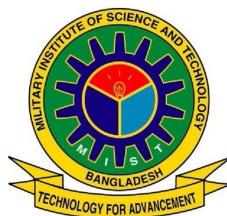
Supervised by

Dr. Syed Akhter Hossain

Professor

Head of the Department Of Computer Science & Engineering

Daffodil International University



**Department of Computer Science and Engineering  
Military Institute of Science and Technology**

December 2017

# **CERTIFICATION**

This thesis paper titled **"Automated Meeting Scheduling and Document Management System"**, submitted by the group as mentioned below has been accepted as satisfactory in partial fulfillment of the requirements for the degree B.Sc. in Computer Science and Engineering in December 2017.

## **Group Members:**

1. Sirajum Munira
2. Tasbiraha Athaya
3. Afsana Zaman

## **Supervisor:**

---

**Professor Dr. Syed Akhter Hossain**  
Head of CSE Department  
Daffodil International University

## **CANDIDATES' DECLARATION**

This is to certify that the work presented in this thesis paper, titled, "**Automated Meeting Scheduling and Document Management System**", is the outcome of the investigation and research carried out by the following students under the supervision of Dr. Syed Akhter Hossain, Professor, Head of CSE Department, Daffodil International University.

It is also declared that neither this thesis paper nor any part thereof has been submitted anywhere else for the award of any degree, diploma or other qualifications.

---

Sirajum Munira  
201414010

---

Tasbiraha Athaya  
201414014

---

Afsana Zaman  
201414042

## ACKNOWLEDGEMENT

We are thankful to Almighty Allah for his blessings for the successful completion of our thesis. Our heartiest gratitude, profound indebtedness and deep respect go to our supervisor, **Dr. Syed Akhter Hossain**, Professor, Head of CSE Department, Daffodil International University, for his constant supervision, affectionate guidance and great encouragement and motivation. His keen interest on the topic and valuable advices throughout the study was of great help in completing thesis.

We are especially grateful to the Department of Computer Science and Engineering (CSE) of Military Institute of Science and Technology (MIST) for providing their all out support during the thesis work.

Finally, we would like to thank our families and our course mates for their appreciable assistance, patience and suggestions during the course of our thesis.

Dhaka

December 2017 .

1. Sirajum Munira

2. Tasbiraha Athaya

3. Afsana Zaman

## **ABSTRACT**

Meeting is an important part of our professional life which takes place to take important decision, to discuss about issues and multiple confidential documents are managed. It is an event in which a group of people come together to discuss things or make decisions. To perform all these tasks for a meeting, there are different types of approaches, methods and techniques are used in organizations. To keep track of meetings and to store all the meeting files accordingly, we represent a new idea of meeting scheduling system where all the related files of the meeting will be stored as per corresponding meeting with security. Our aim is to develop a software that can automatically schedule meeting by using necessary information and make a secure system to preserve all the documents related to the meeting. We are going to build this system thorough some point of view. Firstly we have maintained a meeting List in a sorted way and schedule a meeting without conflict of time. If there is no generate available time automatically. Secondly we have gave the option to upload files of the meetings. To maintain the security of the files, we have the encode and decode system for encryption and decryption. Our system can give notice of scheduling, rescheduling and cancellation of the meetings automatically.

# Contents

<i>CERTIFICATION</i>	ii
<i>DECLARATION</i>	iii
<i>ACKNOWLEDGEMENT</i>	iv
<i>ABSTRACT</i>	1
<b>1</b> <i>Introduction</i>	8
1.1 <i>Introduction</i> . . . . .	8
1.2 <i>Objectives</i> . . . . .	8
1.3 <i>Thesis Organization</i> . . . . .	9
<b>2</b> <i>Background</i>	10
2.1 <i>Introduction</i> . . . . .	10
2.2 <i>Literature Review</i> . . . . .	10
2.3 <i>Comparative Studies</i> . . . . .	16
2.4 <i>Scope of the Problem</i> . . . . .	17
2.5 <i>Challenges</i> . . . . .	17
<b>3</b> <i>Requirement Specification</i>	18
3.1 <i>Business Process Modeling</i> . . . . .	18
3.2 <i>Requirement Collection and Analysis</i> . . . . .	20
3.3 <i>Use Case Modeling and Description</i> . . . . .	23

3.4	Logical Data Model . . . . .	30
<b>4</b>	<b>Design Specification</b>	<b>32</b>
4.1	Front-end Design . . . . .	32
4.2	Back-end Design . . . . .	34
4.3	Implementation Requirements . . . . .	38
<b>5</b>	<b>Implementation and Testing</b>	<b>40</b>
5.1	Implementation of Database . . . . .	40
5.2	Implementation of Interaction & UX . . . . .	42
5.3	Testing Implementation . . . . .	55
<b>6</b>	<b>Conclusion and Future Scope</b>	<b>57</b>
6.1	Discussion . . . . .	57
6.2	Conclusion . . . . .	57
6.3	Further Developments . . . . .	58
	<b>References</b>	<b>58</b>

# List of Figures

3.1	Business Process Modelling - 1	19
3.2	Business Process Modelling - 2	20
3.3	Use Case Modeling	23
3.4	Logical Database Model	30
4.1	The Base64 indexes	35
4.2	Example of Base64 encoding	36
5.1	Database Schema Diagram	41
5.2	Implementation of Database	41
5.3	Encryption of File	42
5.4	User Login	42
5.5	About our software	43
5.6	Our features	43
5.7	Registration	44
5.8	Registration2	44
5.9	Dropdown list	45
5.10	Meeting form	46
5.11	Meeting time confliction	47
5.12	Meeting Rescheduling form	48
5.13	System provide time	48
5.14	Upcoming meeting	49
5.15	Recent Meeting	49

5.16	Called Meetings . . . . .	50
5.17	Post Meeting Form . . . . .	50
5.18	Comment area for a particular meeting . . . . .	51
5.19	Meeting Edit Form . . . . .	51
5.20	Files for a meeting . . . . .	52
5.21	Files of that meeting with upload option . . . . .	52
5.22	Files of that meeting without upload option . . . . .	53
5.23	All Meetings . . . . .	53
5.24	Notices . . . . .	54
5.25	User Profile . . . . .	54

# List of Tables

3.1	Use case:Registration . . . . .	24
3.2	Use case:Call a meeting . . . . .	24
3.3	Use case:Update information . . . . .	25
3.4	Use case:See notices . . . . .	25
3.5	Use Case:Login . . . . .	25
3.6	Use Case:Provide type of meeting . . . . .	26
3.7	Use Case:Select Person . . . . .	26
3.8	Use Case-upload agenda . . . . .	27
3.9	Use Case-upload resolution . . . . .	27
3.10	Use Case-upload Discussion . . . . .	28
3.11	Use Case-upload Follow Up . . . . .	28
3.12	Use Case-cancel meeting . . . . .	29
3.13	Use Case-Control Database . . . . .	29
3.14	Use Case-Log out . . . . .	30

# List of Algorithms

1	Base64 Working Procedure . . . . .	35
2	Detect time Conflict among attendees . . . . .	36
3	Calculate free time . . . . .	37

# CHAPTER 1

## INTRODUCTION

### 1.1 Introduction

In today's world, every organization requires number of interactions among the employees. So meeting is considered as one of the common tasks. However, proper scheduling of these meetings helps in the completion of the tasks and activities on time. But this scheduling of a meeting within different or similar group of people is a time consuming and a critical task. Different tools, techniques and technologies have been developed for the meeting scheduling system [16]. But even after that, people are facing problem to spend less time to schedule a meeting. Most of the time, there arise conflict of time among the attendees. Moreover, in an organization, several meetings are held daily. Those meetings may have same or different agenda. It may happen that while holding a meeting of an old agenda, previous documents of the meeting may be missing or messed up. All the documents may be accessible to all of employee. Agenda of the meeting needs to be in the hand of the attendees before attending the meeting. To provide agenda to each of the attendee is also a time consuming task. Besides, after meeting there are different kind of documents like resolution, follow-up, discussion etc. which need to be preserved securely so that it can be available whenever it is needed.

To solve these problems, we intend to develop a software named **Automated Meeting Scheduling and Document Management System** which will send invitation to the attendees and automatically fix meeting schedule. At the same time, it will maintain and preserve meeting related all the documents securely.

### 1.2 Objectives

Scheduling is considered one of the old problems in the field of the computer science. In response, number of unified solutions have been developed. The multi agent based applications are also facing lot of issues in terms of the scheduling. If the agent based application are not scheduled, the issue may raise and results in the overall failure of the application or proposed solution. A large number of the research works and implementations of different algorithms have been designed, developed and simulated for solving the critical problems

linked with the scheduling. Our main objective is to build a software to automatically schedule a meeting.

**Firstly**, some information will be provided by the meeting caller. And with the information attendees for the meeting will be invited. A notice will be send to them by the system. The attendees can also see the agenda of the meeting in the software. No other person than the attendees can have access to the information.

**Secondly**, if there arises any conflict of time for the meeting (i.e. most of the attendees cant attend the meeting in the given time), then our system will automatically generate a suitable time for the meeting.

**Finally**, the software will store documents of each meeting separately in an organized and safe way with proper encryption and decryption method. It will keep track of sequence of meeting and documents.

### 1.3 Thesis Organization

In chapter 2, the related works regarding meeting scheduling and document management is described. We also shown a comparison between our study and related works. The new innovation and the challenges we faced to develop the software is broadly mentioned.

In chapter 3, the discussion regarding specific requirement has been shown. The business process modeling, requirement collection, use case modeling, logical data model and design requirements are described with figures and tables.

In chapter 4, we have described about our front-end and back-end design. The use manual of the software is given in this section with screen-shots of our software.

In chapter 5, all the technical part of our software is described here. The tools and languages that we use to develop our software is shown.

In chapter 6, we concluded our thesis work with discussion. Our future plan for this software is also illustrated here.

# CHAPTER 2

## BACKGROUND

### 2.1 Introduction

All organization from large scale to small scale hold meetings daily. There are several steps of planning a meeting i.e. selecting appropriate date and time, selecting personnel who will be attending the meeting etc. There are many related works to simplify manual meeting scheduling system to an improved and efficient process or system where less burden were put on human. Each new process can lead to more new ideas to make the system more improved and efficient.

### 2.2 Literature Review

There are many of software tools which are being used in many organization for the scheduling of meetings. Along with the software tools, there are different methods and process to optimize the time slots for meeting scheduling. Advanced algorithms are used to keep privacy of users and importance of meetings and schedule of the meetings, to notify users of upcoming meetings along with agenda, time slots, related papers etc.

Rajiv et.al. introduced Distributed Multi-Event Scheduling (DiMES) framework including the design of congruent DCOP formulations [1]. Using the formalism emerged from DCOP, agents in the large scale applications can optimize time of users to schedule multiple meetings. Here agents control some variable to optimize a global objective function. These objective functions are aggregation of utility function constrained by subset of values. In meeting scheduling using DCOP, the time of employees are managed efficiently along with maintenance of the privacy of the information about importance of meeting. Here two problems were evolved. First one was - there was no automatic mapping process to map specific problems to DCOP. That's why, whole process was to be done i.e. modelling new environment, choosing variables and designing constraining utility functions. Second one - there is no clarity if complete algorithms for DCOP from concrete problems with NP complexity are fast. DiMES (Distributed Multi-Event Scheduling) framework solves many real-world problems where multiple agents generate a coordinated schedule to execute the joint activities

and resource usage in service of multiple events.

Nazia S et.al. developed a scheduling system for people who can schedule their appointment with the doctor by accessing the website of the doctor or hospital [2]. In addition, patients can include more information about current situation. This helps doctor to prepare necessary information before patients arrive. This work represents features like- scheduling daily appointment list of patients, viewing patients complete information, rescheduling the appointment, assigning time slot to each patient, canceling the appointment, checking doctors availability for patients appointment, sending remainder SMS to Patients. The system represented in this work is web based. There is no need to setup the system in advance. Patients can access the website and register if he is first to use the system. Then after being registered users, patients can schedule appointment with the doctors. In appointment scheduling, patients are given an appointment number. Therefore patients dont need to waste time in queue in hospital. This work demonstrate the Browse/Server Model consisting of two sets of functions. The first set of functions are online registration functions including register and login, selection of department, date, doctor and other registration booking functions. The second set of functions are data management functions allowing the database administrator has to add, delete, modify, inquiry, restore and back up. Add, delete, modify and inquiry are the basic operations of database management. They can effectively maintain the consistency of the database to meet the actual need. The data backup and restore can enhance security of the system. Even if the data loss occurs the system can be restored easily.

Kenny Qili Zhu et.al. represented a scheduling system based on different approach [3]. In this work, a meeting scheduling system is represented based on Open Constraint Programming (OCP) paradigm. OCP extends CLP with reactivity and openness. OCP is a concurrent programming framework which allows special queries that react to internal and external events, hence the reactivity. The queries can be submitted by concurrently running, distributed client programs that can be written in any language, hence the openness. These queries are written using a simple language with synchronization and timeout capabilities. This system stands out from the rest due to this declarative programmability of scheduling constraints. Here User-denied constraints and their preferences can be added, removed or changed dynamically. For example, as a user adds more meetings to attend to his calendar, the addition will be automatically translated into changed preferences. Each such change will trigger a constraint solving procedure. The system is knowledge based. Therefore to use the system for any organization or institution some terminals must be defined in the system based on where it is going to be implemented. These terminals are Users, User preferences, Public resources, Public Meetings, Bundles, Optimization Rules and Integrity constraints. The system consist of a constraint store in the form of a constraint logic program, a CLP solver, an OCP server, and a web-based CGI user interface. The store contains global and user data, integrity and user constraints and other rules that govern the behavior

of the system. The users may send a query by clicking buttons on the user interface. These are translated into OCP reactors and sent to the server. Some of the reactors can be delayed in the server and triggered later when some condition is satisfied.

Peter George et. al. invented a process where if there is any request of meeting or event which conflicts another meeting or events scheduled earlier, then one or more variable is optimized. Then selected event is scheduled on the optimized variable [4]. This can be used to scheduling events where most of the personnel are going to attend a meeting or event to eliminate the difficulty of scheduling an acceptable time and location for a majority of people to conduct a meeting or event. Here different variables represents whether a person can travel and reach at the specified location at the predetermined time. Here variables are travel schedules, working periods, convenience of meeting locations etc.

Donna Sue Shaw described technologies in et. al. for collaborative management of activities which occur during the lifecycle of a meeting [5]. Here collaborative management of activities are included in collaborative program. These activities occurs during the entire lifecycle of a meeting. The collaboration program helps meeting participants to collaborate on the creation of a meeting agenda in advance to the occurrence of a meeting. Another benefit is to able to take and exchange meeting notes during a meeting independently. Meeting participants can also collaboratively generate a meeting summary after the meeting.

Pradhan et. al. represented a system and method for dynamic meeting agenda [6]. Here event firing progress indicators is used. An agenda is dynamically managed which corresponds to a meeting by meeting holders using a computing device. The holder selects agenda items as per discussion with other members. The order of agenda items and the computing device can be rearranged by meeting holder and corresponding action items are adjusted accordingly. Meeting participants are notified of timing of meeting. Subscribers are also notified allowable timing to join a meeting at a particular entry point.

The work of Sandip et.al emphasizes on designing and implementing software systems that automate and share information processing tasks of corresponding human users [7]. According to the study, we observed that the benefit of such software was two-fold: First of all, we saw that they allow users to concentrate on more productive tasks, and secondly they improve the quality of information processing by preventing errors that might be introduced by human users.

In this proposed system, authors tried to develop a meeting scheduling, where each employee in the organization would be provided with an automated (computational) meeting scheduling agent. When a user would want to schedule a meeting with other users, he/she needed to input a meeting request to the associated meeting scheduling agent. This agent would negotiate with the agents corresponding to the other users to schedule the meeting. According to the writers, since all meeting requests and calendar accesses would be routed

through the meeting scheduling agent, it could protect the privacy of its associated user while following other preferences of individual. The meeting scheduling agent used the calendar manager software to manipulate the user's calendar, and used the e-mail system to communicate messages with other meeting scheduling agents.

Their System Requirements are to build an intelligent information agent which would be a device independent system. They tried to approximate the privacy and security concern of the user in order to make a distributed meeting scheduling system and tried to provide autonomous scheduling capabilities through restricted information exchange between intelligent meeting scheduling agents. After our study we saw that the approach was considerably more difficult because of the significantly limited information available to scheduling agents about the state of the calendar of other users. The researchers said that when using linear early search bias, agents proposed the earliest available time slot for a requested meeting. When using linear least dense bias, agents searched the entire calendar and then proposed a time interval with maximum free time around it. When using hierarchical search bias, agents scheduling a meeting scheduling, first identified which would be the most desirable week to schedule the meeting, followed by the most desirable day within the chosen week, and finally the particular time interval within that day. Whereas the linear early search bias produced front-loaded calendar leaving room for long meetings later on, it could not accommodate the scheduling of meetings at short notice (and hence triggers meeting cancellations). Linear least dense and hierarchical search biases produced evenly loaded calendars that can more readily accommodate schedule disruptions, but can fail to schedule long meetings when the calendar is dense. They also faced some challenges at the time of information exchanging.

The present implementation was being used by a small group of researchers in their department, and they were using the feedback to improve the interface and expand the functionality of the system. The implemented system, however, needed to be more testing before it could be used in large work groups. The techniques developed so far would work well for meetings with up to approximately 10 users. They would try to schedule meetings with larger number of attendees in future.

The cancellation schemes that they had developed needed to be augmented with mechanisms to utilize user preferences to handle these meetings with a very large number of attendees. In future, they were going to concentrate on developing structured knowledge representation and reasoning mechanism that would allow users to easily specify requirements, and allows the automated negotiator to efficiently use those requirements while scheduling meetings. They used Distributed Meeting Scheduling (DMA) which uses heuristic negotiation strategy for scheduling a meeting. They had also implemented an algorithm for their purpose.

The work of Li et. al. described the design and development of a Web-based scheduling information system [8]. This particular system was intended to allow workers to enter the hours they would like to request to work so that the scheduler can query the database to

see who was available to work on which days in order to make the schedule. The workers are able to query the database as well, so that they can see how many other people have requested the days they plan to request for.

Thomas Herlea et. al. emphasized on different techniques of securely scheduling a meeting [9]. For scheduling a meeting security is must. Mainly in this paper, a negotiation is done among the invitees for the suitable schedule to hold a meeting in a secured manner. A discussion is made between custom-made protocol and protocol based on securely distributed computing for secure meeting scheduling. These protocols are compared on the basis of security properties and complexity. Then the usability of mobile agents are discussed for secure meeting scheduling. The author also presents a prototype called agenTa implementation.

In the custom made negotiation protocol, data representation is done in binary format. The representation minimizes the difficulty of taking a decision about whether the meeting can be scheduled by doing a logical AND operation at a certain moment. The model represented here starts with an invitation phase. The negotiation is done in a protected form of the scheduled. This protocol analyses security of the system by finding bad slots, entropy attack, number of parties and rogue negotiators. The complexity is measured by counting the messages that are sent to the negotiators.

Next protocol for secure distributed computing is described. It is performed on hidden data without revealing the data that means sharing secretly, using encryption etc. It is done by XOR operation. The storage and computation needed for the SDC protocol is higher than the custom-made protocol.

To reduce the communication overhead of the above two protocols, in the next section of the paper the use of mobile agents is shown. The author has described three options for implementing secure meeting scheduling. The first one is trusted third party. To improve the 1st technique cryptographic secure meeting scheduling is used. Finally to combine both of them, mobile agents are involved. Here no longer is a single trusted party needed and high bandwidth communication between the negotiators is also not necessary. At the end the writers show the current implementation with aglets. They introduced a prototype called agenTa for a secure meeting scheduling which used the custom-made protocol.

The Wu et. al. designed and implemented a file encryption and decryption system using USBKey and hardware code [10]. They developed the system and compared it with traditional encryption file system. According to the authors, their system stored the main key of the system as USBKey. It provided authentication by giving login access only to the users who have legal USBKey and entered a correct PIN code. As far as the paper, the files were stored in cipher text and only the authenticated users can decrypt the file by providing USBKey and PIN and view them.

Alexander et. al. claimed that their invention is related to computerized scheduling and management of meetings [11]. It generally explained using a computer software application to automatically schedule and manage agendas for presentation style meetings. The authors described the meeting types in which selected participants are required to attend only portion(s) of the same meeting which are referred as presentation-style meetings; including those in which a number of different presentations are given by more than one individual or group. Presenters are often required to attend a presentation-style meeting only during their presentation times, while audience members are often required to attend the entire meeting. They used intelligent software agents to automate the scheduling of presentation-style meetings, by providing a scheduling agent that generates one or more possible candidate agendas for the meeting based upon user-specified criteria or constraints. The scheduling agent then sends the candidate agendas to the meeting participants, with a request for each participant to respond with a list of preferences selected from the candidate agendas. The scheduling agent then analyzes the responses of the meeting participants using standard optimization techniques to determine one or more optimal agendas for ordering the timing and sequence of presentations given in the meeting. A calendaring agent then automatically updates the calendars of each of the meeting participants with the optimal agendas, or it optionally sends a confirmation request seeking an acceptance of the meeting invitation from a participant before updating his or her calendar with the agendas.

Steve Krasnick and Marc Schecter Online meeting planning program [12] provides the necessary tools for a person coordinating the meeting (hereafter,coordinator) to develop and produce a successful event, including meetings for groups of fewer than one hundred persons. Typically, the meeting planning program functions in a communications environment such as the Internet, wherein one or more persons desiring meeting planning services (hereafter, clients) access the meeting planning site via a Web server on the World Wide Web. The site architecture associated with the meeting planning provides globally accessible, easily navigable forum for planning, and permits private label, co-branding, and personalization for all services. Further, the site content of the meeting planning program adheres to standardized style guidelines and considered writing approach to assure consistent tone, terminology, branding, and spelling there through.

Raymond Lee in Method, system and program product for interactive electronic meeting scheduling [13]represents a calendaring system Wherein a meeting requester interacts With online invitees and calendars of off-line invitees. The meeting scheduler is presented a GUI interface to the calendaring system. The scheduler provides various criteria for the meeting including an invitee list and attributes of the invitee such as whether he is required for the meeting or a quorum of a minimum number of invitees to successfully schedule the meeting. The scheduler also provides criteria such as a date or date range, a time or time range and the

like for the meeting to be scheduled in. The calendaring system then negotiates a schedule with invitees who are online interactively. This may be done real time by instant messaging techniques or it may be done in background by e-mail communication. The calendaring system has access to invitee calendars and by using a deadline algorithm, will opt for the invitee calendar when an interactive response is not forthcoming.

Jacky Kuo Jacky Kuo uses a system for managing meetings in Meeting management system [14]. The system has a database with data fields for entering a meeting definition of a future meeting as meeting definition data; a computer memory storing the data for retrieval by the database; and an auto-mailer sending an encrypted and read only meeting definition document and meeting reminders that are generated by the database and sent by the auto-mailer to notify meeting members identified by the data about a future meeting described by the data. The system electronically records meeting information, and uses the meeting information to notify meeting members of meetings. Further, the system uses the meeting information to send reminders of the meetings via a security controlled electronic mailing system. The reminders are automatically generated to improve meeting communication and to improve readiness for the meetings. At a Workplace, meetings are held among meeting members to contribute to a project or to solve problems. Meeting minutes are recorded as part of updated meeting information, and is sent to meeting members via an electronic mailing system.

Eric J. Horvitz in Time management representation and automation for allocating time to projects and meetings within an online calendaring System [15] includes providing several services, including automated scheduling of project focus meetings With the owner of the system, reminders. It can also track the time and resources used for the tasks, rescheduling of meetings given user input about putting off the task and juggling and rescheduling appointments with other appointments as they come in, given the priority of the appointment.

## 2.3 Comparative Studies

The features of related works are similar to features of our thesis. In the work of Rajiv et.al. about DiMES (Distributed Multi-Event Scheduling) and congruent DCOP(Distributed Constraint Optimization) formulations [1], multi-agent applications are optimized in joint performance. Each user has personal assistant agents which optimizes users time. User have privacy about their meeting. In our work, though there is no personal assistant agents to optimize users time, but users privacy of meeting is maintained.

The work of Nazia S et. al. [2] is more similar with work of our project. In both work, appointment is fixed in online, prior discussion before appointment/meeting is allowable. In the work of Peter George et. al. [4], if there is any conflict of time of meeting, then system generates a time which is acceptable by majority of the participants where as in our work,

there will be an option if the meeting planner wants all of the possible participants in the meeting or not. If he wants all participants in the meeting, then system will suggest a time when everyone has no meeting. Though in his work, many factors like travel schedules, meeting location etc were considered, in our work the system is built for all members of same organization.

The Wu et. al. worked on file encryption and decryption system but they havent worked on safe transmission of the files [10]. Whereas, we used base64 algorithm in our software for secure transmission of the documents.

## 2.4 Scope of the Problem

After all the studies of the papers and other related works, we observed that meeting scheduling is an issue in recent time and there is a huge scope of working on it. So, we decided to build a software to maintain the smooth conduction of meeting. It will automatically schedule meeting and preserve all the documents of the meeting in a secured way so that it can save the time wasted to do these works.

## 2.5 Challenges

Major challenge of our project was to work on use case diagram, logical database design and implementing the use case diagram, logical database design. Use case diagrams are used to describe a set of actions or use cases that a system can perform with external users of the system. In our project each use case provide observable result to the stakeholders of the system. The implementation of use case diagram is useful for designing the system.

Logical design of the database is more conceptual and abstract. It is required to design the data ware house where all information of personnel of the organization, meeting and all types of documents will be stored and this falls in our scope of our system design. In logical database design, specific data contents are defined along with the relationships between groups of data. The design is oriented toward the needs of the end users.

Another major challenge is the implementation of automatic meeting time generation by system. While planning a meeting, meeting planner may not know others scheduling. Therefore, if there is any conflict in meeting time, then by maintaining the privacy of the user, an automation process is used which will give a preferable time for meeting after checking other scheduling.

Linking the database and HTML forms was certainly a challenge. The biggest challenge, however, was coding and debugging in PHP.

# CHAPTER 3

## REQUIREMENT SPECIFICATION

### 3.1 Business Process Modeling

Business Process Model is the activity of representing processes of an organization or an enterprise to analysis and improve. The Business Process Model of our project is here followed by the depiction of the model in fig3.1-

Main Event of our project is the planning or calling a meeting. Processes are desctried below as per order-

1. **Call a meeting:** This process starts when user needs to call a meeting.
2. **Upload Agenda:** Every meeting has agenda. In this process, agenda must be written beforehand planning the meeting and then upload it during the process.
3. **Fill the meeting information:** To call a meeting, the meeting planner needs to fill necessary information of the meeting like time, date, venue etc.
4. **Selects members of the meeting:** : In this process, all the members of the meeting are selected.
5. **Decide if there is any conflict of time among members of the meeting:** Members of the meeting may have meeting with others on the same day at same time. This process will give output if there is any conflict of the meeting.
6. **Decide if attendance of all the members are required or not:** : In this process, it is asked from the user that if it is required for all members to present in the meeting or not. Based on the decision of the user exclusive process works.
7. **Decide time when all will be free:** : In this process a time is determined by the system when all the members dont have any meeting by maintaining privacy.
8. **Send invitation to all members:** After determining the time, meeting invitation is sent to all of the member.

9. **Send invitation of members:** This is the exclusive part of the processes where meeting invitation is sent to only those users who don't have other meeting on same day at same time.

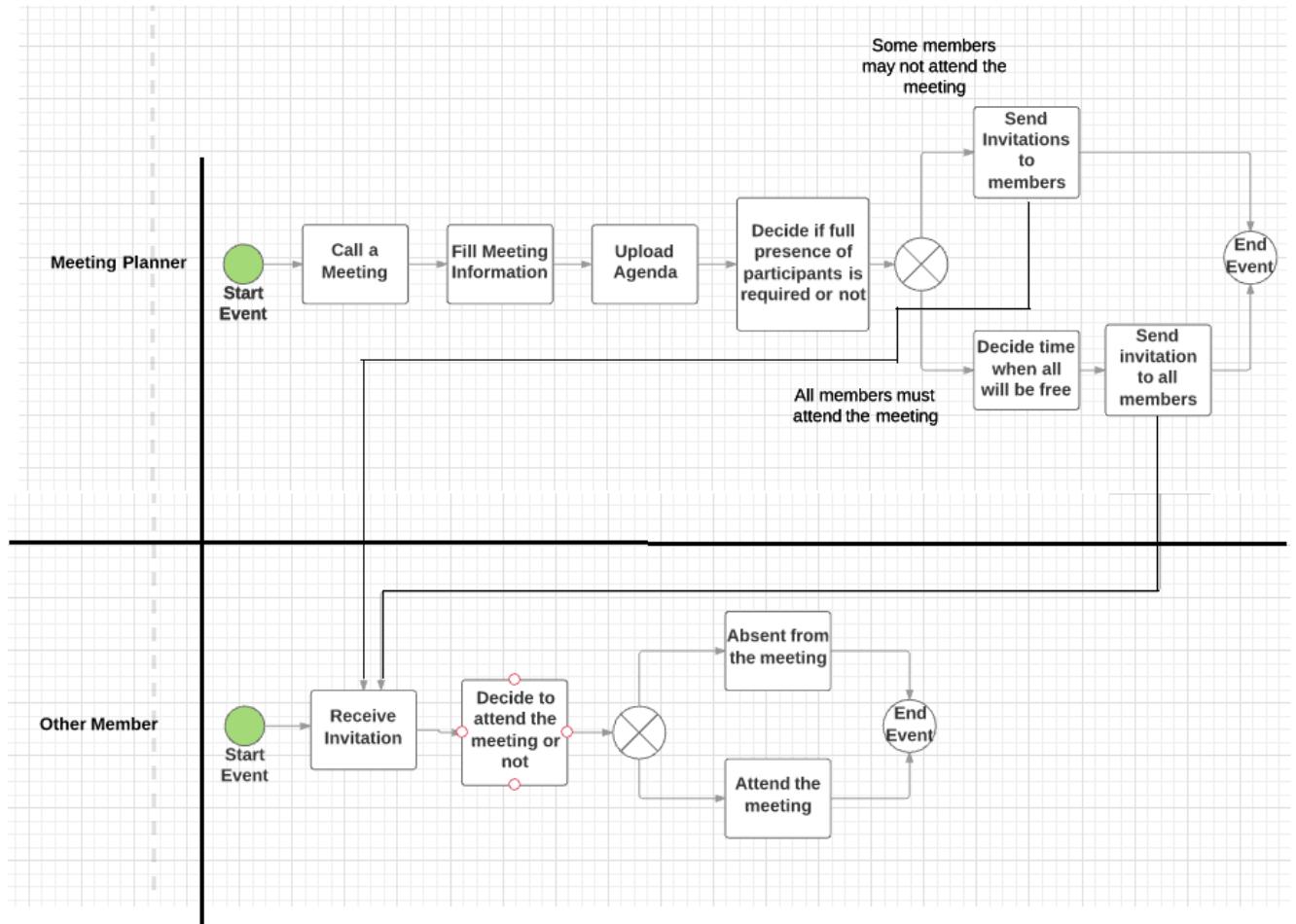


Figure 3.1: Business Process Modelling - 1

Processes in the event of receiving invitation is described below and depicted in fig:3.2-

- Receive Invitation:** When other members log in their account, using this process the notice of meeting invitation is displayed.
- Decide to attend the meeting or not:** User can either attend the meeting or absent from the meeting by pressing buttons which takes either of the processes.
- Absent from the meeting:** In this process, name of the user will not be in the list of attendees of the meeting and then the event is closed.
- Attend the meeting:** In this process, name of the user will be in the list of the attendees of the meeting and get all information regarding the meeting and then meeting is closed.

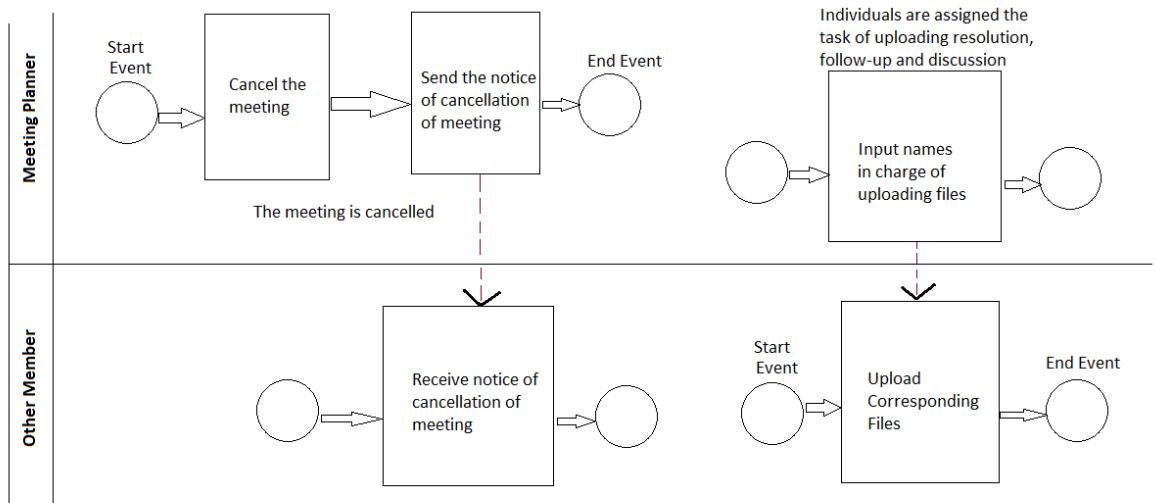


Figure 3.2: Business Process Modelling - 2

### 3.2 Requirement Collection and Analysis

A requirement is some quality or performance demanded of a person in accordance with certain fixed regulations. Requirement Collection and Analysis means process of finding out, analyzing, documenting, and checking the services that a customer requires from a system and the constraints under which it operates.

**Requirements may be classified into two class:**

1. User Requirement

2. System Requirement

#### 1. User Requirement:

Its the abstract requirements.

- (a) Our software have to be user friendly so that user can't face any difficulty in using it.
- (b) Our software have to be device independent for easy accessibility.
- (c) Security system should be high and information should be kept private.
- (d) Should have easy access to call a meeting and giving opinion about his/her attending that meeting.
- (e) Should have the facility to show all his/her meeting details and documents

#### 2. System Requirement:

It includes detailed description of what the system should do. It defines exactly what

is to be implemented. It may be part of contract between the system buyer and the software developers.

- (a) Software should be device independent. It would have its own domain address. And anyone willing to use the application would have to use that specific domain address. Any device whether it is running WINDOWS, LINUX, or MAC OS should be able to use the software if it can have an html viewable browser.
- (b) Platform would provide an individual profile for everyone.
- (c) System should have the facility that every profile have contact fluency.
- (d) Have to give the particular privacy for particular user. As our software has four types of user. System have to differentiate them through their privacy system.
- (e) All the documents of files preserved in database have to be secured in encrypted form.

### **Requirements can be classified based on functional and non-functional requirement:**

- 1. Functional Requirement
- 2. Non-functional Requirement

#### **1. Functional Requirement:**

It describes functionality or system services and depends on the type of software, expected users and the type of organization where the software is used. In general, functional user requirements may be high-level statements of what the system should do.

- (a) **Request From user:** Request from user will go to the server to access their account
- (b) **Acceptance or rejection :** Server will accept or reject the user's request according to password matching with the given password.
- (c) **Calling a meeting:** User will have easy access to call or cancel a meeting.
- (d) **Show all files:** User should have the access to show all the files what he has attended
- (e) **Check notices:** User have to check the notices for attending the meeting and uploading the files.
- (f) **Comments:** User should have the access to comment for a particular meeting.
- (g) **Secure :** The access of a user's profile information should be only that user and he/she will have to be able to update their information

## 2. Non-functional Requirement:

It defines the system properties and constraints e.g. reliability, response time and storage requirements. Constraints are I/O device capability, system representations, etc. It is more critical than functional requirements. If these are not met, the system may be useless.

- (a) **Security:** For security of the document of meetings, documents should be kept encrypted in database. Many algorithm like base64 can be applied for it.
- (b) **Accuracy:** The features and procedures that planning to use have to be more accurate so that system which will help the user a lot.
- (c) **Quick performance :** This system saves enough time for a user which will need to schedule a meeting. This software can automatically gives the schedule if anyone planning to go for a meeting.
- (d) **User usability and friendly:** Any user find it very easy to use and it will be very friendly for him.

### 3.3 Use Case Modeling and Description

In software and systems engineering, a use case is a list of actions or event steps typically defining the interactions between a role (known in the Unified Modeling Language as an actor) and a system to achieve a goal. The actor can be a human or other external system. Use case of our system can be described as below:

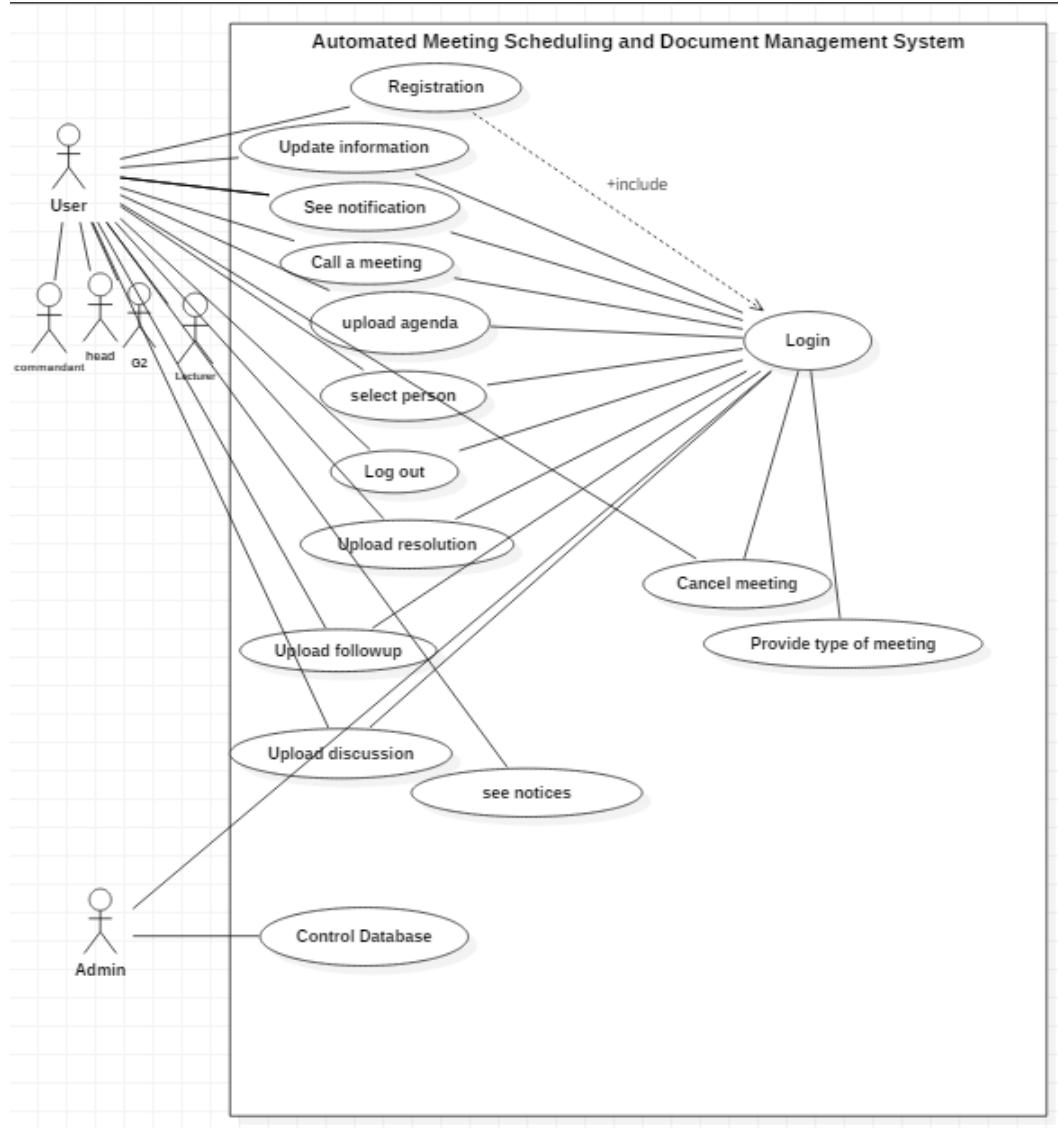


Figure 3.3: Use Case Modeling

## **Usecase Description: Register:**

<b>Usecase name</b>	Register
<b>Primary Actor</b>	Acting member, Admin block
<b>Goal in Context</b>	To register in the system to use the system to conduct meetings properly
<b>Precondition</b>	None
<b>Trigger</b>	To use the system, the members of the organization will have to register
<b>Primary Path</b>	<ol style="list-style-type: none"> <li>1. User clicks on the button 'Register'</li> <li>2. Clicks on the button 'Register'.</li> <li>3. Provides necessary information to register i.e. name, username, email, password etc.</li> <li>4. Clicks on 'Submit' button.</li> </ol>
<b>Alternative Path</b>	<ol style="list-style-type: none"> <li>3.1 If there is already an existing profile of the user, then the system will notify.</li> <li>3.2 If key attribute of the given information matches with others' information in the database, then system will notify.</li> </ol>

Table 3.1: Use case:Registration

## **Call a Meeting:**

<b>Usecase name</b>	Call a meeting
<b>Primary Actor</b>	Acting member
<b>Goal in Context</b>	To call a meeting among members of the institution
<b>Precondition</b>	User must be registered
<b>Trigger</b>	There must be an issue to discuss
<b>Primary Path</b>	<ol style="list-style-type: none"> <li>1. User clicks on the link called 'Call a meeting'.</li> <li>2. After Login, user enters into his profile to select necessary information to finalize everything about upcoming meeting.</li> </ol>
<b>Alternative Path</b>	<ol style="list-style-type: none"> <li>2.1 If user is not logged in, then the system will notify him.</li> </ol>

Table 3.2: Use case:Call a meeting

**Update information:**

<b>Usecase name</b>	Update information
<b>Primary Actor</b>	Acting member
<b>Goal in Context</b>	To update or change information in profile
<b>Precondition</b>	User must be registered
<b>Trigger</b>	With the change of any real information, user needs to update profile in the system
<b>Primary Path</b>	<ol style="list-style-type: none"> <li>1. After logging in, User clicks on the 'Update info' button.'</li> <li>2. Enters into own profile.</li> <li>3. Edit necessary information.</li> <li>4. Save change by clicking 'Save' button.</li> </ol>
<b>Alternative Path</b>	3.1 User cant make his username which is already available.

Table 3.3: Use case:Update information

**See Notices:**

<b>Usecase name</b>	See Notices
<b>Primary Actor</b>	Acting member
<b>Goal in Context</b>	To check notices of meetings and system updates.
<b>Precondition</b>	User must be registered
<b>Trigger</b>	If there is any recent notifications, then user will see on it.
<b>Primary Path</b>	<ol style="list-style-type: none"> <li>1. User logs in.'</li> <li>2. Go to his my account page.</li> </ol> <p>See recent notices.</p>
<b>Alternative Path</b>	

Table 3.4: Use case:See notices

**Login:**

<b>Usecase name</b>	Login
<b>Primary Actor</b>	Acting member,Admin
<b>Goal in Context</b>	To be enabled to have access to use the system..
<b>Precondition</b>	A member must be registered in the system
<b>Trigger</b>	User log in to the system to use.
<b>Primary Path</b>	<ol style="list-style-type: none"> <li>1. User clicks on 'Log In' button on screen.</li> <li>2. User types the username and password.</li> <li>3. With correct username and password, user will be logged in successfully.</li> <li>4. User will have access to the system.</li> <li>5. Likewise, Admin will have the control over system.</li> </ol>
<b>Alternative Path</b>	2.1 User may type wrong user-id and password. Then user will be able to try again to log in.

Table 3.5: Use Case:Login

**Provide type of meeting:**

<b>Use case name</b>	Provide type of meeting
<b>Primary Actor</b>	Acting member
<b>Goal in Context</b>	To hold a specific type of meeting
<b>Precondition</b>	Meeting planner must be logged into the system
<b>Trigger</b>	Meeting planner needs to specify the type of the meeting.
<b>Primary Path</b>	<ol style="list-style-type: none"> <li>1. The meeting planner i.e. acting member selects the 'Call a Meeting' option from his profile.</li> <li>2. Several options of different types of meeting (departmental, inter-departmental, inter-university) will pop up on screen.</li> <li>3. From these options, the caller will select preferable one.</li> </ol>
<b>Alternative Path</b>	2.1 If any option doesn't matches with the purpose, user may choose other and give the name of the meeting.

Table 3.6: Use Case:Provide type of meeting

**Select Person:**

<b>Use case name</b>	Select Person
<b>Primary Actor</b>	Acting member
<b>Goal in Context</b>	To select specific persons to perform different task
<b>Precondition</b>	The meeting conveyor should be logged in and select a type of the meeting
<b>Trigger</b>	Some persons or group of persons have been selected to attend the meeting and among them some are assigned to do specific task like uploading agenda, discussion and follow-up.
<b>Primary Path</b>	<ol style="list-style-type: none"> <li>1. The meeting conveyor goes to the post meeting form.</li> <li>2. Then insert the name of a user who will upload the resolution.</li> <li>3. Then insert the name of a user who will upload the discussion.</li> <li>4. Then insert the name of a user who will upload the followup.</li> </ol>
<b>Alternative Path</b>	

Table 3.7: Use Case:Select Person

**Upload Agenda:**

<b>Usecase name</b>	Upload Agenda
<b>Primary Actor</b>	Selected Person
<b>Goal in Context</b>	Selected person uploads the agenda of the meeting so that all the invited members can read it
<b>Precondition</b>	The selected person should be assigned the responsibility of uploading the agenda by the caller
<b>Trigger</b>	The agenda of the meeting called has been uploaded and the attendees can see it
<b>Primary Path</b>	<ol style="list-style-type: none"> <li>1. The selected person clicks on 'Upload Agenda' option.</li> <li>2. He/she browses the agenda from the PC and upload it.</li> <li>3. The agenda is shared with the attendees in the system.</li> </ol>
<b>Alternative Path</b>	

Table 3.8: Use Case-upload agenda

**Upload Resolution:**

<b>Usecase name</b>	Upload Resolution
<b>Primary Actor</b>	Selected Person
<b>Goal in Context</b>	To upload the resolution after meeting.
<b>Precondition</b>	Resolutions are stored which are taken through the meeting
<b>Trigger</b>	To store the resolutions of the meeting, uploading meeting will help to store and keep track of the meeting.
<b>Primary Path</b>	<ol style="list-style-type: none"> <li>1. The selected person collects the resolution from the host who call the meeting .</li> <li>2. Then he login to our system.</li> <li>3. After logging he enters the specific meeting segment which resolution he wants to upload.</li> <li>4. He selects upload resolution from that segment.</li> <li>5. Brows the file to complete his task.</li> <li>6. After uploading he leaves that segments and do log out.</li> </ol>
<b>Alternative Path</b>	

Table 3.9: Use Case-upload resolution

**Upload Discussion:**

<b>Usecase name</b>	Upload Discussion
<b>Primary Actor</b>	Selected Person
<b>Goal in Context</b>	To upload the discussion when meeting is over.
<b>Precondition</b>	Discussions are stored which are discussed through the meeting
<b>Trigger</b>	Selected Person have attended the meeting.
<b>Primary Path</b>	<ol style="list-style-type: none"> <li>1. The selected person collects the discussion from the meeting. It will be a record if the meeting is online or it will be the file i.e docx or pdf if the meeting is physical.</li> <li>2. After logging he enters the specific meeting segment which discussions he wants to upload.</li> <li>3. He selects upload discussion from that segment.</li> <li>4. He browses the file to complete his task.</li> <li>5. After uploading he leaves that segments and do log out.</li> </ol>
<b>Alternative Path</b>	

Table 3.10: Use Case-upload Discussion

**Upload Follow Up:**

<b>Usecase name</b>	Upload Follow Up
<b>Primary Actor</b>	Selected Person
<b>Goal in Context</b>	To upload the follow up to track the work of any department or university.
<b>Precondition</b>	To make a follow up for a meeting every resolution and results are preserved correctly
<b>Trigger</b>	Selected Person have attended the meeting.
<b>Primary Path</b>	<ol style="list-style-type: none"> <li>1. The selected person makes follow up from the resolution and result of a meeting and do check from the host of the meeting.</li> <li>2. After logging he enters the specific meeting segment which follow up he wants to upload.</li> <li>3. He selects 'Upload Follow Up' from that segment.</li> <li>4. He browses the file to complete his task.</li> <li>5. After uploading this follow up goes to the own database of that department meeting or university meeting.</li> <li>6. Then he leaves that segments and do logout.</li> </ol>
<b>Alternative Path</b>	2.1 The selected person enters wrong segments or upload wrong file.

Table 3.11: Use Case-upload Follow Up

**Cancel Meeting:**

<b>Usecase name</b>	Cancel Meeting
<b>Primary Actor</b>	Acting Member
<b>Goal in Context</b>	To cancel any meeting which was planned by the same user.
<b>Precondition</b>	A member should call that meeting
<b>Trigger</b>	Canceling the meeting.
<b>Primary Path</b>	<ol style="list-style-type: none"> <li>1. The meeting planner clicks on the 'Cancel Meeting' option.</li> <li>2. List of all the meeting pops on screen which was planned by same person.</li> <li>3. Click on the option which meeting user wants to cancel.</li> <li>4. The notifications of cancellation of meeting are sent to all the members who were invited in the meeting.</li> </ol>
<b>Alternative Path</b>	

Table 3.12: Use Case-cancel meeting

**Control Database:**

<b>Usecase name</b>	Control Database
<b>Primary Actor</b>	Admin
<b>Goal in Context</b>	Controls and monitors the user access to the database
<b>Precondition</b>	Admin is logged into the system
<b>Trigger</b>	Canceling the meeting.
<b>Primary Path</b>	<ol style="list-style-type: none"> <li>1. Admin checks whether there is need to upgrade database server or not. If yes, he upgrades the server.</li> <li>2. If need, he creates reports by querying from database.</li> <li>3. He selects upload follow up from that segment.</li> <li>4. Monitors the whole database performance and optimize it.</li> </ol>
<b>Alternative Path</b>	

Table 3.13: Use Case-Control Database

### Log out:

<b>Usecase name</b>	Log out
<b>Primary Actor</b>	Acting Member
<b>Goal in Context</b>	To give no more access to the system to do any other functions.
<b>Precondition</b>	A member should be logged in the system
<b>Trigger</b>	After completion of desired tasks, logged in user will have no other purpose to use the system for that time.
<b>Primary Path</b>	<ol style="list-style-type: none"> <li>1. The faculty member selects Log Out option from his profile.</li> <li>2. He/she no more has access to the system.</li> </ol>
<b>Alternative Path</b>	

Table 3.14: Use Case-Log out

### 3.4 Logical Data Model

A logical data model describes the data in as much detail as possible, without regard to how they will be physical implemented in the database. Features of a logical data model include: Includes all entities and relationships among them. All attributes for each entity are specified. Logical Data Model for our system is given below.

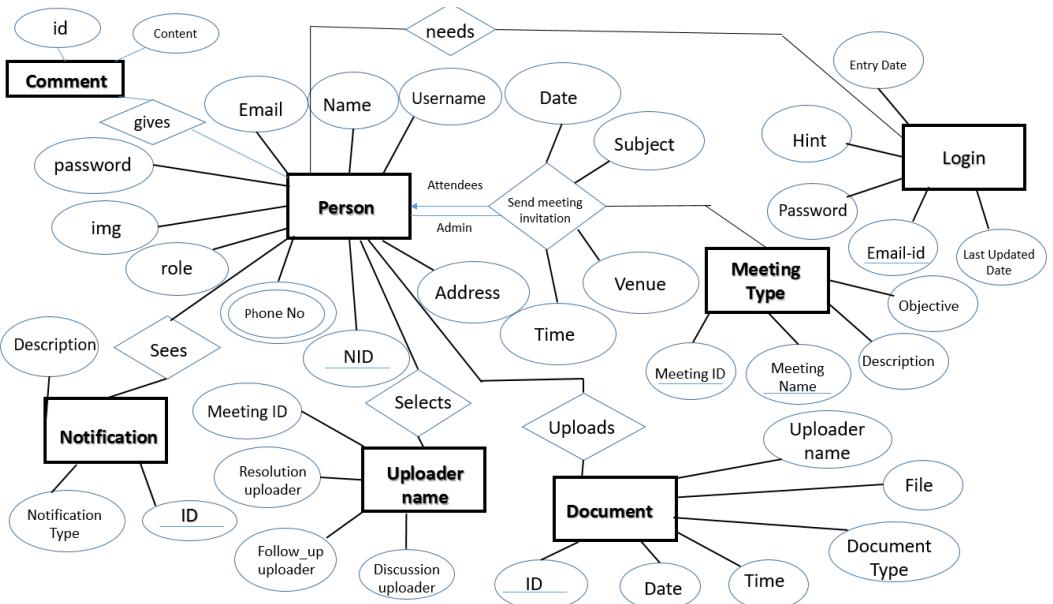


Figure 3.4: Logical Database Model

Here we consider seven table for our database. Each table has its own entity. We will go through one by one.

- **Person:** Its vital table in our system. When one get registered, information of that person will entry in this table and that person became the user of our system. One have to give his/her unique user-name and email address for this table to get registered. One's name, user-name, email, phone no, NID, address, password, img, role are the entity of this table. This table has its primary key and its ID and phone number of that person is multi-valued attribute.
- **Meeting type:** Its another important table for our database. When a user call a meeting, many information will have to fillup to submit the meeting form. When meeting is called information is stored in this table. Primary key of this table is meeting id. Another attributes are meeting date time, conveyor name, meeting name, venue.
- **Document:** Document is the another table of our system. Its primary key is ID. Documents of a particular meeting are uploaded in this table. If any meeting have more than one resolution or discussion file, then all files of resolution or discussion will be uploaded against a same meeting id. Its attributes are meeting id, file type, file name, uploader name, time, date.
- **Comment:** This table for commenting for a particular meeting. Comment for a specific meeting is considerable. For this meeting id, username, date, time is necessary attribute for this table.
- **Notification:** If one has a upcoming meeting or ones any meeting is rescheduled or cancel then he/she will get notice from our system and this table is for that. Its has it primary key id. and another attribute also.
- **Uploader:** This table is necessary when a meeting is over. After closing a meeting, one have to upload the documents of that meeting like resolution, discussion etc. Meeting caller decide this and information is saved this table. It has meeting id, resolution uploader name, discussion uploader name, followup uploader name attributes.
- **Login:** Last entry time of a user, that email, password and hint is the attribute of this table. This table helps the system to track every person

# CHAPTER 4

## DESIGN SPECIFICATION

In the context of software, Design Specification is usually a design document that describes all data, architectural, interface and component-level design for the software. A design specification provides explicit information about the requirements for a product and how the product is to be put together. Its use is called for where a product has to be specially made to meet a unique need. [17]

### 4.1 Front-end Design

To design the front-end of the software both technical and non-technical terms are needed. We used following tools and languages to build the software:

#### Tools:

1. Editor.

##### (a) **Dreamweaver:**

Adobe Dreamweaver is a proprietary web development tool developed by Adobe Systems. We have used it for code Highlighting as With each type of code there is a different set of highlights to help you differentiate between HTML, CSS, PHP, Javascript and the list goes on. It scans any error quickly. And most importantly it can give code suggestion, For this you don't have to write full with your own effort.

##### (b) **Phpstorm:**

PhpStorm is a PHP IDE that actually gets your code. It supports PHP 5.3/5.4/5.5/5.6/7.0/7.1/7.2, provides on-the-fly error prevention, best auto-completion code re-factoring, zero configuration debugging, and an extended HTML, CSS, and JavaScript editor.

##### (c) **Notepad++:**

Its a free source code editor which supports several programming languages running under the MS Windows environment. We have used it because doesnt take up much space on laptop/desktop PC and its low specification software it means that it doesnt take a long time to load.

**Platform:** windows

**Browsers:**

The software can run on any browser. We mainly used **Opera mini** and **Chrome** to run our software.

**Languages:****1. HTML:**

Hypertext Markup Language is the standard markup language for creating web pages and web applications. We use it to give the structure of Web pages using markup.

**2. CSS:**

Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language. We used it to design our pages

**3. CSS3:**

It is also a style sheet language used for describing the presentation of a document written in a markup language. For some design we use it in our software.

**4. JAVASCRIPT:**

JavaScript, often abbreviated as JS, is a high-level, dynamic, weakly typed, prototype-based, multi-paradigm, and interpreted programming language. We have used it to design our software because it makes our software user interface easy.

**5. BOOTSTRAP:**

Bootstrap is a free and open-source front-end web framework for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. We have used it to make our page more responsive.

**AJAX:**

AJAX stands for Asynchronous JavaScript And XML. It is not a programming language. It uses a combination of:

- A browser built-in XMLHttpRequest object (to request data from a web server)
- JavaScript and HTML DOM (to display or use the data)

AJAX allows web pages to be updated asynchronously by exchanging data with a web server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page. [18]

## 4.2 Back-end Design

To design back-end of the software, we needed

- PHP
- MySQL and
- Wamp server

Different algorithms were also run on the back-end algorithm for automation, conflict detection and base64 algorithm for encryption and decryption were used for our software.

### **PHP:**

PHP is an acronym for "PHP: Hypertext Preprocessor". It is a widely-used, open source scripting language. PHP scripts are executed on the server. It is free to download and use. PHP code are executed on the server, and the result is returned to the browser as plain HTML. We used PHP,

- To generate dynamic page content, to create, open, read, write, delete, and close files on the server.
- To collect form data
- To send and receive cookies
- To add, delete, modify data in your database
- To control user-access
- To encrypt data in the database using base64 algorithm

### **MySQL:**

MySQL is the most popular Open Source SQL database management system. MySQL databases are relational. A relational database stores data in separate tables rather than putting all the data in one big storeroom. The SQL part of MySQL stands for Structured Query Language. SQL is the most common standardized language used to access databases.

### **WampServer:**

WampServer is a Web development platform on Windows that allows to create dynamic Web applications with Apache2, PHP, and MySQL. WampServer automatically installs everything needed to intuitively develop Web applications. Server can be tuned without even touching its setting files.

## Algorithms:

- **Base64 Description:**

Base64 encoding schemes are commonly used when there is a need to encode binary data that needs be stored and transferred over media that are designed to deal with textual data. This is to ensure that the data remains intact without modification during transport.

---

### Algorithm 1 Base64 Working Procedure

---

Base64 encoding takes the original binary data and operates on it by dividing it into tokens of three bytes. A byte consists of eight bits, so Base64 takes 24bits in total. These 3 bytes are then converted into four printable characters from the ASCII standard.

The first step is to take the three bytes (24bit) of binary data and split it into four numbers of six bits. Because the ASCII standard defines the use of seven bits, Base64 only uses 6 bits (corresponding to  $2^6 = 64$  characters) to ensure the encoded data is printable and none of the special characters available in ASCII are used. The algorithm's name Base64 comes from the use of these 64 ASCII characters. The ASCII characters used for Base64 are the numbers 0-9, the alphabets 26 lowercase and 26 uppercase characters plus two extra characters '+' and '/'.

The Base64 indexes are showed in fig 4.1: The ASCII conversion of 3-byte, 24-bit

Value	Char	Value	Char	Value	Char	Value	Char
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/

Figure 4.1: The Base64 indexes

groups is repeated until the whole sequence of original data bytes is encoded. To ensure the encoded data can be properly printed and does not exceed any mail server's line length limit, newline characters are inserted to keep line lengths below 76 characters.

If the size of the original data in bytes is not a multiple of three, we might end up

with only one or two remaining (8-bit) bytes. The solution is to add the missing bytes by using a byte value of '0' to create the final 3-byte group. Because these artificial trailing '0's cannot be encoded using the encoding table, we introduce a 65th character: '=' to represent '0'. Naturally, this character can only appear at the end of encoded data [19].

Example of Base64 encoding is given in fig 4.2:

source ASCII (if <128)	M	a	n
source octets	77 (0x4d)	97 (0x61)	110 (0x6e)
Bit pattern	0 1 0 0 1 1 0 1 0 1 1 0		
Index	19	22	5
Base64-encoded	T	W	F
encoded octets	84 (0x54)	87 (0x57)	70 (0x46)
			117 (0x75)

Figure 4.2: Example of Base64 encoding

- **Conflict detection Algorithm:**

---

**Algorithm 2** Detect time Conflict among attendees

---

**Require:** attendees list, meeting start & end time, meeting date

**Ensure:** whether time conflict occurs or not

```

 $a \Leftarrow$  no. of attendees
 $s \Leftarrow$  meeting start time
 $e \Leftarrow$  meeting end time
 $i \Leftarrow 1$ 
 $f \Leftarrow 0$ 
while  $i \leq a$  do
    while start and end time of attending meeting of every  $i$  attendee
    do
         $r[stime] \Leftarrow$  attending meeting start time from db
         $r[etime] \Leftarrow$  attending meeting end time from db
        if  $(s \geq r[stime] \& s \leq r[etime])$  or  $(e \geq r[stime] \& e \leq r[etime])$  then
             $f \Leftarrow 1$ 
            break
         $i \Leftarrow i + 1$ 
    end if
    end while
end while
if  $f = 1$  then
    Conflict
else
    No Conflict
end if

```

---

- **Automation Algorithm:** The algorithm correctly works for meeting of one hour duration. Mainly hour values are considered in our algorithm.

---

**Algorithm 3** Calculate free time

---

**Require:** office start & end time, attendees list, meeting start & end time, meeting date

**Ensure:** whether there is any free time slot or not

```
a  $\Leftarrow$  no. of attendees
j  $\Leftarrow$  office start time
k  $\Leftarrow$  office end time
i  $\Leftarrow$  1
c  $\Leftarrow$  0
s  $\Leftarrow$  meeting start time
e  $\Leftarrow$  meeting end time
d  $\Leftarrow$  e - s
while j  $\leq$  k do
    r[stime]  $\Leftarrow$  attending meeting start time from db
    s[0][i]  $\Leftarrow$  j
    j  $\Leftarrow$  j + d
    i  $\Leftarrow$  i + 1
    c  $\Leftarrow$  c + 1
end while
i  $\Leftarrow$  1
while i  $\leq$  a do
    while start and end time of attending meeting of every i attendee
    do
        r[stime]  $\Leftarrow$  attending meeting start time from db
        r[etime]  $\Leftarrow$  attending meeting end time from db
        n  $\Leftarrow$  1
        while n  $\leq$  c do
            if (s[0][n] = r[stime]) or (s[0][n] = r[etime]) then
                s[i][n]  $\Leftarrow$  1
            end if
            n  $\Leftarrow$  n + 1
        end while
    end while
    end while
identify columns with all zero value
if found then
    Consider it as free time slot
else
    No free time
end if
```

---

### **4.3 Implementation Requirements**

After all the requirement analysis we observed that the software needed following features with necessary security:

**1. Register:**

To use the system, personnel of an organization must register with basic information like name, email-id, user name etc. and set password.

**2. Login:**

Registered user will have to login using username and password to do any actions within the system.

**3. Meeting List:**

After login, user will see the list of his/her upcoming meetings, recently attended meetings, meetings called by him/her and all the meetings together.

**4. Schedule a meeting:**

To schedule meeting one need to choose the Call Meeting option and he/she can see a form. He needs to fill-up the form with all the required information. Here conflict checking will occur. That is if the attendees are busy at the time given for meeting, system will notify the meeting caller and the free time slots of the attendees will also be shown. The caller either can choose any of the free time slot or call the meeting at the previous time.

**5. Upload files of meeting:**

After completion of a meeting the meeting caller will give the name of the persons to upload files like resolution, agenda and follow-up. Only the authorized person can upload and delete a file. Other attendees can only see and download the file. All the files will be kept encrypted in the database. It will be decrypted when the authorized person will download it.

**6. Reschedule a Meeting:**

Only the meeting caller and the personnel with high rank than the caller can reschedule a meeting to another time.

**7. Cancel a Meeting:**

Once a meeting is called only the meeting caller and the personnel with high rank than the caller can cancel the meeting.

**8. See Notice:**

After scheduling, rescheduling or canceling a meeting, attendees of the meeting will get notice of the news.

**9. Comment:**

The attendees of the meeting can discuss about the meeting and give any kind of query regarding the meeting in the comment section.

**10. Update Information:**

Users personal information are stored in the database and can be updated.

**11. Logout:**

After performing necessary functions, user will logout from the system.

# CHAPTER 5

## IMPLEMENTATION AND TESTING

Implementation of software is a structured method where all components are combined which gives service of particular certain features. In this chapter, outline of implementation of the software is narrated properly.

### 5.1 Implementation of Database

The DBMS is installed on the required hardware, optimized the database to run best on that hardware and software platform, and the database is created and the data is loaded in the implementation phase. The phase of implementation are described below-

1. **Installing DBMS:** The name of the DBMS is the MySQL, an open source SQL database management system. This system was installed first for implementing the database.
2. **Creating the database and tables:** The logical database database model in figure - [3.4](#) is used to create Database Schema or Physical Schema which is used to implement the basic part of the implementation of the database.

The Physical Schema diagram is shown below following the description-

- (a) **Meeting:** The columns Meetingid, Meetingname, task, Date, Time, Location, Type, MeetingCaller and MeetingConveyor are defined as auto generated primary key of the table, name of the meeting, task to do for the meeting, date and time, location and type of the meeting respectively. Meetingcaller means the person who creates the invitation of a meeting and MeetingConveyor means the person who conveys the meeting.
- (b) **Person:** : This table holds necessary information of users. 'Username' is the primary key. Rest of the column represents basic information of a user name, NID, phone number, address, role.
- (c) **Uploader:** The primary key is auto generated. Each meeting has different types of documents and users are assigned to upload these documents. This table holds the information that which user will upload which documents.
- (d) **Document:** This table stores the files of meeting uploaded by the users with meeting id.

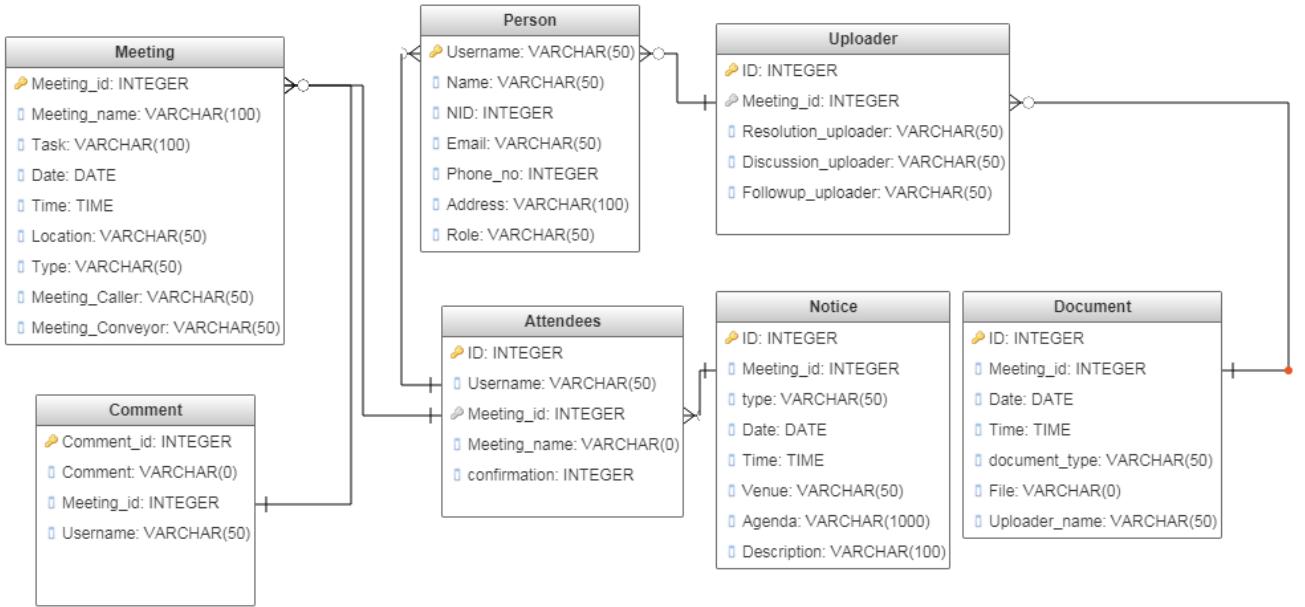


Figure 5.1: Database Schema Diagram

- (e) **Comment:** Each meeting has comment section. Users can interact with each other in comment section. Comments are stored in Comment table.
  - (f) **Attendees:** This table stores list of participants of the meeting and confirmation of the participants of attending the meeting.
  - (g) **Notification:** Notices of upcoming meeting and cancellation of the meeting are stored in this table
3. **Loading data:** In database, dummy information of meeting, meeting scheduling, files of meetings are loaded.

In fig - 5.2, the implementation of database is shown.

Table											Action	Rows	Type	Collation	Size	Overhead
attendees	★	<a href="#">Browse</a>	<a href="#">Structure</a>	<a href="#">Search</a>	<a href="#">Insert</a>	<a href="#">Empty</a>	<a href="#">Drop</a>	185	MyISAM	utf8_general_ci	10 KiB	-				
comment	★	<a href="#">Browse</a>	<a href="#">Structure</a>	<a href="#">Search</a>	<a href="#">Insert</a>	<a href="#">Empty</a>	<a href="#">Drop</a>	13	InnoDB	latin1_swedish_ci	16 KiB	-				
documents	★	<a href="#">Browse</a>	<a href="#">Structure</a>	<a href="#">Search</a>	<a href="#">Insert</a>	<a href="#">Empty</a>	<a href="#">Drop</a>	0	InnoDB	latin1_swedish_ci	16 KiB	-				
meeting	★	<a href="#">Browse</a>	<a href="#">Structure</a>	<a href="#">Search</a>	<a href="#">Insert</a>	<a href="#">Empty</a>	<a href="#">Drop</a>	130	MyISAM	utf8_general_ci	15.2 KiB	-				
notice	★	<a href="#">Browse</a>	<a href="#">Structure</a>	<a href="#">Search</a>	<a href="#">Insert</a>	<a href="#">Empty</a>	<a href="#">Drop</a>	19	MyISAM	latin1_swedish_ci	3.3 KiB	-				
person	★	<a href="#">Browse</a>	<a href="#">Structure</a>	<a href="#">Search</a>	<a href="#">Insert</a>	<a href="#">Empty</a>	<a href="#">Drop</a>	10	MyISAM	latin1_swedish_ci	2.9 KiB	-				
uploader	★	<a href="#">Browse</a>	<a href="#">Structure</a>	<a href="#">Search</a>	<a href="#">Insert</a>	<a href="#">Empty</a>	<a href="#">Drop</a>	5	MyISAM	latin1_swedish_ci	2.2 KiB	-				
7 table(s)											Sum	362	MyISAM	latin1_swedish_ci	65.6 KiB	0 B

Figure 5.2: Implementation of Database

In fig - 5.3, encryption of files using base64 algorithm is shown.

	ag_id	ag_file	date	meeting_id	time	uploader_name
<input type="checkbox"/>	1			1	Acaaca	aaa
<input type="checkbox"/>	2	MTYwNjctY2xhc3NEaWFncmFtLmRvY3g=	NULL	NULL	NULL	NULL
<input type="checkbox"/>	3	NDgyMDEtY2xhc3NEaWFncmFtLmRvY3g=	10 december	1	10 a.m	Munira
<input type="checkbox"/>	4	NzE5NjYtY2xhc3NEaWFncmFtLmRvY3g=	10 december	8	10 a.m	Munira
<input type="checkbox"/>	5	MTI5MDktY2xhc3NEaWFncmFtLmRvY3g=	10 december	8	10 a.m	Munira
<input type="checkbox"/>	6	NjQ5MjctY2xhc3NEaWFncmFtLmRvY3g=	10 december	8	10 a.m	Munira
<input type="checkbox"/>	7	NTMwNzQtY2xhc3NEaWFncmFtLmRvY3g=	10 december	8	10 a.m	Munira
<input type="checkbox"/>	8	ODk5MzctY2xhc3NEaWFncmFtLmRvY3g=	10 december	8	10 a.m	Munira
<input type="checkbox"/>	9	Mzk3MzUtY2xhc3NEaWFncmFtLmRvY3g=	10 december	8	10 a.m	Munira
<input type="checkbox"/>	10	ODIzMzgtY2xhc3NEaWFncmFtLmRvY3g=	10 december	8	10 a.m	Munira

Figure 5.3: Encryption of File

## 5.2 Implementation of Interaction & UX

Implementation of Software describes how a user is able to adapt dynamic system of our software. It will be described in several screen shot of our system.

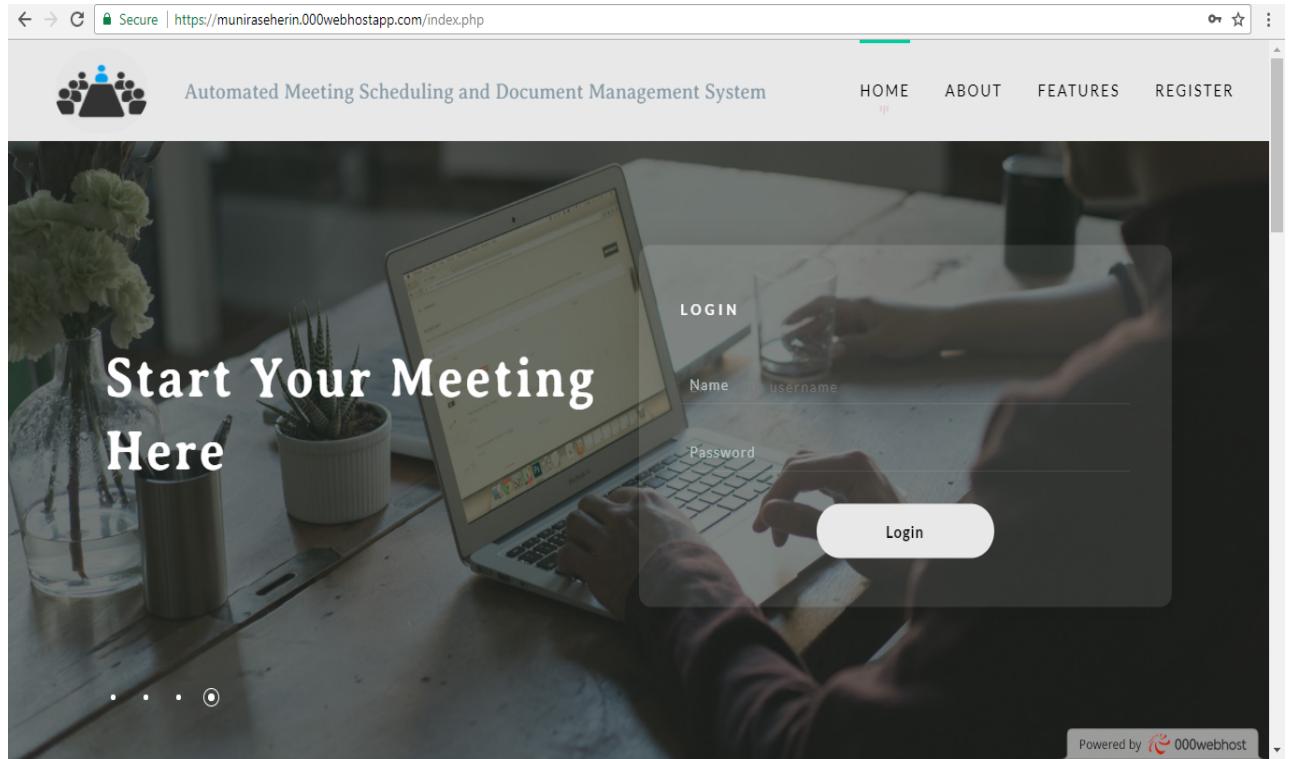


Figure 5.4: User Login

First part of our first page is shown in fig:5.4. User have to be logged in as usual. For this he/she will have to use his/her unique username and password which he selected in registration.



Figure 5.5: About our software

In Fig:5.5, a short summary about our system is shown. Before registration or log in any user can know about our software through the second part of our first page. In fig:5.6, three

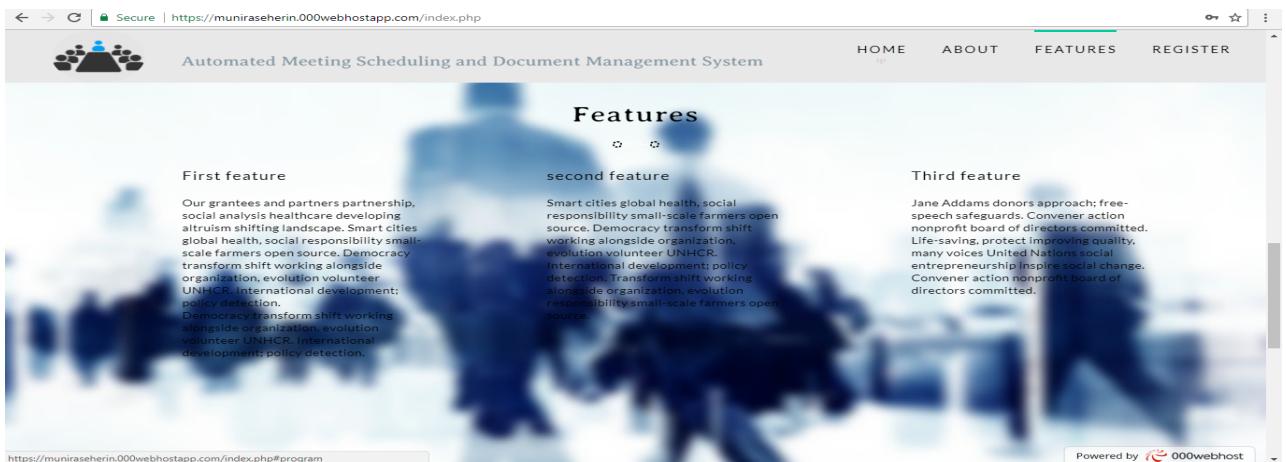


Figure 5.6: Our features

main features of our are shown.

- One of them is the automated scheduling system. While calling a meeting, user does not need to know the time schedules of the meeting attendees. Our system generates the time confliction automatically and inform the meeting caller about appropriate time.
- Another feature is system stores the all documents of the meeting. User can easily find the document of a particular meeting. He/she will quickly see his/her upcoming meeting as he logged into our system.
- Also maintaining the security of the document of meeting is another feature of our system.

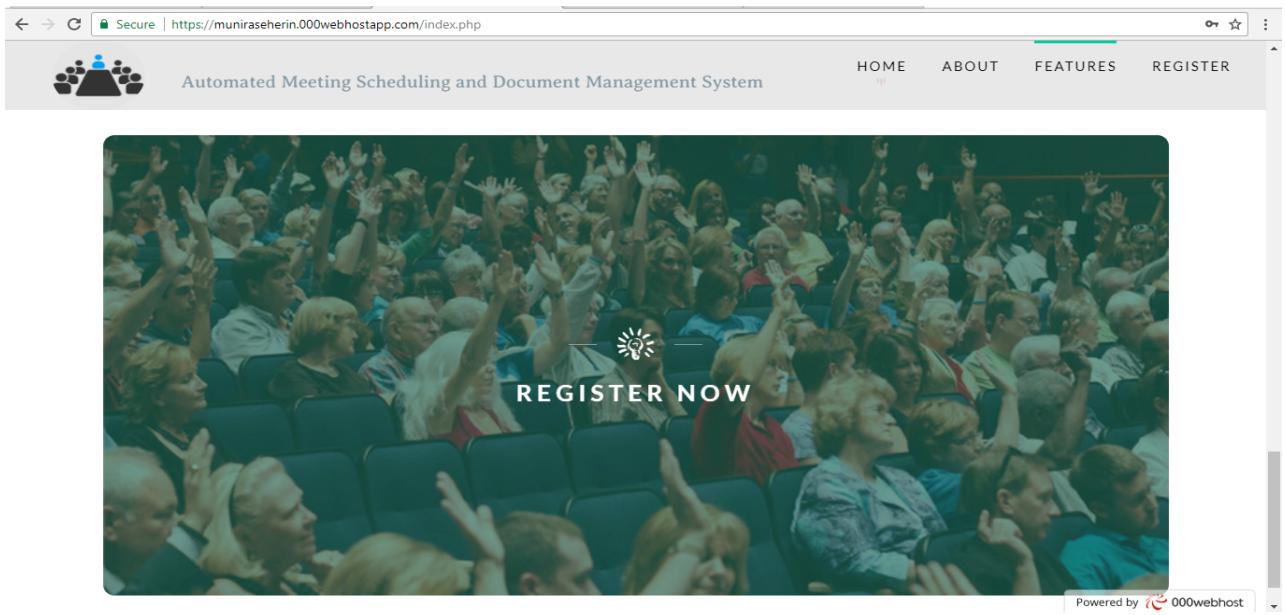


Figure 5.7: Registration

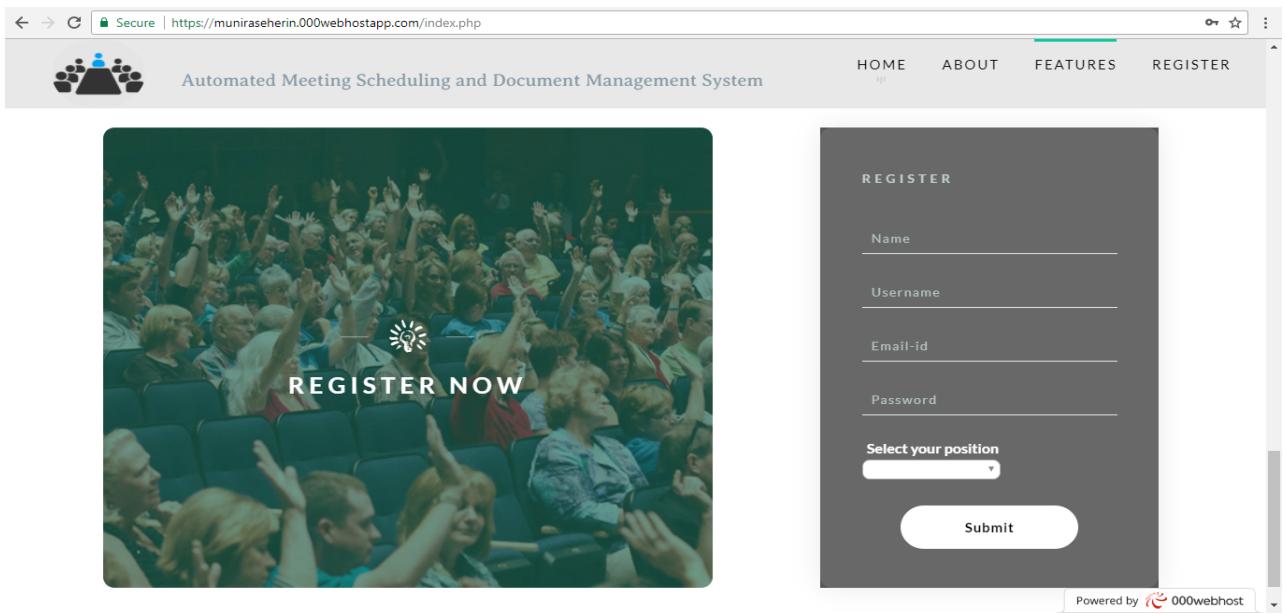


Figure 5.8: Registration2

In fig:5.7 and fig:5.8 registration process is described. To use our system, user will have to be registered to access our system using name, username, email, password and select respective position.

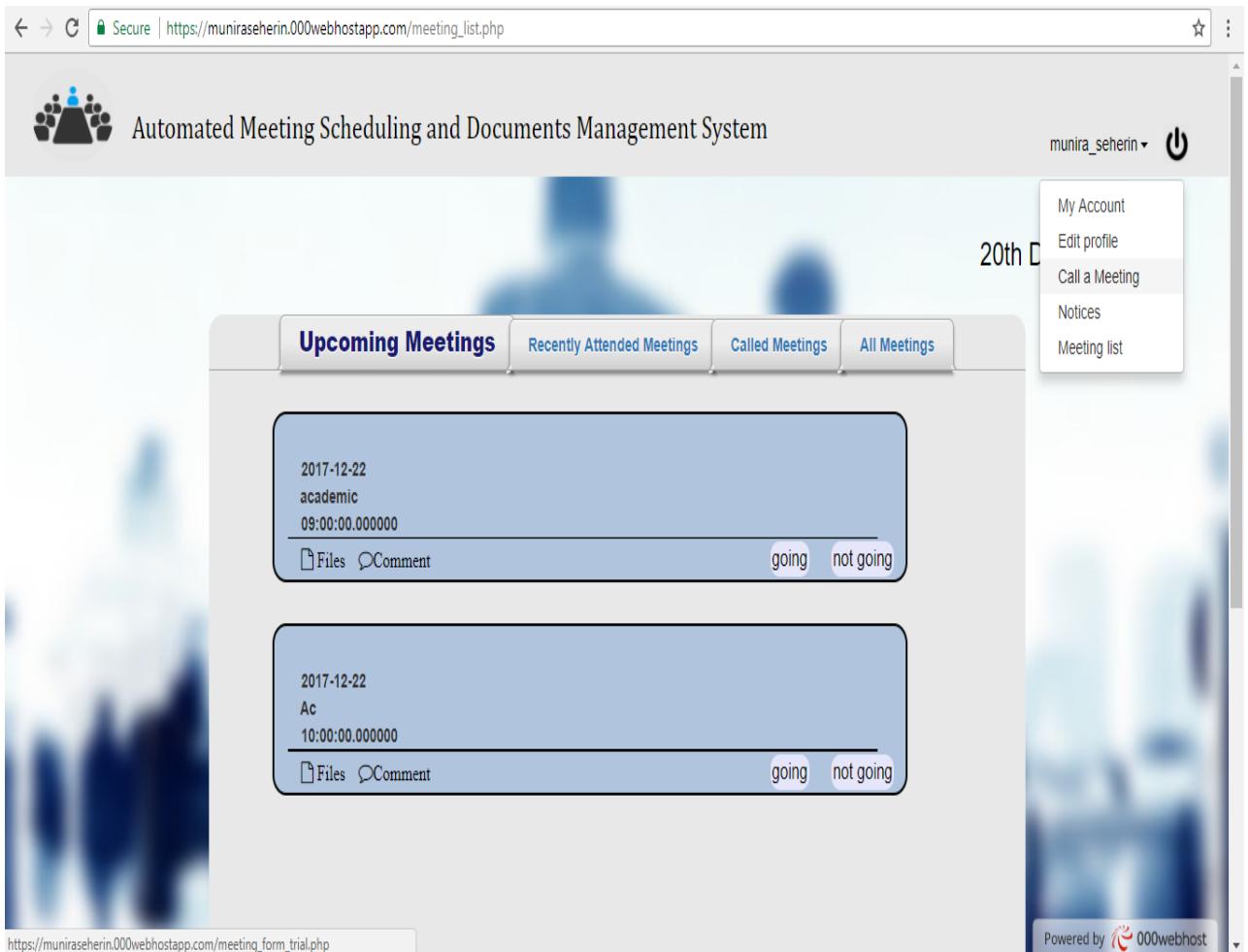
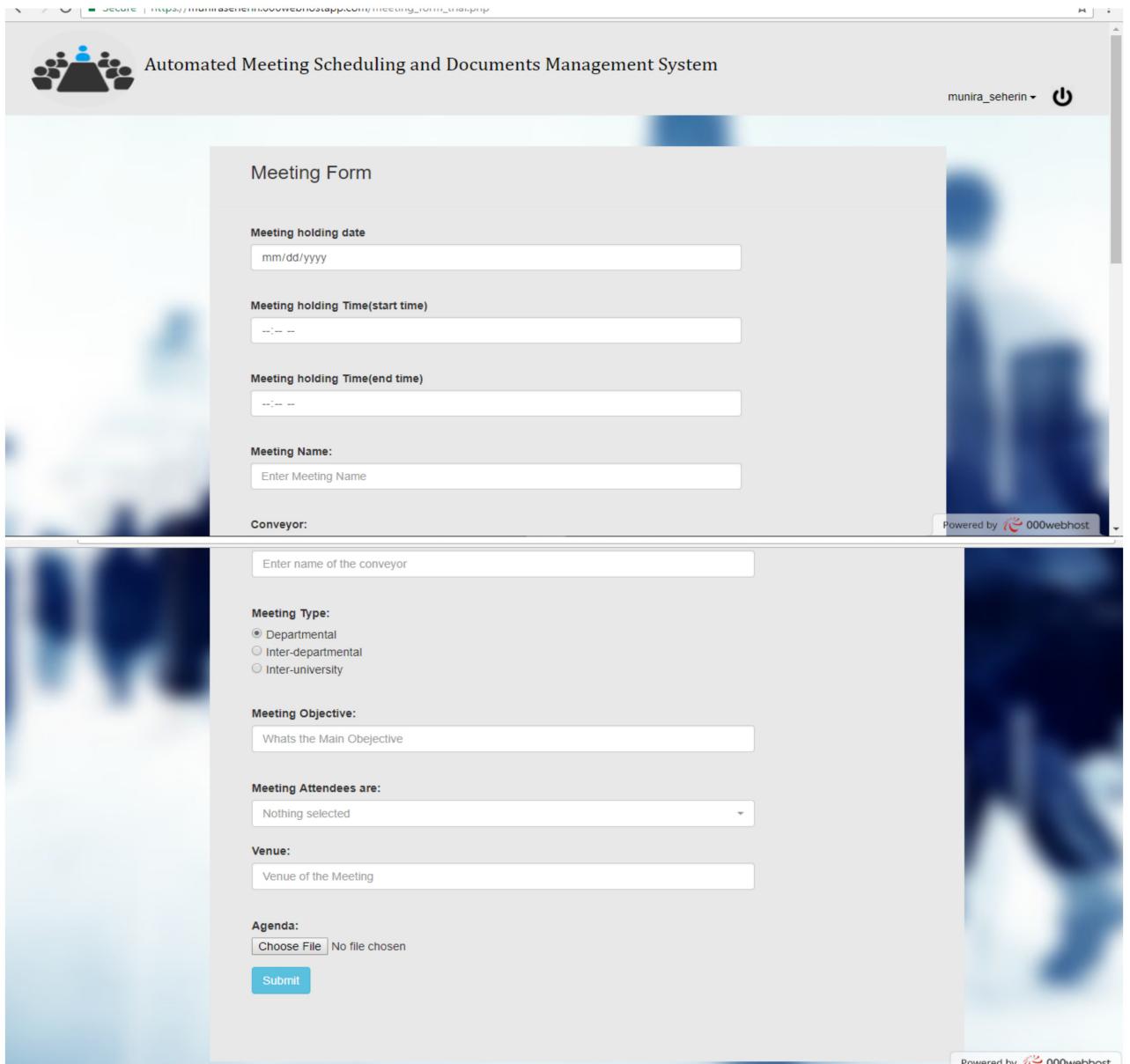


Figure 5.9: Dropdown list

After getting access of our system, user will first see the page of fig:5.9. If he/she is the newly registered person, then list of upcoming meeting will be empty. He has the option to call a meeting which is shown through drop down. There are also other options in the drop down. The options are "My Account" which is the path to the user's profile, "Edit profile" which is the path to edit user's profile, "Notice" is the option to see user's notice and "Meeting List" is the link of the first page to see all the meetings.



The screenshot shows a web-based application for meeting scheduling. At the top, there is a logo of a mountain with a person on top, followed by the text "Automated Meeting Scheduling and Documents Management System". A user profile "munira\_seherin" is visible on the right. The main content area is titled "Meeting Form". It contains several input fields and dropdowns:

- Meeting holding date:** A text input field for "mm/dd/yyyy".
- Meeting holding Time(start time):** A text input field for "HH:MM:SS".
- Meeting holding Time(end time):** A text input field for "HH:MM:SS".
- Meeting Name:** A text input field with placeholder "Enter Meeting Name".
- Conveyor:** A text input field with placeholder "Enter name of the conveyor".
- Meeting Type:** A dropdown menu with three options: "Departmental" (selected), "Inter-departmental", and "Inter-university".
- Meeting Objective:** A text input field with placeholder "Whats the Main Objective".
- Meeting Attendees are:** A dropdown menu with placeholder "Nothing selected".
- Venue:** A text input field with placeholder "Venue of the Meeting".
- Agenda:** A file input field with placeholder "Choose File" and "No file chosen".
- Submit:** A blue "Submit" button.

At the bottom right of the form area, there is a "Powered by 000webhost" logo.

Figure 5.10: Meeting form

It is the process to call a meeting. One has to submit fig:5.10 form before the meeting. Here is main and important field is the date and starting ending time of the meeting. Then comes the conveyor name, meeting type and objective and venue. And finally the attendees of the meeting which will be managed by advanced search option. Meeting caller have to submit the agenda so that meeting attendees get the agenda before coming to that meeting. After submitting the form, if there is no conflict then meeting will be created and attendees will see the meeting to their upcoming meeting list and also get the notice about this meeting.

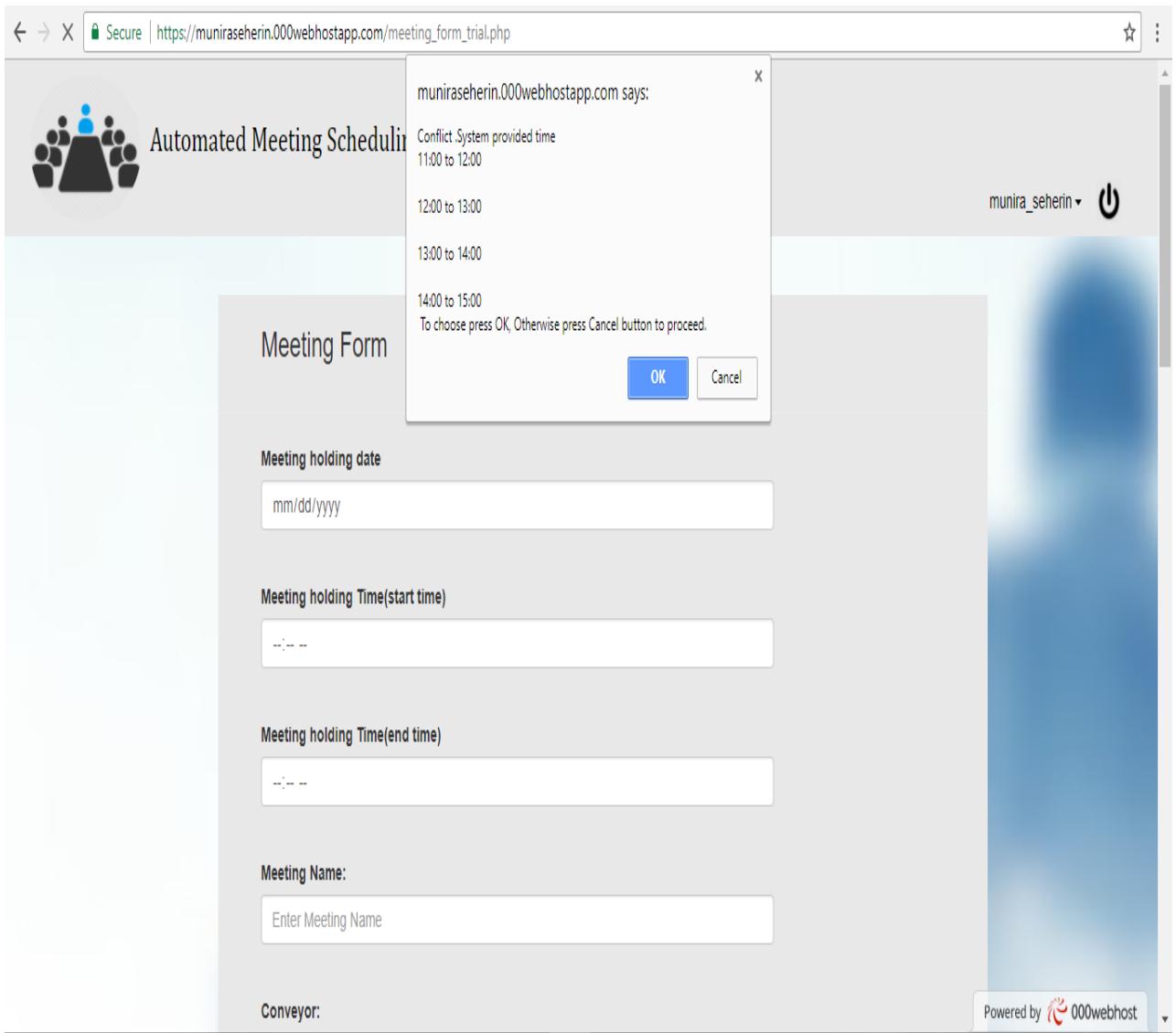


Figure 5.11: Meeting time confliction

Submitting the meeting form means meeting caller want to arrange the meeting at the given time and date. But there can be conflict which is not known to meeting caller. Our system helps him to know that confliction if there is any conflict. Our system calculate the free time of all attendees of that meeting using a algorithm. Then detects the free time when all attendees will be free of that date. That times will be shown through a alert box which is shown in fig:5.11.

Meeting Rescheduling Form

Meeting holding date: 2017-12-22

Meeting holding Time: Select time

Meeting Name: academic

Conveyor: academic

Figure 5.12: Meeting Rescheduling form

Fig:5.12 shows the meeting rescheduling form. If one calls a meeting where meeting time overlaps with the meeting time of other members of the meeting, then our system generates available time for that meeting and provides to that meeting caller. Then user has two options, either he can call the meeting while still there is conflict of time or select the system provided time. If he uses the system provided time, he will have to press 'ok', otherwise 'cancel'.

Meeting Form

Meeting holding date: 2017-12-25

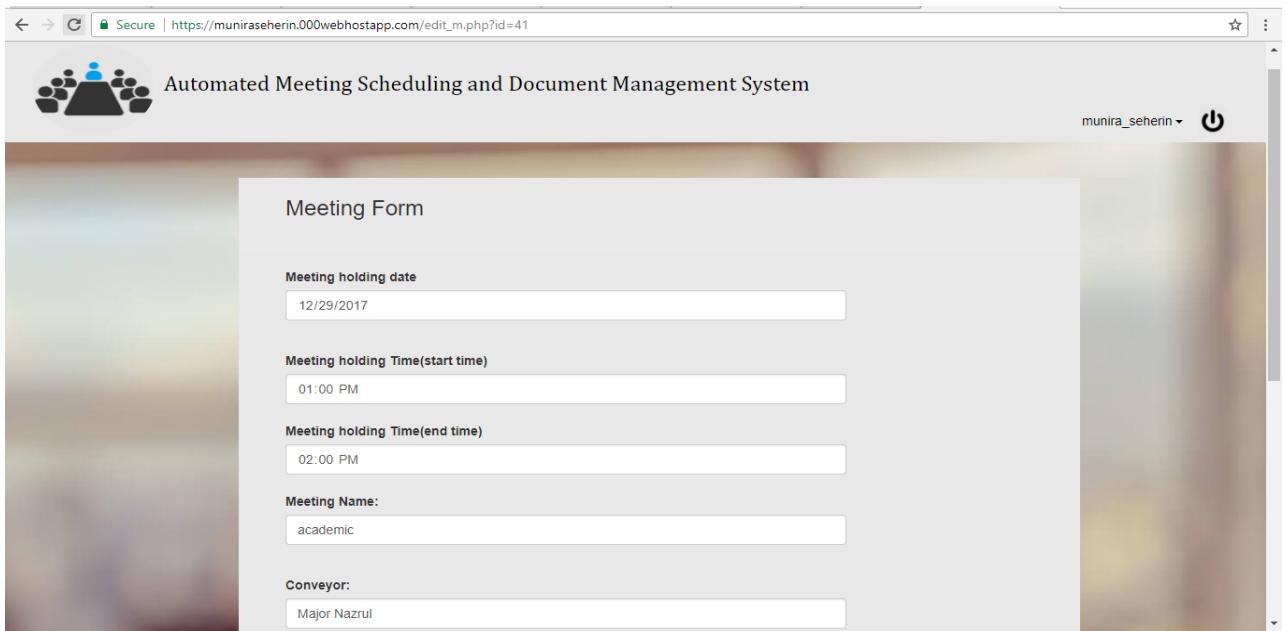
Meeting holding Time: Select time

- 10:00to 11:00
- 11:00to 12:00
- 11:00to 12:00
- 12:00to 13:00

Conveyor: academic

Figure 5.13: System provide time

Fig:5.13 shows the meeting rescheduling time. When conflict of time occurs, then system provides the available time which will be presented through drop down menu. It is noted that only system generated available time will be shown in this drop down menu.

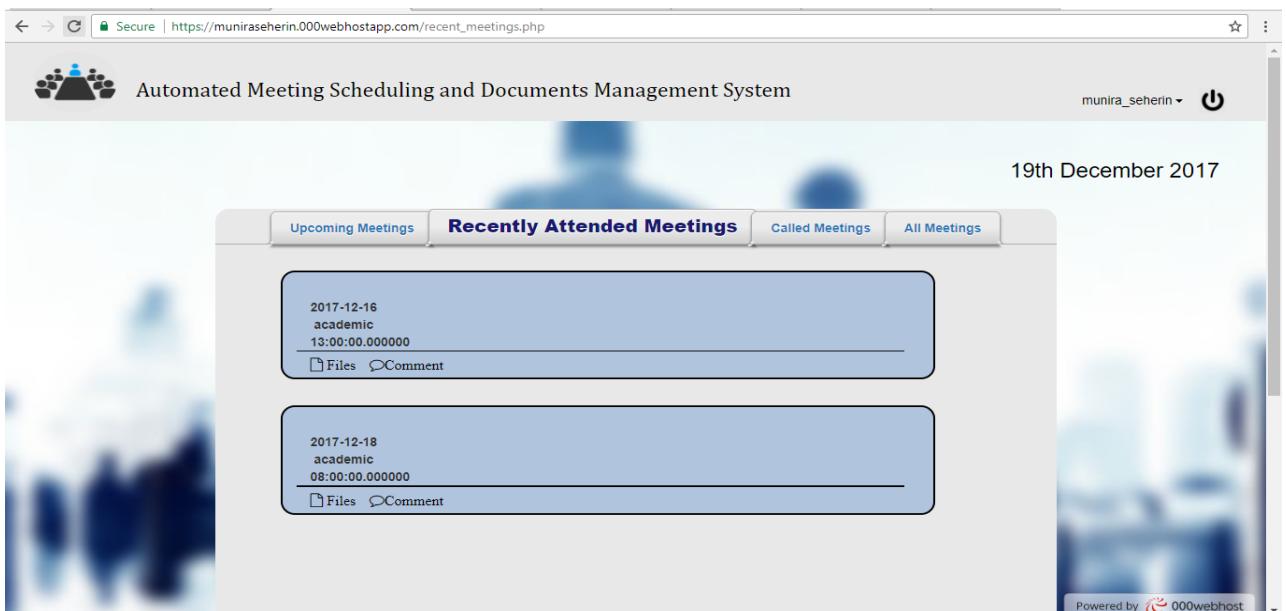


The screenshot shows a web browser window with the URL [https://muniraseherin.000webhostapp.com/edit\\_m.php?id=41](https://muniraseherin.000webhostapp.com/edit_m.php?id=41). The page title is "Automated Meeting Scheduling and Document Management System". The main content is a "Meeting Form" with the following fields:

- Meeting holding date:** 12/29/2017
- Meeting holding Time(start time):** 01:00 PM
- Meeting holding Time(end time):** 02:00 PM
- Meeting Name:** academic
- Conveyor:** Major Nazrul

Figure 5.14: Upcoming meeting

Fig:5.14 is the page where list of upcoming meeting is shown after login of a user. In this page, detailed information of meetings will be shown where he is invited in the meeting. He has the option to attend the meeting or not. If he does not want to attend the meeting, he can click 'not going' and details of the meeting will not be shown on that page.



The screenshot shows a web browser window with the URL [https://muniraseherin.000webhostapp.com/recent\\_meetings.php](https://muniraseherin.000webhostapp.com/recent_meetings.php). The page title is "Automated Meeting Scheduling and Documents Management System". The date displayed is 19th December 2017. The navigation tabs at the top are "Upcoming Meetings", "Recently Attended Meetings" (which is currently selected), "Called Meetings", and "All Meetings". The main content area displays two recent meetings:

- Meeting 1:** Date: 2017-12-16, Name: academic, Start Time: 13:00:00.000000. Buttons:  Files,  Comment.
- Meeting 2:** Date: 2017-12-18, Name: academic, Start Time: 08:00:00.000000. Buttons:  Files,  Comment.

Figure 5.15: Recent Meeting

In fig:5.19, user will see the recent meeting details which has been held during last one month. On this page, user can get the files of the meeting and comment.

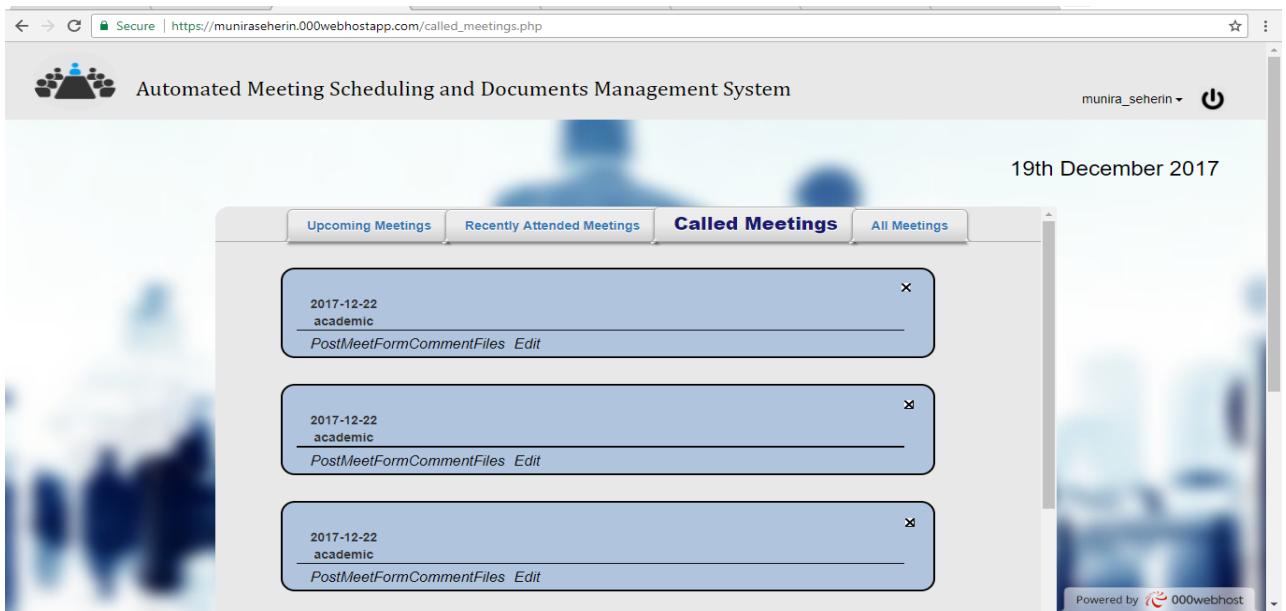


Figure 5.16: Called Meetings

All the meetings that are called by the user are displayed on fig:5.16. On this page, user can edit his upcoming called meetings. That means meeting that have been called by him and have not been held yet, then he can change information of that meeting."PostMeetForm" the option for that meeting which is already occurred. Meeting caller will select the users among the attendees of that meeting to upload the resolution, discussion and followup. As these documents are very important and confidential, it is important to select the person. Only the caller of the meeting can see the "Edit" option to edit the meeting. The caller can only see the cancel option to cancel the called meeting.

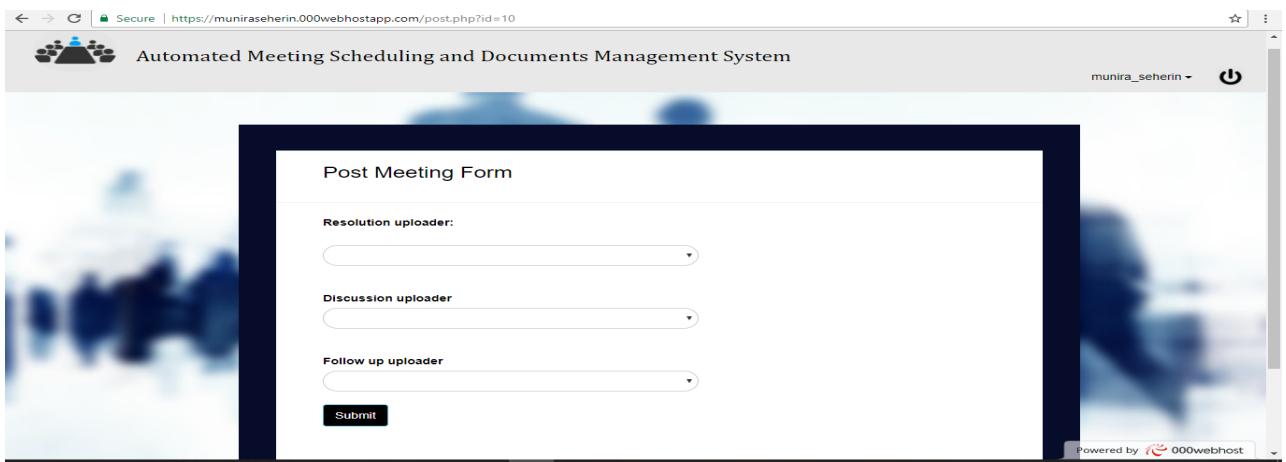


Figure 5.17: Post Meeting Form

It is the process of selecting user who will upload the document of the meeting. Meeting caller have to submit the form of fig:5.17.

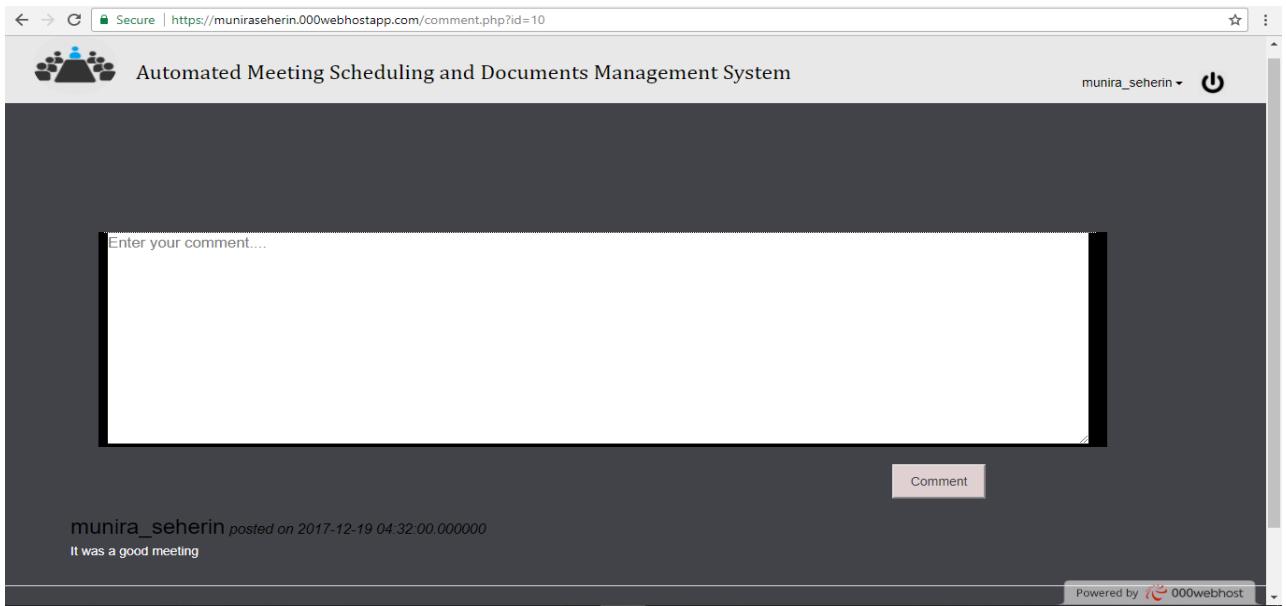


Figure 5.18: Comment area for a particular meeting

Fig:5.18 the comment section for a particular meeting. Attendees of the meeting can comment. Anything about that meeting and any discussion can be done here among the attendees.

A screenshot of a web browser showing a "Meeting Edit Form". The URL in the address bar is https://muniraseherin.000webhostapp.com/edit\_m.php?id=10. The page title is "Security based Meeting Scheduling and Documents Management System". The form fields are: "Meeting holding date" (12/22/2017), "Meeting holding Time(start time)" (09:00 AM), "Meeting holding Time(end time)" (10:00 AM), "Meeting Name:" (academic), and "Conveyor:" (Major Nazrul).

Figure 5.19: Meeting Edit Form

Fig:5.19 shows the meeting edit form for a user who called that meeting and want to change anything about that meeting. This authority is given only the meeting caller and the user which has the upper rank than the meeting caller like Head of any dept of a varsity can change the meeting schedule or cancel the meetings of a lecturer but he cant do it for the meetings which is called by any Dean.

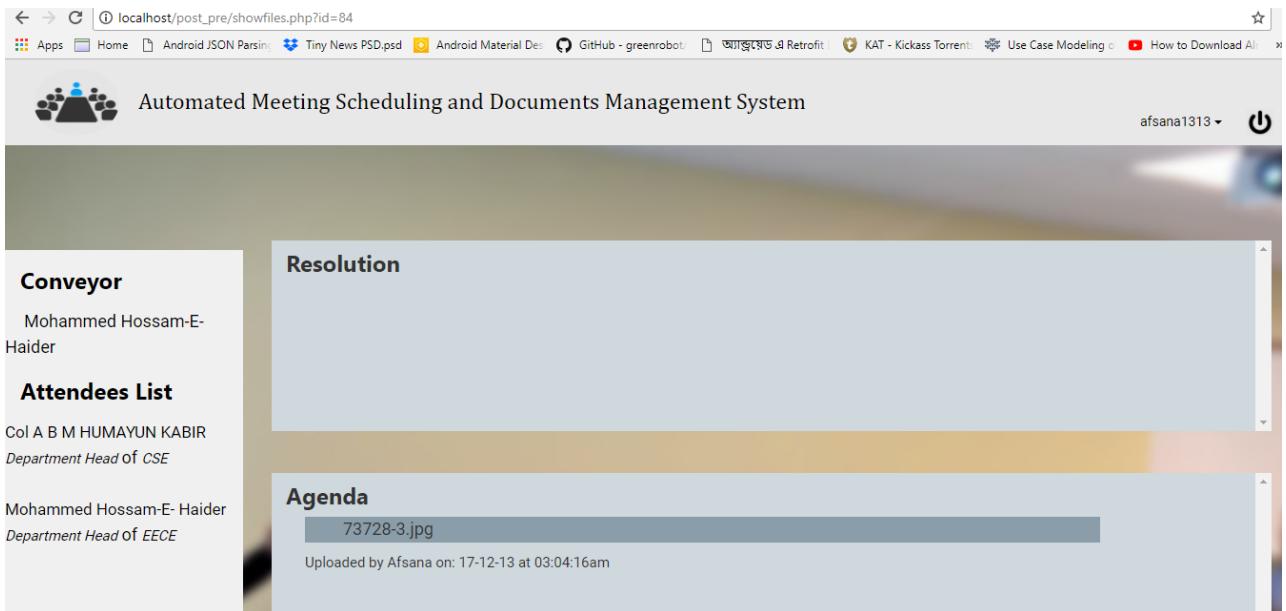


Figure 5.20: Files for a meeting

In fig:5.20 user will get the documents of a meeting. If the meeting is not yet held then user will find only the agenda of that meeting, Others fields will be empty as shown in figure. In this page user will also see the attendees list and conveyor of that meeting. Upload filled will not be visible to the attendee if he is not given the right to upload any file. He can only see and download a file.

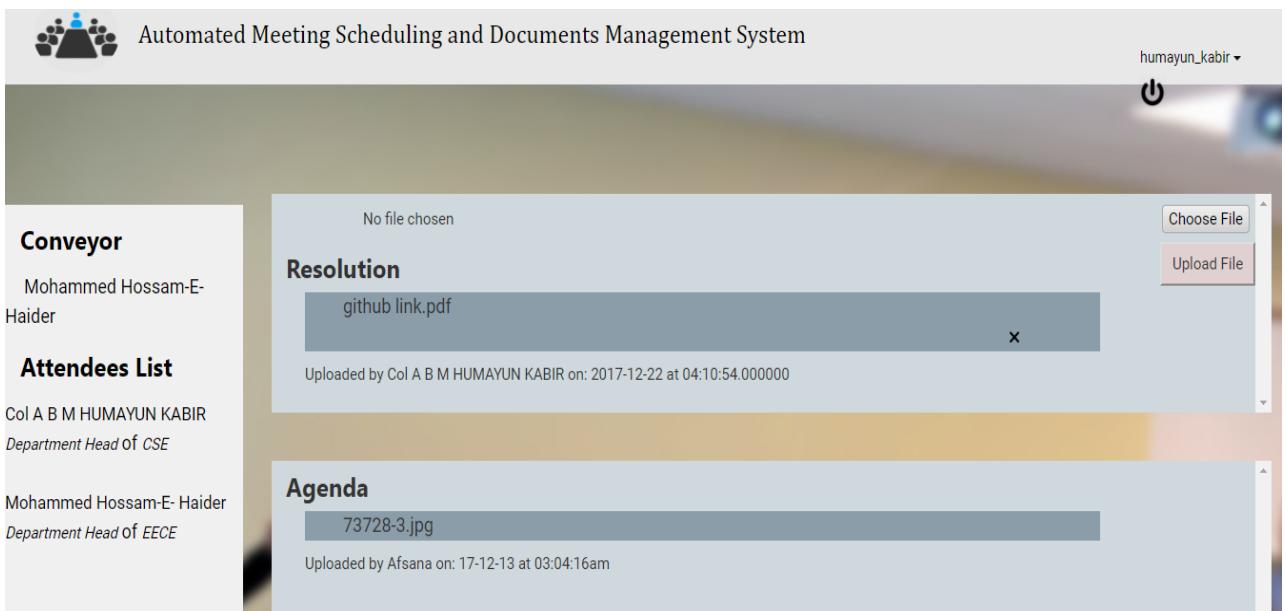


Figure 5.21: Files of that meeting with upload option

Here the files will be uploaded after the meeting is held. Only the person who has the access to upload particular document will see the "Upload File" option and can upload and delete the file that is shown in fig:5.21.

**Conveyor**  
Mohammed Hossam-E-Haider

**Attendees List**  
Col A B M HUMAYUN KABIR  
Department Head of CSE  
Mohammed Hossam-E-Haider  
Department Head of EECE

**Resolution**  
github link.pdf  
Uploaded by Col A B M HUMAYUN KABIR on: 2017-12-22 at 04:10:54.000000

**Agenda**  
73728-3.jpg  
Uploaded by Afsana on: 17-12-13 at 03:04:16am

Figure 5.22: Files of that meeting without upload option

Other attendees will only see and download the files as shown in fig:5.22.

Since 2017  
Since 2016  
Since 2015  
Since 2014

19th December 2017

Upcoming Meetings Recently Attended Meetings Called Meetings All Meetings

2017-12-22  
academic  
12:00am  
 Files  Comment

2017-12-22  
academic  
12:00am  
 Files  Comment

2017-12-22  
Ac  
12:00am

Powered by 000webhost

Figure 5.23: All Meetings

Fig:5.23 shows the all meeting details which has attended by this logged-in user. Here the number of meetings details will be many if he/she is the attendee of many meeting. It has filtered system of these meeting according to years so that user can easily find their desire meetings and files. It makes our system very user friendly.

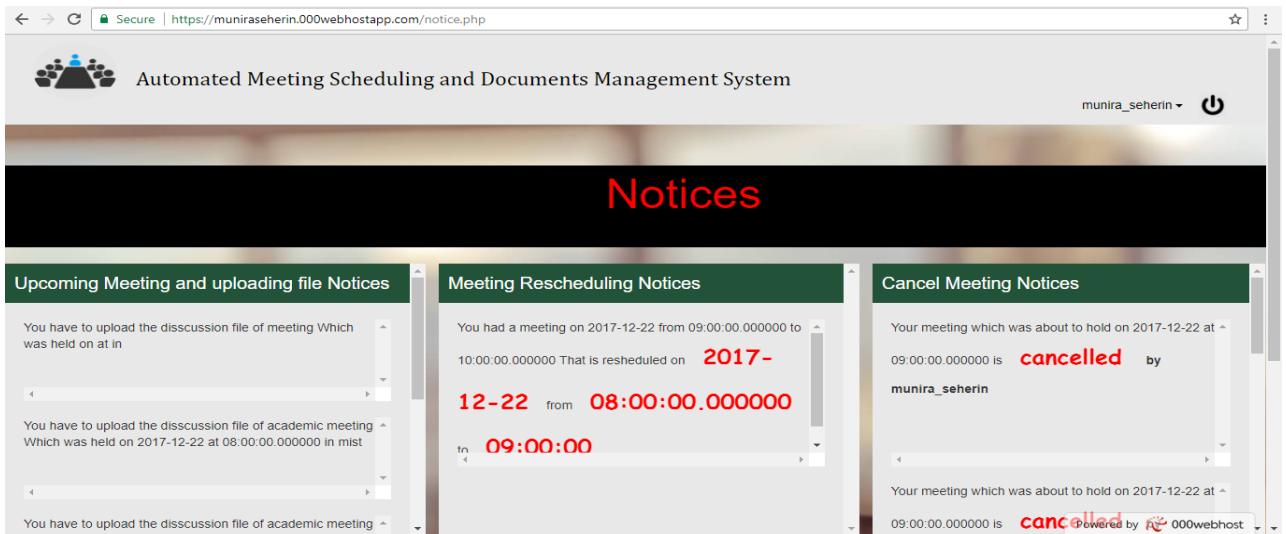


Figure 5.24: Notices

When user log in into our system then he/she will quickly know about the meeting which will occur in short time. If any meeting is rescheduled or cancel, he will also know this through the notices as shown in fig:5.24.



Figure 5.25: User Profile

Fig:5.25 is the user profile page where everything about the user is shown. He/she can update his/her profile by edit profile option.

### 5.3 Testing Implementation

1. **Title:** Call a meeting Successfully call a meeting without conflict of times of participants.

**Description:** A user will successfully call a meeting where meeting time is not overlapped with meeting time of other participants. The invitations will be sent to the users.

**Precondition:** The user must be registered user and must be logged in.

**Test Steps:**

- (a) Click on the user name on top-right side.
- (b) In the drop down menu, click Call a meeting.
- (c) Fill all the information in the new page to call a meeting.
- (d) Click Submit button.

**Expected Result:** Input of the information will be stored in the database. Selected users will get information of upcoming meetings after login.

2. **Title:** Call a meeting Successfully call a meeting with conflict of times of participants.

**Description:** A user will successfully call a meeting where meeting time of the meeting caller is overlapped with meeting time of other participants. A message will pop up stating that there is a time conflict and if the user wants to call the meeting with conflict of meeting time.

**Precondition:** The user must be registered user and must be logged in. The time of the meeting must conflict with time of at least one of the meeting participants.

**Test Steps:**

- (a) Click on the user name on top-right side.
- (b) In the drop down menu, click Call a meeting.
- (c) Fill all the information in the new page to call a meeting.
- (d) Click Submit button.
- (e) It is checked if there is any conflict of time of the meeting with time of other participants.
- (f) If there is any conflict, a search of available time for all the selected members of the meetings is done.
- (g) Available meeting times are shown and one time is selected by the user which will be saved in the database

**Expected Result:** The page will be as - figure - [5.11](#). Input of the information will be stored in the database. Selected users will get information of upcoming meetings after login.

3. **Title:** Meeting List - Showing list of upcoming meetings.

**Description:** : When user will log in, list of upcoming meetings will be on the page.

**Precondition:** User must be registered user and must log in.

**Test Steps:**

- (a) Log in to the account.
- (b) List of upcoming meetings will show up.
- (c) For each meeting, two options will be available, going or not going.
- (d) If user confirms to attend the meeting, user will click on going.
- (e) If user dont want to attend the meeting, then user will click on not going.

**Expected Result:** If user clicks on going i.e. confirms to attend the meeting, he will be able to see information of that meeting. Otherwise, this meeting will not be shown on the list further.

4. **Title:** Files of the meeting Showing files of a meeting.

**Description:** For every meeting, there is an option to view the files of the meeting. If the link is clicked, a new page is opened where all the files of the meeting is shown.

**Precondition:** : User must be logged in and navigated in meeting list page, recent attended meeting page, called meetings or all meetings page.

**Test Steps:** In the list of the meetings, click on files icon.

**Expected Result:** In a new page, all the files of the meeting with upload option and without upload option are shown as figure - [5.21](#) and figure - [5.22](#).

5. **Title:** Comments of the meeting Showing the comments of the meeting.

**Description:** For every meeting, there is an option to interact with others in comment section. In a page, all comments of meeting is shown.

**Precondition:** User must be logged in and navigated in meeting list page, recent attended meeting page, called meetings or all meetings page.

**Test Steps:** In the list of the meetings, click on Comments icon.

**Expected Result:** In a new page, all the files of the meeting are shown as figure - [5.18](#).

# CHAPTER 6

## CONCLUSION AND FUTURE SCOPE

### 6.1 Discussion

Our aim was to make the conduction of meeting easy and save the documents of meeting properly i.e. in an organized way. Our software can detect the time conflict of the meeting while calling. Available free time of the day is shown so that caller can easily schedule the meeting. At the same time, all the records are saved properly. So finding a file is simple and easy.

Database security is also maintained in the software. Database security is a very broad sector that deals with many issues. Some of these issues are: legal and ethical issues regarding who has rights to access certain information, policy issues as to what types of information should not be made publicly available, security policies within an organization as to who should have what level of access to certain classified information [8]. In our software, only organization related personnel can use the software. No other person will have access to the software. Only the meeting caller or high ranked person of the organization than the caller can edit or cancel the called meeting. The documents of a particular meeting can only be accessed by the attendees of that meeting. The caller fixes the persons from the attendees to upload files related to the meeting. None but the up-loader can delete the file. Other attendees can see and download the documents but can't delete them.

The comment section of the software will help the meeting attendees to discuss about any problem related to the meeting in online. The comments can only be seen by the attendees.

### 6.2 Conclusion

Meetings are an integral part of any company and are typically where employees get to discuss and communicate ideas as well as where they get feedback on the status of different work processes which then allows for informed decisions to be made. Without meetings, employees would essentially be working almost in isolation and work flow would be inhibited. Although meetings are critical for companies, planning for the meeting should not consume more time and energy than required.

By completing the full project, we think, the aim is achieved. It will save time, money and will be an easy way of connecting the persons related to an organization. As a result, this software will be a dynamic tool for meeting scheduling and managements.

### 6.3 Further Developments

We have developed a prototype of the software which is working on web version. In future we would work on more precise automation, Securely storing & transforming documents, online meeting management, video and audio file upload and android version of the software.

- **Precise Automation:**

We worked on automatically providing free time for meeting call. But it works only for one hour meeting. In future, we would like to work on any duration of meeting. It can be multiple hours and even for few minutes. The algorithm will be modified accordingly. Now the automation algorithm is in primary stage.

- **Securely Storing & Transforming Files:**

At present, base64 algorithm is used to encode the content, size and type of every file that are saved in database. When the attendees want to download the file, it is decoded by the system. In future, to make the system more secure, we will keep a key for encryption. And the key will be sent to the user by SMS or Email. If he/she wants to download the file, then the system will ask for the key.

- **Online Meeting Management:**

Our software presently schedule meeting for offline purpose. In future we have plan to work on online based meeting where an attendee will be able to present virtually. This will save the time.

- **Video & Audio File Upload:**

As the software is working on document management, it can manage images, pdf, ppt and txt files. In future we will work to preserve audio and video files.

- **Android Version:**

The software is in web version. In future android version of the software will be developed.

## REFERENCES

- [1] Maheswaran, R. T., Tambe, M., Bowring, E., Pearce, J. P., Varakantham, P. (2004, July). Taking DCOP to the real world: Efficient complete solutions for distributed multi-event scheduling. In Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 1 (pp. 310-317). IEEE Computer Society.
- [2] Nazia, S., and S. Ekta. "Online Appointment Scheduling System for HospitalsAn Analytical Study." *Int J Innov Res Sci Eng Technol* 4.1 (2014): 21-7.
- [3] Zhu, Kenny Qili, and Andrew E. Santosa. "A meeting scheduling system based on open constraint programming." International Conference on Advanced Information Systems Engineering. Springer, Berlin, Heidelberg, 2002.
- [4] Copek, Peter George, et al. "Event scheduling with optimization." U.S. Patent No. 7,343,312. 11 Mar. 2008. APA
- [5] Shaw, Donna Sue, Gregory Scott Klabish, and Gail Elaine Slapikoff. "Collaborative management of activities occurring during the lifecycle of a meeting." U.S. Patent Application No. 12/039,789.
- [6] Karstens, Christopher. "System and method for dynamic meeting agenda with event firing progress indicators." U.S. Patent Application No. 10/648,991. APA
- [7] Sen, Sandip. "Developing an automated distributed meeting scheduler." *IEEE Expert* 12.4 (1997): 41-45.
- [8] Li, An. "Scheduling Information System: the Design and Development of an Online Database for Tracking Scheduling Information." (2004)
- [9] Herlea, Thomas, et al. "On securely scheduling a meeting." IFIP International Information Security Conference. Springer, Boston, MA, 2001.
- [10] Wu, Kehe, et al. "Design and implementation of encrypted and decrypted file system based on USBKey and hardware code." AIP Conference Proceedings. Vol. 1839. No. 1. AIP Publishing, 2017.
- [11] Alexander, Geoffrey David, and Mohamad R. Salahshoor. "Method and system for automatically scheduling and managing agendas for presentation-style meetings." U.S. Patent Application No. 11/309,048.
- [12] Krasnick, Steve, and Marc Schechter. "Online meeting planning program." U.S. Patent Application No. 09/836,757. APA

- [13] Lee, Raymond, et al. "Method, system and program product for interactive electronic meeting scheduling." U.S. Patent Application No. 10/173,264.
- [14] Kuo, Jacky, et al. "Meeting management system." U.S. Patent Application No. 10/636,439.
- [15] Horvitz, Eric. "Time management representations and automation for allocating time to projects and meetings within an online calendaring system." U.S. Patent Application No. 10/882,010.
- [16] Khattak, Abid Saeed, Malik Sikander Hayat Khiyal, and Sanam Shahla Rizvi. Verification validation of a multi agent meeting scheduling simulation model. *Journal of Computer Science* 2, no. 1 (2014): 47-64.
- [17] *https://stackoverflow.com/questions/14999348/what-is-the-meaning-for-design-specification-in-documentation-of-a-project*,  
Last accessed on 20December, 2017
- [18] *https://www.w3schools.com/xml/ajax\_intro.asp*,  
Last accessed on 20December, 2017
- [19] *https://en.wikipedia.org/wiki/Base64*,  
Last accessed on 20December, 2017
- [20] *https://www.1keydata.com/datawarehousing/logical-data-model.html*,  
Last accessed on 20December, 2017