

Abstract

The purpose of this project is to solve the problems in manual management of activities pertaining to the clubs running within a University. In recent times, where technology is at an all-time high, it can easily assist an individual to make his/her life less troublesome. By receiving a complete set of requirements from the stakeholders we prepared an efficient design of the end product incorporating all the features and functionalities desired by stakeholders. The standard procedures for requirement collection are sincerely obeyed and the all the steps are clearly explained in this document.

0.1 Acknowledgement

We are over helmed in all humbleness and gratefulness to acknowledge our depth to all those who have helped us to put our ideas and assigned work, well above the level of simplicity and into something concrete. We would like also like to extend our gratitude to our Software Engineering Lecturer, Dr Trisila Devi Nagavi, who guided and provided us with all the aid we required to bring forth this submission. We would also like to thank all the different stakeholders of the project who agreed to take out time from their busy schedules to provide us with in-depth knowledge about the ground problems of the project.

0.2 Project Team details

0.2.1 Arunima Rashmi Giri



USN:01JST16CS015
Roll No: 6

0.2.2 Sanmayee Mahabaleshwar Hegde



USN:01JST16CS093
Roll No: 39

0.2.3 Sindhu Rashmi M S



USN:01JST16CS102

Roll No: 47

0.2.4 T S Kushal



USN:01JST16CS133

Roll No: 62

Contents

0.1 Acknowledgement	1
0.2 Project Team details	2
0.2.1 Arunima Rashmi Giri	2
0.2.2 Sanmayee Mahabaleshwar Hegde	2
0.2.3 Sindhu Rashmi M S	3
0.2.4 T S Kushal	3
1 Introduction	7
1.1 Introduction to problem domain	7
1.1.1 Purpose	7
1.1.2 Document Conventions	7
1.1.3 Intended Audience and Reading suggestions	8
1.1.4 Product scope	8
1.2 Problem statement	8
1.3 Project schedule	8
2 System requirement engineering	10
2.1 Overall descriptions	10
2.1.1 Product perspective	10
2.1.2 User class and characteristics	11
2.1.3 Diagramatic Description	12
2.1.4 Operating Environment	17
2.1.5 Design and Implementation Constraints	18
2.2 External Interface Requirements	19
2.2.1 User interface:	19
2.2.2 Hardware interface:	19
2.2.3 Software interface:	20
2.2.4 Communication interface:	20
2.3 Requirement Engineering	20
2.3.1 Inception:	20
2.3.2 Elicitation	22
2.3.3 Elaboration	36

2.3.4	Negotiation	38
2.3.5	Specification	39
2.3.6	Validation	42
2.3.7	Requirement Management	44
3	System design	46
3.1	Component Level Design	47
3.1.1	Component - Member Registration	47
3.1.2	Component - Event Planing	48
3.1.3	Component - Financial Management	49
3.1.4	Component - Student Tracking	50
3.1.5	Component Interactions	51
3.2	Architecture Design	52
3.2.1	Architectural Style	53
3.2.2	Architectural Genre	53
3.2.3	Design Approach	54
3.3	Data/Class Design	59
3.3.1	Entity Relation Description	59
3.3.2	The Identified Entities:	59
3.3.3	The identified Relationships:	60
3.3.4	Schematic Description	61
3.4	Interface Design	61
3.4.1	'Log in' activity diagram	61
3.4.2	'Log out' activity diagram	62
3.4.3	'Registration' activity diagram	62
3.4.4	'Volunteers Registration' activity diagram	63
3.4.5	'Membership selection' activity diagram	63
3.4.6	'Rating of an event' activity diagram	63
4	System Implementation	71
4.1	Introduction	71
4.1.1	Purpose	71
4.1.2	Objective	71
4.2	Implementation	72
4.2.1	Event	72
4.2.2	Transact	72
4.2.3	Tasks	73
4.2.4	Passbook	74
4.2.5	Reports	74
4.2.6	Event Log	76
4.2.7	Power for Student Co-ordinator	76

4.2.8	Student Dashboard	77
4.2.9	Registration	78
4.2.10	Rating	78
4.3	Coding	79
4.3.1	Introduction to SQL	79
4.3.2	Relational Algebraic Queries	80
4.3.3	Queries designed using SQL Commands	80
5	System testing and result analysis	86
5.1	Introduction	86
5.1.1	Purpose	86
5.1.2	Objective	86
5.2	Overall Strategy and Approach for Testing	87
5.2.1	Usability Testing	87
5.2.2	Functional Testing	87
5.3	Testing Implementation	88
5.3.1	Unit Testing	88
5.3.2	Acceptance Testing	89
6	Conclusion and future work	91
6.1	Conclusion	91
6.2	Future Work	92
6.3	References	93

Chapter 1

Introduction

1.1 Introduction to problem domain

1.1.1 Purpose

The purpose of this documentation is to give detailed description of requirements and features which would be provided by our software. The main objective of our project, “Club Management System”, is to provide services to the stakeholders which would help them in organizing all their club related activities. We hope to provide these services by automating most of these activities which is presently done manually by the stakeholders and their teams, by automating these activities we hope to decrease the human errors which occur and also to make the entire process of managing a club quicker and more efficient. With the help of this document we hope to explain the requirements of the services which we are attempting to provide and also to describe the features, interfaces, functionalities and constraints under which our service would be provided. The scope of this software service is to ease some of the burden of the tremendous paperwork and manual activities that have to be addressed on a day-to-day basis by the stakeholders. We hope that our service can be widely adopted by the stakeholders to manage all of their club related activities and also to make it secure enough that the stakeholders would not have to worry about their privacy.

1.1.2 Document Conventions

We have based this document on the IEEE System Requirement Specification (SRS) template. We decided to use the “Times New Roman” font for all of our documentation. In this document the main headings are in bold and have the font size of 16 and the subheadings are as well bold and have a font

size of 14 and the rest of the document has a font size of 12. We decided to use the convention of writing any and all of the concepts which describe our service in a vital manner in italics.

1.1.3 Intended Audience and Reading suggestions

We intend this document to be an effective reference for understanding the various requirements put forward by the different stakeholders involved and also the approach adopted by us to handle each of them. Readers are suggested to go through the document conventions and the product scope to have an overview of the domain of the project.

1.1.4 Product scope

The product concerns itself with providing an efficient and virtually paperless mechanism of providing a common platform for the stakeholders to store all data related to their clubs such as information about their members, financial transactions, various events which they have conducted and hope to conduct in the future. We hope to provide a considerable improvement to the financial health of all the club availing our services. To do so we plan to a rugged level of security and encryption in the form of digital signatures, two-step verification and biometric authentication to name a few. The service would facilitate notification and broadcast messaging services for club publicity. The product would considerably reduce the amount of manual efforts and errors in the management of a club.

1.2 Problem statement

To build a software with a capability of managing the clubs in the University, to overcome the problems in manual management.

1.3 Project schedule

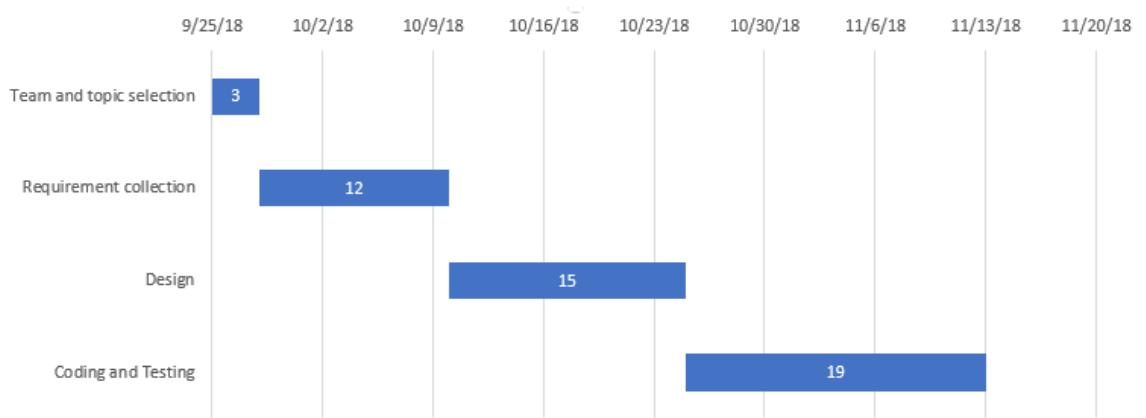


Figure 1.1: Project schedule

Chapter 2

System requirement engineering

2.1 Overall descriptions

2.1.1 Product perspective

At the time of the inception of the product the perspective of the product was basically to provide an open service of managing activities of a club or association both mainstream and background in an automated manner as much as possible. Our very aim was to eliminate the burden of every person involved in putting up a successful working club, be it the administration, the coordinating faculty, the student coordinators, the office bearers to the very extent of the end users or receptors that is the students. Some of the various features we had in mind for implementation were:

- Cross platform support: Offers operating support for most of the known and commercial operating systems.
- User account: The system allows the user to create their accounts in the system and provide features of updating and viewing profiles.
- Number of users being supported by the system: Though the number is precisely not mentioned but the system is able to support a large number of online users at a time.
- Financial tracking: The system facilitates a rigid financial monitoring and recording system almost paperlessly.

- Work management: The system also supports functionalities for keeping track of the workloads of different members, the involvement of the volunteers and the details of the participants of every event.
- Discussion Forum: Provides users with a platform to discuss and help each other with their problems.

Ticketing system: Allows user to submit his issue to the admin in case his problems are not solved by FAQs and discussion forums.

FAQs and Suggestion section: Frequently asked section contain answer of problem which the product users frequently faced and let users provide the administrators with any functionality suggestion.

2.1.2 User class and characteristics

The product is so designed that it can be accessed and utilized by various types of users. They are mentioned below.

2.1.2.1 Club members

The members of the club are involved in activities such as event management, financial transactions, tracking of volunteers and participants etc. To accomplish all of these activities they would need access to the event logbook and also the registry of all the students who are acting as either participants or volunteers. They would also need access to update the date, venue, event description etc. for a particular event.

2.1.2.2 Volunteers

The volunteers for a particular event are involved in activities such as hosting the event, teaching a session, helping out the participants etc. For these tasks they would need access to the event details and also, they will need a method for noting down the various tasks that they have been assigned for a particular event.

2.1.2.3 Participants

The participants are the basic uses who would register for an event hosted by a particular club. They would only need to be able to register for an event and see the details and prerequisites of the event such as knowing a particular programming language for example.

2.1.2.4 Club coordinators

The coordinators of the club would need complete access to all of the details of the clubs such event details, financial records, member registry etc. They would also need to be able to change any and all details regarding to the any of the events and the club in general. They would also need a method to keep track of all of their members and volunteers and also to check their performance.

2.1.2.5 Teacher coordinators of the club

The teacher coordinators would need to be able to keep track of the progress of all the events conducted by the club and an easy way to give their input whenever required. They also need to able to give authorization for financial and technical requests made by the member with ease and without having to worry about security issues.

2.1.2.6 HOD of the department to which the club belongs

The HOD of the department only has to oversee the various activities without being actively involved in them. They would require an easy and effective method to grant authorization to request from the teacher coordinators and members alike. The HOD would also need a provision to stop any activity when feel that it is not going according to plan.

2.1.2.7 Principal of the institution

The principal would have to have access to all of the clubs in the institution and all of their data. The principal would need a method of checking on the progress of all the clubs and also give input wherever it is required. The principal should also have complete monopoly over the financial transactions of all the clubs and should also be able to stop a transaction if it is not properly authorized.

2.1.3 Diagrammatic Description

2.1.3.1 Product Perspective

Product administrator has the freedom to create new clubs on the request of club managers. There may be n number of clubs as shown in the figure. Members, coordinators, HOD are associated with these clubs. A single

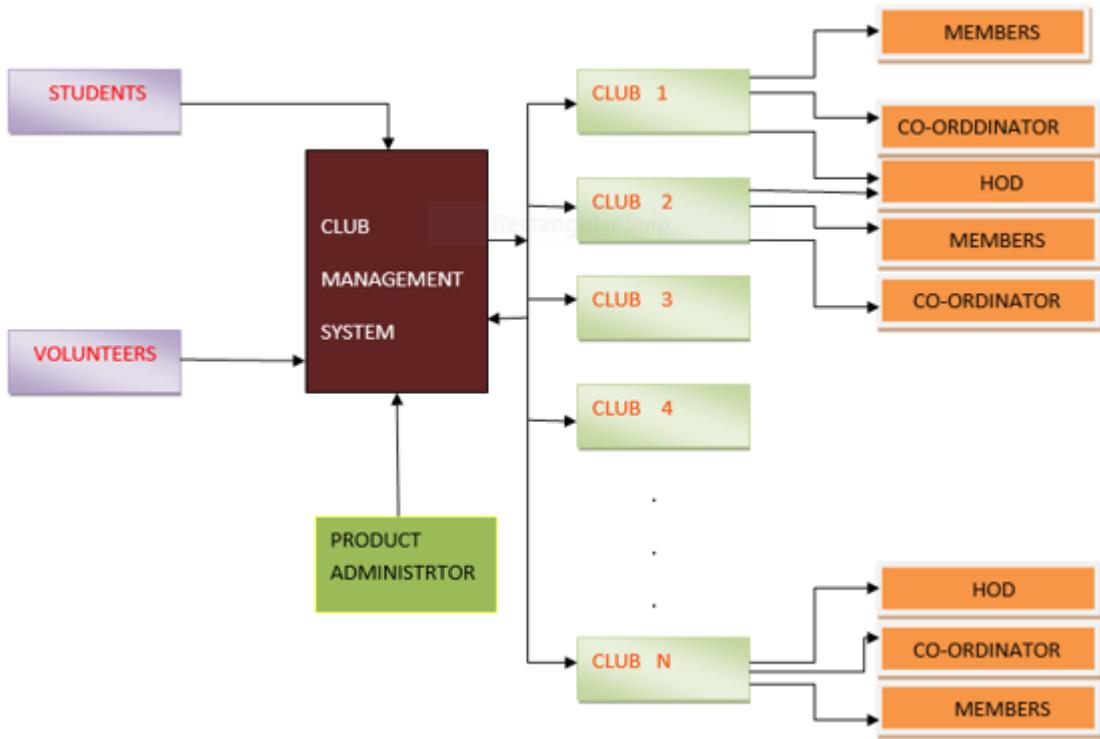


Figure 2.1: Product perspective Diagram

HOD can be associated with more than one club. Students fetch information about upcoming events. Club managers will receive volunteering request and requests for registration from students. When we interacted with IEEE members they came up with a problem that information about students who have registered for membership will be known by them but students will not receive any kind of information. So, a confirmation receipt will be sent to the students' e-mail account containing information about how much amount they have paid. Earlier this was done manually. We would now be providing digitalized receipt to all the students.

We try to build customer's trust by not allowing all students to get club managers phone number easily. Club management system keeps the students updated with real-time message notifications (about events) and safely stores information about students and club financial status.

For most of the events that students and volunteers register, through

the club management system will be able to add suggestions to let club managers know about what students are expecting to get by attending their events for instance if an event is about data structures then the students who are weak in graphs, trees can specify in the suggestion box about their difficulties. Based on the suggestions received they can prioritize their topics while conducting sessions. If you are a club member and want to participate in an event conducted by the same club you will be provided with discounts.

2.1.3.2 Use cases

This subsection extends upon the functional requirements through the presentation of detailed use cases to facilitate an unambiguous and clear view of how the end-users interact with the subject club managers, the actors (end-users) involved in the use cases, a use case diagram and detailed use case descriptions are provided. The use cases that find representation are logout, login, certifying participate, give suggestion, volunteering authentication, request for volunteering, event management, registering for events, completing the task, rating an event, assigning the task, pay bill, financial control, fetching information about events, statistical reports.

2.1.3.3 Actors

There are five actors in the club managerial system, club manager, student, volunteers, HOD, principal, coordinator. Some of the use case descriptions are shown below:

Table 2.3.2.1 presents the Log In use case description to show the interaction between a club manager and an APP when logging in to the system.

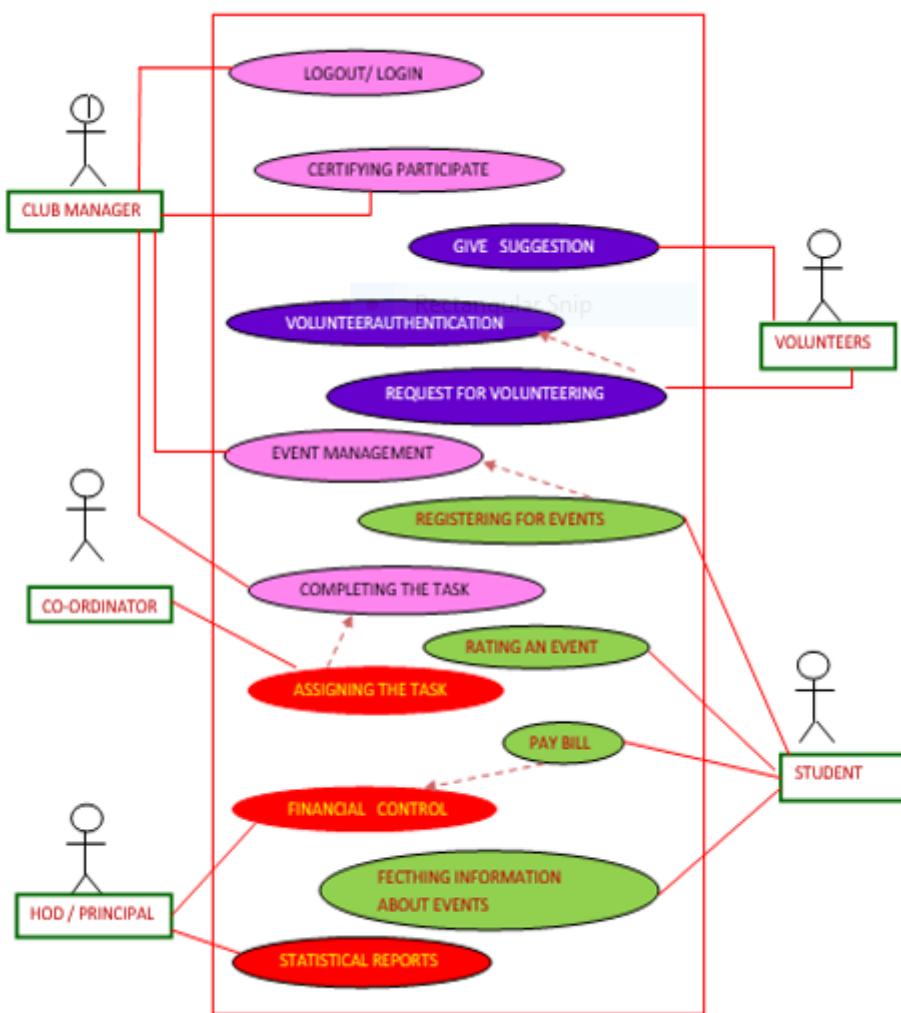


Figure 2.2: Use Case Diagram

Use Case	Log in
Primary Actor	Club manager, HOD, Principal, coordinator
Goal in context	Enable club manager access to the system through a tablet, android phone, laptop.
Preconditions	The club manager has a valid username and password and is not already logged in.
Trigger	The club manager requires access to the system to perform their jobs.
Scenario	<ol style="list-style-type: none"> 1.Club manager selects 'Log in' from the app menu. 2.The app prompts the user for their username and password. 3.The user enters their username and password 4.The tablet access to the system according to the access control.

Table2.3.2.1: Login use-case description

Table2.3.2.2: presents the Log Out use case description to show the interaction between a club manager, HOD, Principal, coordinator and an APP when logging out of the system.

Use Case	Log out
Primary Actor	Club manager, HOD, Principal, coordinator
Goal in context	Disable club manager to the system through a app.
Preconditions	The club manager is already logged in.
Trigger	The club manager no longer requires access to the system to perform their job.
Scenario	1. The club manager selects 'Log Out' from the App menu. 2. App disables access to the system

Table 2.3.2.2 Logout use-case description

Table2.3.2.3: presents the event management, request for volunteering, registering for events use case description

Use case	Accept registration.
Primary Actor	Club manager, coordinator, HOD, Principal.
Goal in Context	Accept registration that has been placed by the students.
Preconditions	Student has to request for membership or to participate in any events. Volunteers requesting for volunteering in events.
Trigger	The club members choose volunteers, participants for events
Scenario	1. The club manager who is involved in event management receives all members who have registered for the events. 2. Club manager certifies all the participants including volunteers after the events.
Expectation	If there are a greater number of volunteers then selecting volunteers who are members.

Table2.3.2.3: Event management use-case description

2.1.3.4 Flow Diagram

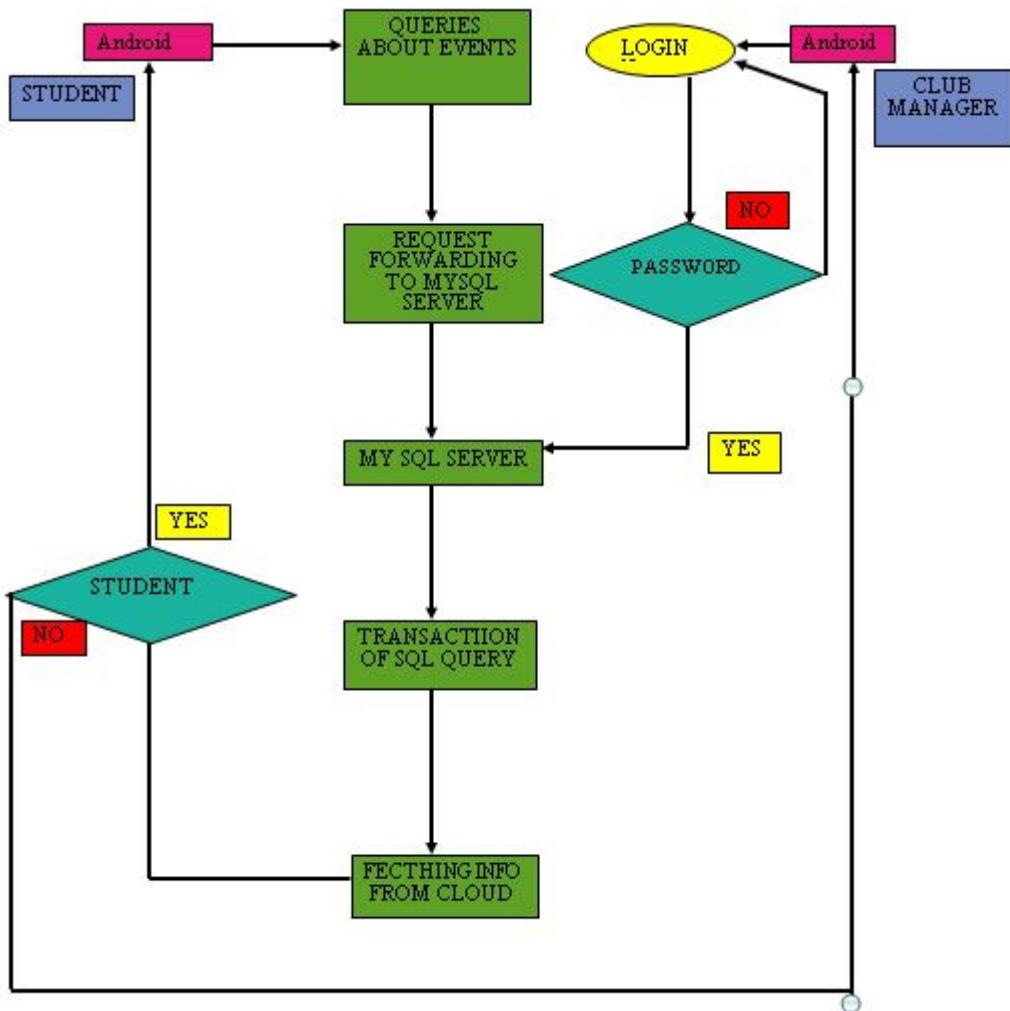


Fig2.3.3: Flow Diagram

Figure 2.3: Flow Diagram

2.1.4 Operating Environment

Any institution supporting various clubs in their institution can opt for the product. The product is designed keeping in mind the hierarchy of the authorities in a generic educational institution, but it can be adapted to suit the needs of various other kinds of institutions such as NGOs, Sports Club, Hospitals, Political Associations etc. The product aims to help manage the activities and tasks of any institutional environment where it is implemented.

The runtime environment can be modified to work with different devices such as smartphones, PCs etc. these different methods of accessing the product would provide flexibility and ease of access to the members of the institution. The product could also be modified based on the environment such as making it available offline or making it available only through a specific network. The product also has enough flexibility to be rugged in different environments and also so that it can be scaled to the requirements to different institutions. There are also no restrictions on the number of sub clubs that can be created by an institution, this is made possible because of the server-based approach which is taken in the design of the product. The product also provides a platform to encourage new clubs to be formed by individuals of common interests, this would allow new associations to pop up and new social networks to be created. This would also inculcate structural thinking and work ethic in the minds of the individuals who avail the services of the product

2.1.5 Design and Implementation Constraints

The systems from end is designed using Android studio using Java as the base language and the back end is supported on relational database system working upon MySQL servers, making the system very interactive and flexible. Currently the system is based on a centralized database but in case of expansion of the data for holding records going way back in the past and also facilitating a large number of Associations may call for a distributed database system approach. The system also facilitates deletion of records in a cascaded manner and will not support any partial deletion so as to maintain data consistency. For the purpose of deletion of records any particular member would wipe off all his/her details from the physical storage. So, data dropping would be done only at the end of a fixed period of time say every 10 years. The club's trying to avail the service should first satisfy a few structural constraints in order to be eligible for the service. The criteria are mentioned below:

- The association shall have a unique identification name and shall conduct various events having a general end public.

- The association would have coordinators (student) who'd manage all the different members a part of the association.
- The association shall have a Higher up authority for instance Faculty coordinators, head of Department, Principal who would be approving the different finance and event-based decisions.
- The association would also be open to outside volunteers for helping out with different events.
- The associations should be open to feedbacks, suggestions and have discussion forums.

2.2 External Interface Requirements

2.2.1 User interface:

Our product has an attractive and easy to understand user interface. The home page has options elevated based on most relevant contents which grabs user attention. It provides the user a logical path to follow for easy navigation. User can bookmark the events of interest which prevents him/her from missing a chance of witnessing it. Our software product is free from usage issues such as slow screen loading or crashes in between the usage. We also have a customer support to answer user queries within hours or even minutes in critical scenarios. Our product thus offers best possible UI experience.

2.2.2 Hardware interface:

Our product provides efficient support to smartphones, tablet and laptops. It is compatible with all kinds of operating systems including android OS and iOS. The product has been developed with enough resources to make sure it works great.

2.2.3 Software interface:

Our software product uses MYSQL database to store the club and user data and information. User input for the club querying information such as events organized by clubs, personal information is taken and the appropriate results are handed over. Presently implemented as an android app the software can be extended to a website as well.

2.2.4 Communication interface:

In our product, users are able to keep track of their activities through the email accounts. The communication among the software component would be performed through message passing over the IP network. From a technical point of view, TCP/IP will be used as the transport protocol, where each CDN server establishes a TCP connection to the network elements using a well-known port number. Messages can be sent bi-directionally between server and network elements. All messages consist of fixed length-headers containing the total data length and a request followed by a reply.

2.3 Requirement Engineering

2.3.1 Inception:

Our institution is full of a number of associations, both technical and non-technical who conduct a number of events, seminars, sessions and other festivities. The successful management of any of these association and club requires handling of a lot of tasks such as event management, tracking, publicizing, financial record handling, internal affairs management, volunteer and participation listing etc. Currently all these tasks are mostly done manually and assigned as a responsibility to one of the members. More often than not

this becomes a tedious job if done in that manner. This puts a lot of needless pressure on single individuals and constraints them to focusing on the documentation of financial and other information rather than putting their time and effort in the actual execution of the event. This reduces performance on the part of the club considerably. Further in this day and age having things on paper or having important responsibilities on the heads of individuals gives way to a lot of scope of human error, malpractices and other intrusions which effectively break the system.

One of our team members (Arunima Rashmi Giri), being closely associated with the Linux Campus Club, identified this issue as a common one among most Clubs of our institution. So, we decided to address this issue by coming up with a software which would facilitate managerial services to all associations willing to opt for it. The software would effectively take care of all financial recordings in a digitized manner as well. The system shall also provide the students with a platform to put forward their doubts, suggestions and views, in the form of discussion forums .We saw a lot of scope of automation for the management of data usually handled manually by the stakeholders and so decided to undertake the endeavor of developing a software that would reduce the burden of these associations so as to let them focus solely on providing better undertakings for the students and not on futile tasks such as manual documentation of their records.

We saw, thereby we could increase the internetworking of the various associations in the college providing a platform for students to put forward their suggestions and views for bettering the working of each club. This would keep the end users of the club constantly in the loop, aiding the club to develop itself on a feedback basis. We also believed that this would encourage healthy competition amongst themselves, as being able to monitor and compare their performances in respect to each other. The system would also help to avoid the clashing of the events between clubs or even the repetition of events by different clubs.

For the end users, that is the students the product would act as an interface to the different clubs the individual is a part of. It would allow him/her to monitor their indulgent ,participation and performance in each of the clubs ,provide a means to send feedbacks, suggestions ,participate in discussions ,look through FAQ's and keep track of the different events coming up on

his/her calendar .The Software would keep the students well informed about the plans of each club with the help of notification boards and also in close connection with the club by having direct means of communication with the club members.

The software's service would also extend to the college administration consisting of the Principal on the top, the head of departments and the faculty in charge. We thought of making the event approval mechanism completely digitalized with an OTP based approval interface. As for the financial the hardcopies would just serve as a backup, the software would generate automated reports about finance of the club giving information about exactly how much is spent when, where and for what with a receipt as proof. This would remove the burden of a single individual taking the responsibility of the financials and would remove scope of any malpractices.

Understanding the domain was our next step. The software is designed with the view point of providing students with a profile representation of their club activities, aiding the club members with finance, events, volunteer tracking and feedback, also providing the performance of the club (represented statistically) to the Principal, Head of Department and the concerned coordinators for performance review. Lastly giving the student coordinators of the clubs an in-hand overview and control of the activities going on in the club in form of an Android Application.

2.3.2 Elicitation

We approached the different stakeholders and interacted with them in order to gather all requisites that they desire in the software.

Topics on which questions were raised by our group to the stakeholders:

1. Problems faced by them in manual management of events, finance and volunteers.
2. Internal hierarchy and job assignment of different members.
3. The background effort going into putting an event together.
4. Publicizing of events.

5. Feedback managed.
6. Hierarchical approval of financial requests for the events.

The section below lists and elaborates all the meetings that were conducted with the different stakeholders for the purpose of collecting their perspective upon the product and their desired requirements. For the purpose of doing so we approached them with requests of meeting at an agreed location and also briefed them about the purpose of the meet.

Stakeholders: The Executive and Technical members of the Linux Campus Club 1. Sharath Hebbar 7th sem B who is a member of the executive committee of LCC. 2. Hemmanth Kumar 7th sem B who is the technical co-ordinator for LCC

Purpose: To find out in depth, the working mechanism that goes behind putting any event together.

Dated:27/09/2018

Venue: CS102

This is the transcript of the meeting conducted with the stakeholders for the Linux Campus Club (LCC). The objective of this meeting was to collect the requirements for the product from these stakeholders and to understand the current working of the club and their various events and activities.

1. We asked them how an event in their club come together. We gathered that the events are proposed by members and then authorized by the higher ups. Our product would then attempt to provide them with the facility to easily organize an event and assign members to the event.
2. We next found out how the events are advertised. It reached our attention that presently the events are promoted through social media (e.g. WhatsApp, Instagram) and also through personal interaction with the students and also recently through their webpage. We then proposed that we

could provide them with a message board where they could efficiently advertise their events.

3. Next, we asked them how they track of the students who have registered for an event and how they keep track of who have paid the amount for the event. They presently keep track of the students with a database which is generated with the help of google forms and they keep track of all the students who have paid the fees by keeping track of the fees receipts' which are issued to the students. We offered them an alternative where they could keep all the registered students details on our server and also keep track of the payment of fees by the storing digitized receipts issued to the students.

4. Next, we asked them how they assign tasks to individual members for the events. Presently they maintain a hard copy of the tasks assigned to each of the members and conduct demo sessions in order to prepare for the events. They also maintain meticulous reports as they have to submit the record of all the events at the end of each year. We proposed a simple interface through our application though which they could keep track of the tasks assigned to the members and also keep track of the demos' provided by the members.

5. Next, we were concerned about how the feedback was managed by the club. The club presently manages the feedback through google generated feedback forms, which are filled out by the participants. They also record the problems they faced in that year so that they may improve their events for the coming year. We proposed a 5-point rating system and suggestion facility so that the ratings and suggestions of all the students could be easily found in one place and also so that the club could determine how well the particular event worked out for that year.

Stakeholder: The Former teacher coordinator of Linux Campus Club (Dr. Manimala M)

Purpose: To find out the purpose of inception of the club, the former ways of working and its contrast with now and also the issues faced on the financial end.

Dated: 28/09/18

Venue:CS001

This is the transcript of the meeting conducted with former teacher co-ordinator of the Linux Campus Club, Dr. Manimala M. The objective of this meeting was to collect the perspective of the teacher co-ordinator of the club and understanding their responsibilities and what functionalities our service could provide them so that they could accomplish their club related activities with ease. We also talked about the goal of the club during its inception days and the issues faced by it. This meeting was arranged to know about the history of the management of the activities of the club, some of these activities are done differently by the club in the present-day scenario.

1. We asked the co-ordinator how many teachers were involved in the activities of the club. we came to know that were two separate teacher co-ordinators who handled the activities related to the finance and technical requirements respectively. We proposed a different for both of the co-ordinators each having their own special privileges.
2. We then gathered that one of the main responsibilities of the financial co-ordinator is the collection and allocation of the funds. In this process the students who are organizing an event approach the financial co-ordinator and then submit the request for the funds for that event in written form, then the form is approved by the financial co-ordinator and then passed on to the HOD and then to the principal for the final approval. We proposed an efficient way to digitize these request and approvals so that the entire process could be completed faster and with fewer errors.
3. We also learnt that the financial manager is also involved in planning the budget for the academic year. This activity is done before the start of the academic year. We proposed a functionality through which they could get a rough estimate of the budget for the academic year based on the data collected from the previous years with the help of machine learning principles.
4. We next asked the co-ordinator how they keep track of the funds spent for event throughout the academic year. We learnt that detailed reports

are given by the students for each event as to how the funds were spent. We proposed a way to form digitized reports so that the co-ordinator could access all the reports very effectively. We also pointed out that if the reports were digitized, we could easily extract the information required for the estimation of the budget from the reports generated for the previous years.

5. We next proposed service to develop graphs which would show the financial expenditure for the various events which are conducted throughout the year. We explained that this would enable the financial co-ordinator to easily keep track of the funds spent throughout the academic year and also observe the trend with respect to the expenditure funds by comparing the graphs of the previous years.

6. Finally, we learnt that the students who organize the events have to approach the technical co-ordinator for permission related to the procurement of technical resources such as lab access and other technical resources. We gathered that this too took place with submission of written requests for submission and the allocation of resources. We proposed a way to digitize this process so that the stakeholders could bypass the overhead of circulating written request forms.

Stakeholder: The Chief coordinator and the Chief Technical coordinator of Linux Campus Club 1.Amogha Subramanya D A (Chief coordinator) 2. Shashank R (Chief Technical coordinator) 3.Ajay B (Technical Member)

Purpose: The objective of the meet was to find the current method of working of the club from the coordinator's perspective and also to find all the responsibilities that are upon them and how their work could be eased out.

Dated:29/09/18

Venue:CS001

This is the transcript of the meeting conducted with the stakeholders Amogha Subramanya D A, the chief coordinator of the Linux Campus Club and Shashank R and Ajay B, who are the technical coordinators of the Linux Campus Club. This meeting was conducted to understand their various

requirements for managing the activities of their club and to discuss with them how our services could help them in managing those activities.

1. First, we asked them how they keep track of their financials, we gathered that they make regular reports detailing each transaction and they also keep all of the receipts pertaining to the purchases made on behalf of the club. We proposed a way for them to digitize their bills by storing only electronic copies of the receipts through our service, we pointed out to them that in this way they could optimize their financial management and avoid the hassle of storing the actual receipts for all of their transactions.

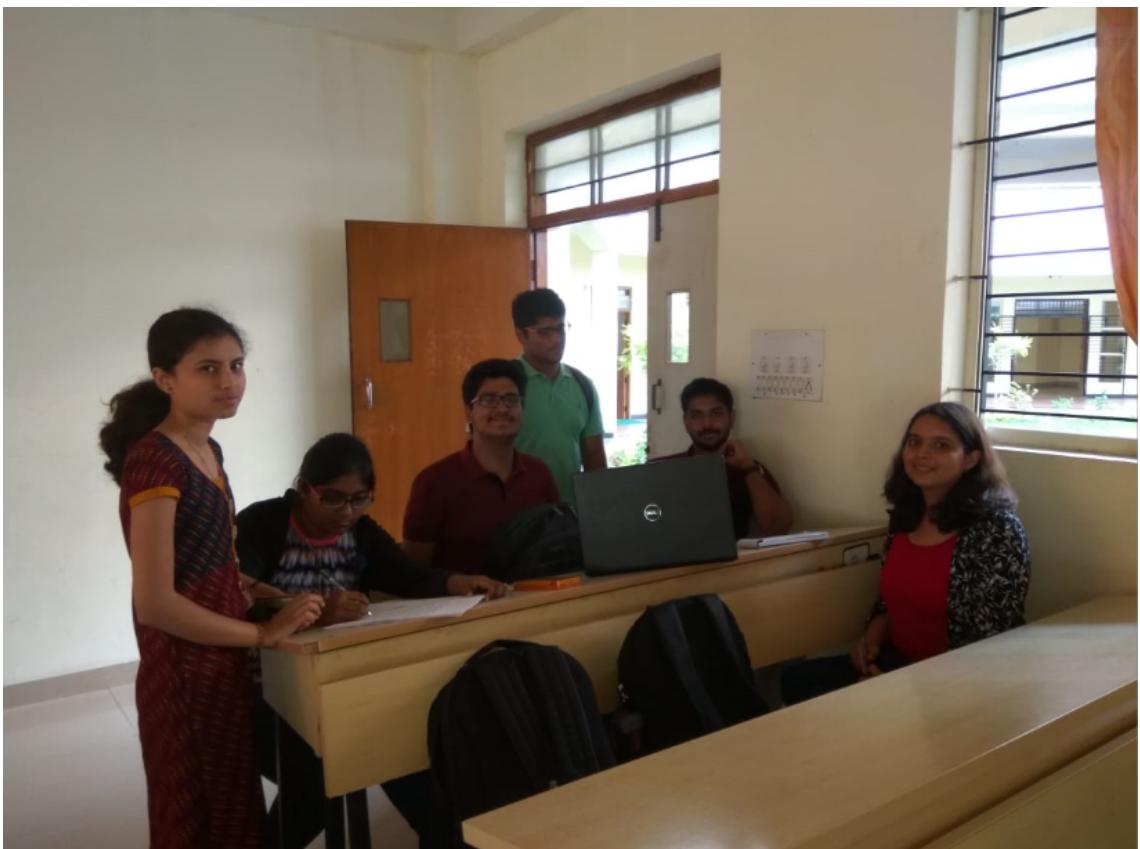
2. Next, we found out that they have given drafts and detailed reports to the respective teacher co-ordinator and also the HOD. We proposed to provide them with an effective platform through which they could easily provide the details of the events and their progress to their respective teacher co-ordinator and HOD.

3. We then asked them how they keep track of how they kept track of the various tasks assigned to the volunteers for the events. They conveyed to us that presently they keep a hard copy of all the tasks and that also they take demo sessions for the volunteers who are involved in teaching activities. We proposed a way for them to store all of this information digitally with the help of our service, so that the entire system would be more efficient and also so that the searching and reassignment of tasks to the volunteers could be done with ease. It would also help them to identify if a volunteer had been assigned which tasks previously so that they would not assign the same tasks to the same individual every time, which they had specified that they wanted to avoid.

4. They also wanted a method to keep track of their members to determine how efficiently they were performing. For this, we proposed to provide them with a service to keep track of various details of a member such as the amount of time spent by them in club related activities, the number of events they helped in managing and so on. We also proposed to provide them with graphs created with the help of machine learning techniques, so that they could pictorially track the performance of their members.

5. Next, we learnt that they have to get authorization from their respective teacher co-ordinator for each event and financial transaction, which they

presently do in a written manner as they require the signature of the co-ordinator to proceed further. We proposed a way for them to overcome this overhead by a service through the co-ordinator could grant authorization digitally with the help of OTPs and digital signatures.



Stakeholder: The Current teacher coordinator (Dr. Anilkumar M)

Purpose: The purpose of the meet was to find out the working of the club from the teach coordinator's perspective and also his/her responsibilities. We also intended to contrast the working mechanism from that of a couple of years ago and analyses the reason of these changes.

Dated:30/09/18

Venue: Faculty Staff Room

This is the transcript of the meeting conducted with the present teacher coordinator of the Linux Campus Club, Dr. Anilkumar M. This meeting was conducted in order to understand the requirements of the coordinator and the present responsibilities and activities of the teacher coordinator.

1. First, we gathered that one of the major problems was the huge amount of paperwork that had to be handled. The paperwork includes activities such as sending proposals, approval for events and finance, report generation and feedback. We proposed to come with a service to digitize all of these reports and activities so that there would be no need for the tremendous amount of paperwork which is presently generated.
2. Next, we learnt from the co-ordinator that there was a problem in keeping track of the events with respect to when and where they are scheduled. We realized that this problem might also be faced by the volunteers and other members of the club. In order to overcome this problem, we proposed to develop a calendar like interface complete with the details about the venue and timings of the various events. We also proposed to provide a reminder feature for upcoming events.
3. We then found out that they needed an effective way of keeping track their volunteers and members. An activity diagram was proposed for this purpose wherein the various activities of the members could be kept track of and analysed easily and effectively. We also proposed an online review system where the sessions for the various events and members who were involved in them could be rated based on the feedback given by the students participating in those events.
4. We then gathered that much of the technical resources and tools are not effectively provided to the students and this in turn leads to a lot time being spent on menial tasks such as installation of programs, configuring systems and so on. We proposed to provide a platform through which the information regarding the source for the technical resources and detailed documentation related to the proper configuration of these resources could be provided to the students, so that they save a lot of time which is being wasted in the current scenario.
5. Finally, the co-ordinator expressed a need for a common forum for interaction between the various personnel and students involved in events and by extension the club as a whole. We proposed to provide a technical forum platform service where discussions related to the events could be held and

publications and technical papers which would contribute to the discussion could also be posted.

Stakeholder: Senior Editor of SJCE Editorial Board. Poornima Vijayan Pisharody, 5th semester, Electronics and Instrumentation

Purpose: To understand the internal working Editorial Board, a non-technical club of our University.

Dated: 28/09/2018

Venue: SJCE girls' hostel

This is the transcript of the meeting conducted with the stakeholders for the SJCE Editorial Board. The objective of this meeting was to collect the requirements for the product from the stakeholders and to understand the problems faced in manual management of their various events and activities.

1. We asked them how an event in their club come together. We gathered that the events are proposed by members and then authorized by the higher ups. Our product would then attempt to provide them with the facility to easily organize an event and assign members to the event.
2. We next found out how the events are advertised. Presently they are using social media (Facebook, WhatsApp, Instagram) as a tool for advertisements. They also make announcements in the classrooms. The stakeholder mentioned the problem of reaching out to the participants regarding change of venues and dates of the events. We then proposed that our product could provide them a platform which enables the event organizers to update last minute changes and make sure that it reaches their intended participants.
3. We asked the stakeholder about the management of volunteers for the events. We found out that the club uses google forms for volunteers to register and they conduct volunteer meets in order to divide the tasks based on their interest. We proposed that our product could provide an interface which divides the volunteers by analyzing their area of interest mentioned while registering and thus could lessen the burden on the part of club.
4. When asked about the problems in execution of events, stakeholder stated the problem regarding clash of resources required for events such as mikes and speakers with the other clubs who are hosting some events concurrently. We proposed the idea of giving an account of resources engaged by different

clubs which help them to plan their tasks accordingly.

5.Finally we asked them about management of feedbacks. Presently they are using google forms for collecting the feedback which gets evaluated by staff in charge, principle and the members of the club. We proposed a 5-point rating system and suggestion facility so that the ratings and suggestions of all the students could be easily found in one place and also so that the club could determine how well the particular event worked out for that year.



Stakeholders: IEEE members Rakshitha k (WIE secretory), Swathi S, EDS JOINT SECREATORY.

Purpose: To understand the internal working and requirements of IEEE, SJCE.

Dated:01/10/18

Venue: Kaveri PG

This is the transcript of the meeting conducted with the present members of the IEEE SJCE. This meeting was conducted in order to understand the requirements of the coordinator and the present responsibilities and activities of the different members of IEEE.

1. We first went through a couple of tasks that are undertaken by the club. They are as follows:

- Teaching juniors, humanitarian activities, weekly visits for government school, Divya Deepa trust.
- Divya Deepa trust is a zero fees residential school and students like drop outs, who can't offer even government fees join this school. Students from this school can directly write 10th board exam.
- IEEE members would teach them English grammar, basic science concepts through experiments.

2. The requirements mentioned are:

- She wants us to provide service of assisting her in announcements. Like she needs which classes are going on, which classes they have announced. Even she needs remainder to inform them that which classes they have not yet announced. Basically, she is finding difficulties in announcing about events.

- Online registration gives IEEE members about the student who has registered, but intern students won't receive any kind of acknowledgement. So, she asked that whether it is possible to send acknowledgement to students.
- When students register for events, confirmation message is sent to their phones rather than giving receipt.
- Means to store names of students, component list, expenditure.
- Sending information about meeting to be held to all members involved in conducting meeting, upcoming events, .etc.
- To download Journal documents.
- IEEE is an independent student branch. Counsellor like branch counsellor, ED's counsellor, WIE counsellor.
- Registration fees would be sent to US.
- Workshop member's estimation is a tedious task since they need to bring kits for the workshop beforehand.
- They will conduct 4 workshops
 Soldering and etching
 VPM
 Robotics
 Snap circuit
- Informing students about Pre-requisites required for attending events.



Stakeholders: CSI member (Apoorva Agarwal)

Purpose: To understand the working and requirements desired by Computer Society of India,SJCE.

Dated:01/10/18

Venue: Kavery PG

She shared problems she faced while conducting events.

- Availability of seminar halls.
- Scheduling of events.
- Finding a person to sponsor for events
- Since events are conducted on the same topic by different clubs members attending to the particular event will be less.
- She all mentioned that events are also conducted by the students and they don't form any clubs.
- She mentioned about clubs like pro-mechanical, Aero SJCE, photograph, gravity, which are not well known by the students and events conducted by these clubs are not known to most of the students.

2.3.3 Elaboration

A detailed explanation for the information obtained from the stakeholders is presented by elaboration phase. The requirements specified by one stakeholder may be repeated by another stakeholder. When the requirements collected are analyzed all together, a kind of pattern may be recognized.

The pattern may be elaborated to facilitate the formation of complete and consistent requirement document. Problems in realizing the product are inevitable. The pattern allows easy identification of such problems and through the reuse of defined and tested solutions to the known problems it guides us throughout the project. Also, it is very important to understand the problem statement completely before attempting to design the solution. The scope of usage, technical risks involved and most essential features that are to be integrated into the system are recognized and an architectural plan for the project called architectural prototype is developed. Elaboration enables organized study of the system to help accommodating changes in user demands from time to time. The prototype is carefully evaluated for stability and adaptability to change and also a validity test is performed to check whether it adheres to the set vision without compromising time and cost of development. This facilitates clear understanding of critical use cases in which the product must prove itself. Elaboration also helps in getting a better picture of development environment and the way in which the tools and software components fit in. With a strategic plan considering component costs and possible reuses, elaboration also acts as a preface for construction. Any bugs identified in overall elaboration activities are resolved to convince the customers that we have a realistic approach to the final product. The outcome of elaboration phase is the use case model.

Use case model is a model of system's intended functions and its environment, and serves as a contract between stakeholders and developers. It consists of a collection of:

Use cases: Sequence of actions performed by the system which offers observable and valuable result to a particular actor.

Actor: Defines a coherent set of roles that users of the system can play when interacting with it. A user can either be an individual or an external system.

Supplementary specifications: Capture the system requirements that are not readily identified in use cases.

Analysis model: It is an object model describing realization of use cases, which serves as an abstraction of Design model. It contains results of use case analysis, analysis classes and associated documents.

Analysis classes: Identifies major responsibilities which are to be handled and represents prototypical classes of the system. It is the 'first-pass' at the major abstraction that the system must handle.

Design model: It is an object model describing the realization of use cases, and serves as an abstraction of implementation model and its source code. It is used as an essential input to activities in implementation and test.

Design classes: It is a description of a set of objects that share the same responsibilities, relationships, operations, attributes and semantics. They are

derived from Analysis classes as a result of design process.

2.3.4 Negotiation

The clients of our software product are the members of the clubs within the University. Each of the club has its own identity and the clubs are also diverse in their functionalities. Accordingly, when they approached us with a requirement of automatize assistance for the management activities of their club, each of them had different visualization of the end product they expected.

The stakeholders cannot completely state their all requirements explicitly. When the requirements collected from all stakeholders are collaborated, certain requirements appeared to be conflicting, such as priority of tasks or the areas with problem which they thought are troubling them the most. It was highly important for a negotiation strategy to be applied to resolve the conflicting requirements. Because we have many live examples of failed projects due to poorly negotiated requirements among the stakeholders. Negotiation leads to benefits such as understanding the problem statement, implicit requirements, constraints applicable along with fostering team learning, resolving ambiguity and complexity, dealing with uncertainty and finding better solutions. But for utilizing the benefits of negotiation to fullest, the process implementing the correct negotiation is necessary. A collaborated discussion regarding functions, features and priorities and delivery dates with all the stakeholders can set a common goal for the product outcome and also resolve the disagreement between the stakeholders. The set of requirements collected from the discussion needs to be properly analyzed for the final template to be prepared with rich set of functionalities which accomplishes the desires of all the stakeholders. When the final software product is delivered to the stakeholders, a smile of satisfaction on their faces is expected which in turn will be a fruitful result of our hard work.

2.3.5 Specification

2.3.5.1 Functional Requirements

Normal Requirements:

The following section mentions all the requirements desired by the different stakeholders.

1. Event Management: The product shall facilitate a mechanism to plan and handle events, decide the dates and venue, tasks assigned to each of the members kin an organized manner and also record the success of the event after completion.

2. Publicity: The product shall also provide a platform and tools for the club to publicize their events, sessions and other festivities with the help of digital posters on the digital notice board and also as notifications sent to all the registered students.

3. Financial Management: The product would aim to provide a reliable aid for the management of financial aspects of the club including total amount sanctioned, total expenditures made on the events and generation of the suitable reports and bills for the same.

4. Resource Management: Provision for the clubs to get an account of resources engaged such as classrooms, mikes, speakers and also the dates of events which are planned by other clubs to avoid any kind of clashes and mismanagement.

5. Notifications and Broadcasting: Provision for the members to make any last-minute updating regarding the events and notifying their intended participants.

6. Volunteer Management: The product shall also aid the clubs in keeping track of the involvement and performance of all its volunteers. Also make the registration and certification process easier with the help of automated listing.

7. Feedback: The product would provide simple techniques for evaluation of feedbacks and ratings received by the participants.

Expected Requirements

The below section includes all the features that are to be included in the product which are not necessarily mentioned explicitly by the stakeholder but are implied either way.

1. Provision for creating a new club.
2. Provision to add new members to the club.
3. Provision to host an event all year round.
4. Provision for painless and secure management of financial issues of the clubs such as amount deposits, withdrawals and generation of total expenditure results for different events.
5. Provision for effective advertisements to publicize events as well as the clubs.
6. Provision for the participants to register for an event.
7. Provision for the volunteers to register for an event.
8. Provision for each user to keep track of his/her involvement in different clubs and events.
9. Provision to record the number of certificates earned by a student.
10. Provision for participants to enter feedback for the events.

Exciting Requirements

1. Every functioning club is monitored by a staff in charge. Activities of the club are also supervised by higher authorities like Head of the Department and Principle. Accordingly, each and every event planned by the clubs must be approved by the above-mentioned higher ups. The method which is currently being used by the clubs is to get a signature of all the supervisors for a planned event so as to obtain their consent. Our product could ease this task through automation. Once the event is finalized by the club members can send a link to the corresponding staff in charge notifying the request for approval. The link when opened by him/her would generate an OTP. If this OTP is entered by the staff in charge within a timestamp will then redirect the link to HOD. The same procedure repeats again and finally when the principle gives his consent by entering the OTP, members of the club will receive a notification that the respective event has been approved and they may go ahead with it.
2. Successful event is a combined effect of efforts of members of the club as well as the work of enthusiastic volunteers. Presently, clubs employ the volunteers by allowing them to register through Google forms. But in majority of cases the proportionality of numbers of volunteers registering and those who actually make it to the event does not match. Involvement of volunteers is manually recorded. This manual recording itself is used as a

decision parameter while qualifying him/her as a future member of the club. But this parameter cannot be trusted completely, as manual observations may have some defaults. Our product could eliminate this problem through automation. Participation of each and every volunteer is kept in track which is evaluated by the members of the club at the end of each event. This information can be brought into a metadata format which is analyzed and interpreted using machine learning concepts. We aim to get a graphical representation of the individual's performance throughout his/her journey with the club.

2.3.5.2 Non-Functional Requirements

The clients came to us with a requirement of automatize assistance in effective management of various activities of the clubs. We have thoroughly analyzed the requirements in order to accomplish the user demands to their utmost satisfaction.

Performance

Our aim is to ensure event success with real-time insights to spot and address problem areas in-the-moment rather than after the fact. This software product is an easy to use tool that is capable of hosting multiple events simultaneously. The product shows all the events organized by the clubs and enables an individual to register himself to take part in events of interest. System shall not collapse with increasing number of clubs/members of the clubs/ participants. Changes and updates shall be made easily within moments. System shall provide the great and attractive user interface which will enrich the user experience.

Security

One of the major functionalities provided by our software product is the management of financial aspects of the clubs. The product shall protect the confidential information pertaining to finance. Also, the product shall preserve the personal details of the members and participants such as phone

numbers, email ids etc. by revealing any such information to only those who are authorized for access. One of the exciting requirements also states the introduction of automation in approvals of the events by higher authorities through generation of OTPs. In this scenario, we shall work hard to achieve critical security requirements.

Quality

Rich set of facilities available with our software provides wonderful experience to the end users. Our product is a simple and elegant tool to keep every detail of event structured and organized. Provisions such as ease of use, customer support, compatibility with multiple events and efficiency in performance are made available.

Business Challenges

- Manual management of club activities is a tedious job. There is a need for stable, maintainable, and reliable interface which minimizes the burden on shoulders of club members. Noticing this, we have developed the software from scratch to accomplish the requirements.
- Mission is to create user friendly and multi-functional software product with high reliability.
- Keeping pace with elevating user demands to make our product up to date to the current trends.

2.3.6 Validation

Requirements should be validated before the software product as a whole is ready. The validation step is basically a review of requirements collected, to

make sure that they meet needs of stakeholders and to identify any problems with the requirement specifications. It is preferable to fix any bugs in the requirements in the initial stages rather than dealing with their complex consequences in the final product because changes in the requirements usually demand changing and re-testing of design and implementations. Any mismatch in the requirements interpreted and those which are desired by the stakeholders may result in an incorrect product to be developed.

Our software product has stakeholders from different clubs of our University. We conducted meetings with all of them personally and listened to their requirements meticulously. We as a team, sat together and carefully analyzed overall bundle of requirements and had a constructive discussion to confirm that we have understood all the aspects appropriately, thus finishing internal validation phase. We carried out external validation by having another meeting with our stakeholders to clarify our doubts and to obtain their consent for the final requirement document that we had created. As our software product has a scope of development throughout its lifetime, we consider the external validation phase to be a continuous process. Most of the requirements passed the validation phase, yet we noticed some problem areas like:

1. We had planned to provide an automatize assistance with an OTP mechanism in order to eliminate the burden of taking the approval signatures for each and every event. Though it could ease the work on the part of club members, higher authorities of the University demand the club members to personally contact them and receive the consent and also a physical record copy is desired for the future reference.
2. We had planned to provide a graphical representation of performance of an individual, so that our software could assist the club to choose its future members by recognizing the active participants with peak performance statistics. But the discussion with one of the stakeholders made us realize that it will be difficult to measure performance of an individual without having a complete understanding of his/her capabilities with respect to each and every field. Also, it demands considerable amount of software expertise to make sure the result of assessments is accurate.



2.3.7 Requirement Management

The product is so designed that it can be adapted to the requirements of different institutions. One of the requirements that is addressed is if they club administrators have no need for the history of the registry of their members such as when the members are no longer associated with the institution, then the product would delete all the member related information without affecting the actual vital functioning of the club. The other requirement that is addressed is when a particular club has multiple administrators, in this case the product can create as many administrators as required and grant them all privileged access so that they can make changes to any details of the club as they see fit. Some of the clubs may also have a hierarchy of different sub-clubs and associations (e.g. IEEE), in such cases the product may have to be modified so that it can accommodate the different sub-clubs and also provide an effective way for communication between them and also to manage joint event conducted by the sub-clubs. This may also require that the members of the main club be grouped into the different sub-clubs, this may lead to grouping conflicts as a member may be involved in two different sub-clubs, the product needs to give unambiguous parameters to handle such situations.

The institutions may require that different members with different levels of authorization should be allowed to view only the information that they have access to, this can be accommodated with the help of passwords and other means of authentication. This restricted could also serve to restrict the ability to modify sensitive information to only those who have the authority to do so. The clubs may require that some information such as that regarding to financial transactions must be encrypted so as to provide a proper level of security. The product can also be modified to allow for private messaging between the members of a club, this may also need end to end encryption to make sure that the conversations remain private. Some clubs may not need all of the features provided by the product, in such cases the product is flexible enough to have the option of turning off those features so that the members can see only the functions in which they are interested. The product could also be linked with email accounts so that the club can send push notifications to their members, volunteers and participants about upcoming events. There can also be a connectivity to third party social media applications such as WhatsApp, Instagram etc. so that sharing and advertising of information related to events can be with ease across multiple platforms. In these and many other ways the product may be modified as and when new requirements are given by the stakeholders. This shows that the product is rugged and flexible enough to handle changes in requirements even late in its life.

Chapter 3

System design

This document details the approach taken in the design of the product from the different viewpoints of abstraction such as architectural, data/class, component and interface level. Here we look into the specifics of the architectural design class and explore the different components associated with the product. The architectural diagrams are explored so as to understand the working of the entire system as a whole. The document also throws light upon the interface components and inter-working. The ER-diagram and schema diagram are also provided to explain the internal storage structure. The components and their interactions with each other are stated explicitly to understand the level of cohesion and coupling between the components. The concepts of modularity, information hiding, functional independence, refinement and refactoring are applied throughout the design of the product.

3.1 Component Level Design

This section specifies the different components associated with the software product. It also aims to describe the basic details of each of the components and attempts to show the interaction among the components. The components are described with a common template, so that the reader can explore only the components that they require. This would also provide for easy coding during the actual implementation of the design. The Component Level Design is at the highest level of abstraction and information hiding and functional independence are exercised wherever possible. Through the different components we explore all of the different functionalities of the product at a very high level of abstraction, giving implementation details only where required. These broad aspects of the components are further refined in the coming sections where details pertaining to those sections are explored.

3.1.1 Component - Member Registration

- Actors**

The actors who are involved in this component are the students who want to register as members for a particular club. They are involved in registering their credentials such as their USN, Name, Semester, Year etc.

- Fuctions**

The main fuction of this component is to record the details of all the students who register for membership of a particular club. The fuctions may also include associating a password with a particular member to provide for a certain level of security, sending a registration link to the email given by the students for authentication purposes.

- Data Stored**

The database has been configured to store all of the details of the student in an efficient manner including multivaled attributes for phone numbers and email addresses.

- **Interface Required**

This component requires a very simple interface consisting of a sign-up sheet with all of the fields required to obtain the details of the students who are registering for membership, it may also involve an authentication page where the students are redirected to in order for them authenticate their account with their registered email accounts.

- **Interaction with other components**

This particular component requires no communication with any of the other components for it to function correctly. In this manner this component exhibits a high degree of cohesion, since it has the one task of registering the students for membership, and also a very low degree of coupling as it requires no other input from any of the other components.

3.1.2 Component - Event Planing

- **Actors**

The actors who are involved here are the members of the clubs which is hosting the event. The other actors who may be the teacher co-ordinators who monitor the entire club activities and hence are required to oversee the planning of all the events hosted by the club. The input of the HOD of the department of which the club is part may also be needed during the planning of the event.

- **Functions**

The functions of this component involve allowing the members to plan an event, this includes the storing of all the event related details such as the title of the event, date, time, venue etc. this also includes giving a brief description for the event and specifying all of the particulars of the event. Here individual members may be assigned tasks which they are responsible for, this provides an efficient way for the members to keep track of who is doing what. These functions allow for efficient planning of an event in such way that no task however small is overlooked and the component also allows for all the preparations needed for a particular event to be completed head of time by providing deadlines to the members relative to their respective tasks. There is also a provision for each member to take notes or for form to-do lists related to a particular task for a particular event which would allow them to

keep track of all the things that have to do in-order to fulfill their part of the responsibilities for an event.

- **Data Stored**

Here the database is structured to provide for storing all of the details of an event such as the title, date, time, venue, brief description of the event etc. The database also records the members of the club who are planning that particular event by establishing relationships between them. The database stores each event associated with its particular club. The events can also be stored as being part of a fest of the club. The data is stored in a very structured manner such that the extraction of the details of an event are very easy for those accessing it and also it ensures that none of the information pertinent to the success of the event is lost or corrupted in any way.

- **Interface Required**

The interface required here must accommodate all of the members who are involved in planning the event and must provide for them to coherently modify the information related to the event. The interface should also give members the ability to assign specific tasks to particular members of the club. The product aims to provide a calendar interface, which would give the members a common space where they could see all the events ongoing and upcoming. The members who are planning the events are given a provision to take notes and form a to-do list so as to make their tasks easier.

- **Interaction with other components**

This component requires interaction with the member registration component as it needs the details of the members who are planning the event. Thus the component can derive all of the information it needs about the members who are planning a specific event by requesting information from the member registration component.

3.1.3 Component - Financial Management

- **Actors**

Here the actors are the members of a club who perform transactions on behalf of an event hosted by a club, i.e. the members who make purchases for the events that they have planned. The actors may also

be considered as the events for which the purchases are made. In reality the actors are a combination of the members and the events.

- **Functions**

The functions here mainly consist of keeping track of how the funds allocated to the club are being spent. The component keeps track of the purchases made by maintaining digitized copies of the receipts for the different purchases. This allows for the easy tracking of where the money is being spent and for what purposes. The component also provides the additional feature of planning out the budget for the entire academic year based on the spending of funds in the previous year. There is also the provision for the component to plot graphs on the spending throughout the year based on the data collected for review purposes.

- **Data Stored**

The database stores the receipts of every transaction along with the member who made the transaction and the reason for it. The annual budget can be derived from the data collected as can be the graphs for reporting the annual expenditure of the club.

- **Interface Required**

The interface here is in the form of a transaction form in which the members can enter the event for which they are making the purchase and also scan the necessary receipts. The scanned receipts are automatically timestamped to correspond to the actual date and time where the purchase was made.

- **Interaction with other components**

This component has to interact with the member registration and event planning components in order to function correctly. It derives the member information from the member registration component and the event information from the event planning component. Hence this component requires input from the two other components to function fully.

3.1.4 Component - Student Tracking

- **Actors**

The actors involved here are the students who register for a particular

event either as participants or volunteers. The actors may also include the club administrators who wish to see the details of the students who are participating in that event.

- **Fuctions**

The component keeps track of all the students who register either as participants or volunteers for a particular event. It also keeps track of the number certificates acquired by each student. This would allow the club administrators to derive statistics about each student individually whenever required. This may also help in choosing the successors to the positions of authority within the club.

- **Data Stored**

The database here stores all of the students details such as Name, Semester, Department etc. Along with the details of the students, the event for which they have registered are also recorded.

- **Interface Required**

Here the interface needed is simply for the students to see all the upcoming events being hosted by the various clubs and then register for the events as either participants or volunteers. The students are also provided with a calender interface to keep track of all the events for which they have registered. The students are given with the provision to know which certificates they have earned and for which events.

- **Interaction with other components**

This component interacts with the event planning component. It may also indirectly communicate with the member registration component when members need to find statistics of a particular student. Thus the component extracts details about the event such as title, date, time etc from the event planning component and provides information about the students to the member registration component. Thus the component needs input from one component and gives output to the another, this allows the component to function fully and efficiently.

3.1.5 Component Interactions

The diagram shows the interactions among the components. The member registration component does not require any inputs, the event planning components takes input from member registration, the financial management

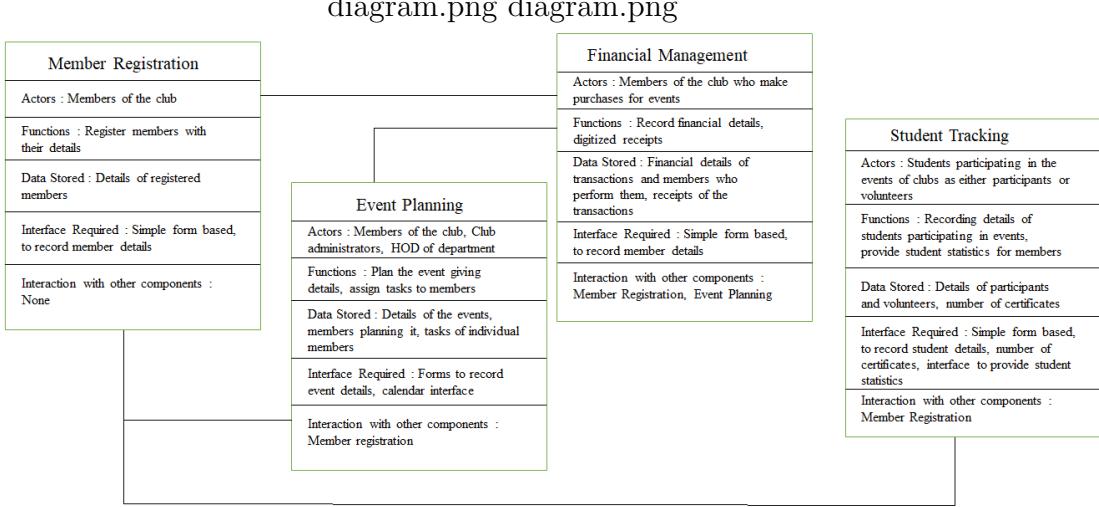


Figure 3.1: Presents 'Membership selection' Activity diagram

component takes input from the member registration and event planning components and the student tracking component takes input from member registration. In this way there is interaction between the various components. The concepts of cohesion and coupling are followed as far as possible.

3.2 Architecture Design

The following section describes Architectural Design of our software product. Being considered as the mother of all the design aspects of the software, it provides an overall view of the software product and represents the system from perspective of a related set of concerns. It acts as a reference model which guides readers in understanding how different hardware and the software parts of the system fit together and collaborate to realize the goals and objectives of the system. It lays out the high-level data and control flows, as well as the various interfaces between the system and the outside world.

3.2.1 Architectural Style

The software under consideration is based on multi tiered architecture, which is basically a client server architecture. The software can be divided into three different tiers:

- 1) Presentation tier:** All the end users such as students, volunteers, club managers, co-ordinators, HOD/Principal are provided with an easily understandable user interface. User can raise the queries according to their needs without any difficulty or complexity.
- 2) Logic tier:** The application logic and app engine constitutes the logic tier. App engine co-ordinates the application, processes commands and make logical decisions, evaluations and performs calculations . It acts as an interface between end user and server with data repository. It allows the complex structure and operations involved in the backend to be hidden from user. The application logic written in Java contains the functional business logic which drives the application's core capabilities.
- 3) Data tier:** Here information is stored and retrieved from a database or file system. The information is returned to the logic tier for processing, which eventually gets back to user.

3.2.2 Architectural Genre

This software product belongs to the combination of 'Communication' and 'Commercial and nonprofit' architectural genre. The main objective of the software is to accomplish the requirements of the various clubs functioning under the university. Its goal is to facilitate ease of operations for the clubs. It also provides various features for data transfers and management, methods to connect various end users of the data and at the same time handles the effective representation of the data.

3.2.3 Design Approach

We have employed top down approach of architectural design in our software product.

3.2.3.1 Context Diagram

The general idea of architecture, with high level of abstraction is represented by the context diagram which allows us (developers) and stakeholders to step back and look at the big picture. The block diagram presenting our system as an application running on a mobile phone is provided at the centre, surrounded by its users and other systems that it interfaces with.

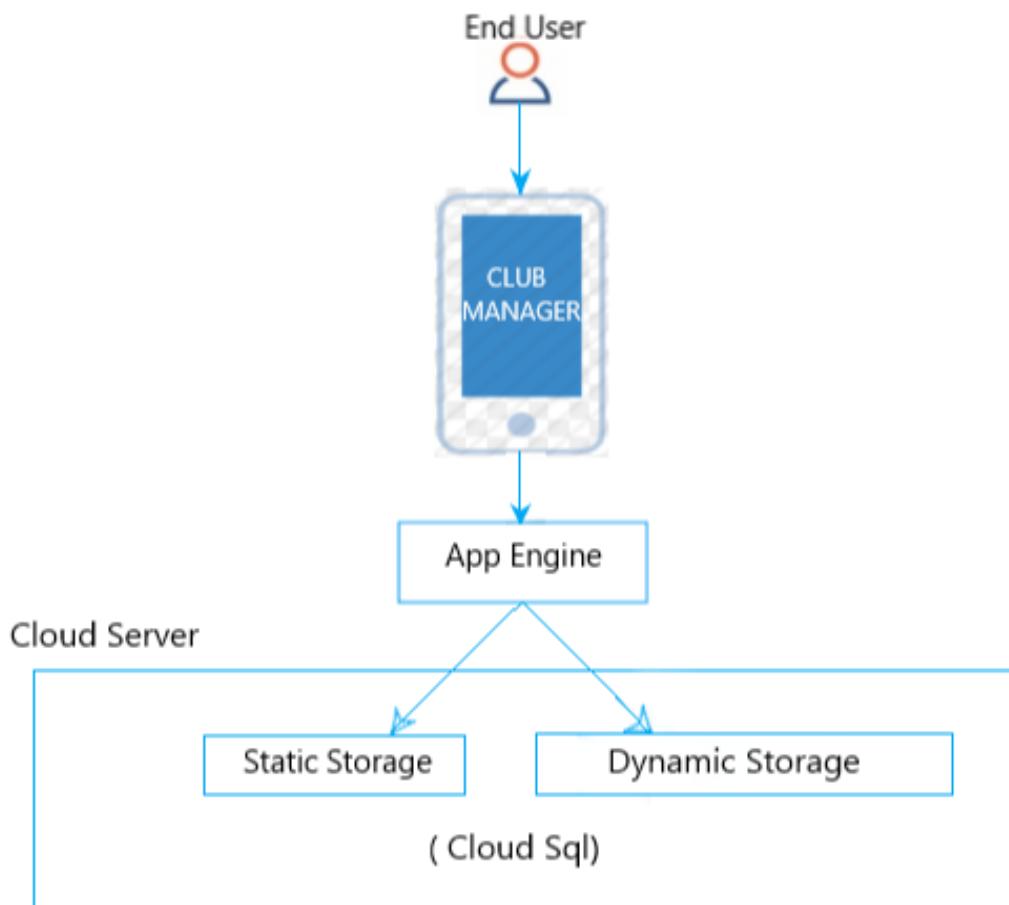


Figure 3.2: fig Context Diagram

- End User End users are the prime consumers of the services provided by software product. Each user has a different perspective of the system. Software caters to their needs and expectations efficiently. They are the key source for inputting the data required for operations of the software.
- Club Manager This is the communication interface between the end user and the system's core functionalities. Attractive and user friendly interface receives all the requests from users and maps those requests to app engine for processing. It has the responsibility of presenting the result of the query in a user understandable manner.

- App Engine App Engine is a processing unit of the software application .It processes user data, and makes it easy to deploy the application that runs reliably even under heavy load and with large amounts of data.
- Static Storage Application consists of numerous images, pictures, objects, vedios, unstructured data storages. For a fully reliable storage and management of above resources, google cloud storage is used.
- Dynamic Storage User requirements are instantaneous. Such requirements are handled well by having a data repository (Database) which provides the contents of interest to the users.
- Cloud Server Google cloud server which provides high levels of performance, security and control is used with an intention of delivering fast and efficient software product.

3.2.3.2 Container Diagram

With the knowledge established by the context diagram about the system's adaption to the real world environment, the following section aims to narrate the high-level technology choices with the help of a Containers diagram.

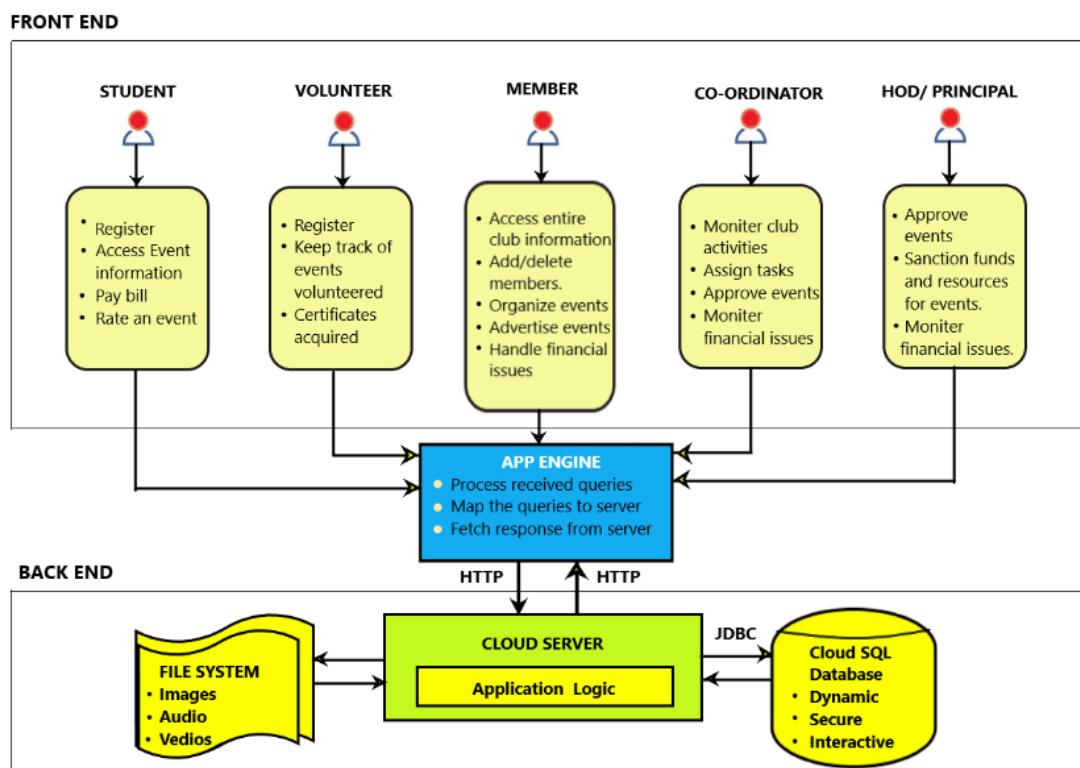


Figure 3.3: Container Diagram

- Users of the software product are students, volunteers, club members, co-ordinators, HOD/principal. Each user has a different set of requirements which he/she acquires by querying the application for the same. Users present their requests through the effective interface made available to them and also receive the adequate response.
- App Engine is an application platform that monitors, updates and scales hosting environment. It is made up of a single application resource that consists of one or more services. It is responsible for processing user

input. It allows the application to run within its own secure, reliable environment that is independent of hardware, operating system, or physical location of the server. Services under app engine handle different tasks such as API requests from mobile devices, internal and administration type requests, backend processing such as data analysis.

- The application has to be connected to the remote server to fetch various kinds of information. This connection is managed by HTTP Protocol, which is chosen because of its simple mechanism to transport the information. The HTTP connection is managed by the Android HTTP library.
- Cloud server is flexible and offers great scalability. Resources can be scaled up or scaled down accordingly. When there is more demand placed on the server, capacity can be automatically increased to meet the demand. The software is highly reliable as it makes use of cloud server for its functionality. Features of the software are accessible from anywhere, and at any time as long as internet access is provided.
- The software has its logic written in programming language Java. Overall functionalities of the software depends on application logic. It contains the code required for the software to accept the user input, perform local calculations to understand the service required, logic to fetch the data from database and to perform any other operation for complete implementation of all the functions and features of the software.
- All the application data, including user data is stored in MySQL relational database. Our software makes use of Cloud SQL which makes it easy to set up, manage and administer the relational database MySQL on google platform. The database is connected to the server using JDBC connection.

3.3 Data/Class Design

The following section talks about the design model from the view point of the flow and control of data and the different identified classes and entities and also the relationship that hold good between them. It also aims to describe the type of participation and the cardinality that exists between these classes. For representing these features we make use of an Entity-relation Diagram and also a data-schema diagram. Further collaboration diagrams can be employed to depict the proper flow of control between these entities. It must be noted that the data-class design possesses relatively less level of abstraction as it does give an idea of the physical structure or the data placement of the design model. Data-class model helps convert design requirements into a representational class description.

Data modeling is the process of documenting a complex software system design as an easily understood diagram, using text and symbols to represent the way data needs to flow. The diagram can be used as a blueprint for the construction of new software or for re-engineering a legacy application.

3.3.1 Entity Relation Description

The above diagram throws light upon entity relation that exists between the identified entities. The square figures represent entities where as the diamond figures depict relationships between these entities. The oval figures represents the attributes that are bundled with each of these identified entities.

The attributes which are underlined represent the primary key i.e the attribute that can uniquely identify each tuple of that particular relation(entity). The arrows are used to show the connection between the entities. A crossed arrow ends signify total participation on the part of the other entity where as a three headed end shows an M:N cardinality type of relationship.

3.3.2 The Identified Entities:

The subsection below lists and describes the identified entities of the design model.

1. Member

This entity member contains all details of all the registered members(both active and inactive) for every club along with all their required personal details.

2. Event

This entity contains all the information of an event that is to be managed.The details are such as the venue, date, time, name and perhaps the description.The rating of the event is also calculated after feedback collection.

3. Student

The entity student contains all the required details of a student acting as either a participant or a volunteer in one of the events held by one of the clubs.Records such as number of certificates,personal info,contact info are saved.

4. Finance

This entity contains all the vital financial information about any of the events being organized.It also contains the receipt of the transactions in an image format.

3.3.3 The identified Relationships:

1. Organizes

This relation contains information of the members and the respective events they are organizing with an additional attribute to hold the task assigned to each member.

2. Notes

This relation holds the respective to-do notes that any member may want to make for any particular event.

3. Participation

This relation maps the students with the respective events they are involved in either as a participant or a volunteer.

4. Transaction

Transaction is a relationship joining three entities namely Member,Event and Finance.It maps every transaction to the respective member who did it and to the event it was done for.

3.3.4 Schematic Description

Schema diagrams provide a high-level view as well as a detailed view of the database design. On a high level, they show the SQL developer which tables and columns are present, as well as their proper spelling. With regard to relationships, they illustrate the requirements for joining tables. For example, if as developer we need to find out which events are most popular with students, we can look at the student table at the bottom right, and see that the participation, event and student tables must be joined in order to answer that question.

The below diagram shows the schematic view of the design model.

In the above diagram the abbreviation FR signifies foreign key and all the attributes in bold signify the primary attributes.

3.4 Interface Design

User interface design (UI) or user interface engineering is the design of user interfaces for machines and software, such as computers, home appliances, mobile devices and other electronic devices, with the focus on maximizing usability and the user experience. The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals

3.4.1 'Log in' activity diagram

- To achieve information hiding, HOD, principal, member of the club are provided with user name and password.

- Students should not be able to access internal information of the club.
- Either user name or password is wrong, then control is transferred to log in page.
- In case HOD, principal, member have forgotten their password then they can change their password.

3.4.2 'Log out' activity diagram

- When user wants to log out he/she should select 'log out' button.
- If user had changed any data and not saved then displays message to confirm logout.
- He/she can either confirm or cancel the log out process.

3.4.3 'Registration' activity diagram

- Event registration is the process by which an event planner organizes attendance for the participants.
- The event registration process collects detailed information about the attendee, processes payments, scans and validates registration or tickets at the door, communicates with attendees automatically and tracks attendance.
- To assist event registration process student has to fill the form then click 'submit' button.
- This information is stored in the database.
- Displays information about the club. 'OK' button is selected.
- Confirmation message is sent.

3.4.4 'Volunteers Registration' activity diagram

- Students should fill form to provide information about them.
- If club selects that student as volunteer then task is assigned with respect to the event in which the student requested for volunteering.
- If there is no space for volunteering then club will not select that student.

3.4.5 'Membership selection' activity diagram

- Students can become member of any club through on-line registration.
- Students should select club name of their interest from the list of clubs.
- Information about membership fees is display on the screen.
- If student is OK with the amount then venue is displayed where student has to come and pay the amount.

3.4.6 'Rating of an event' activity diagram

- Students can give their opinion about the events.
- They can rate the events between 0 and 5.
- Average of all Rates assigned by the students is taken and result is stored.

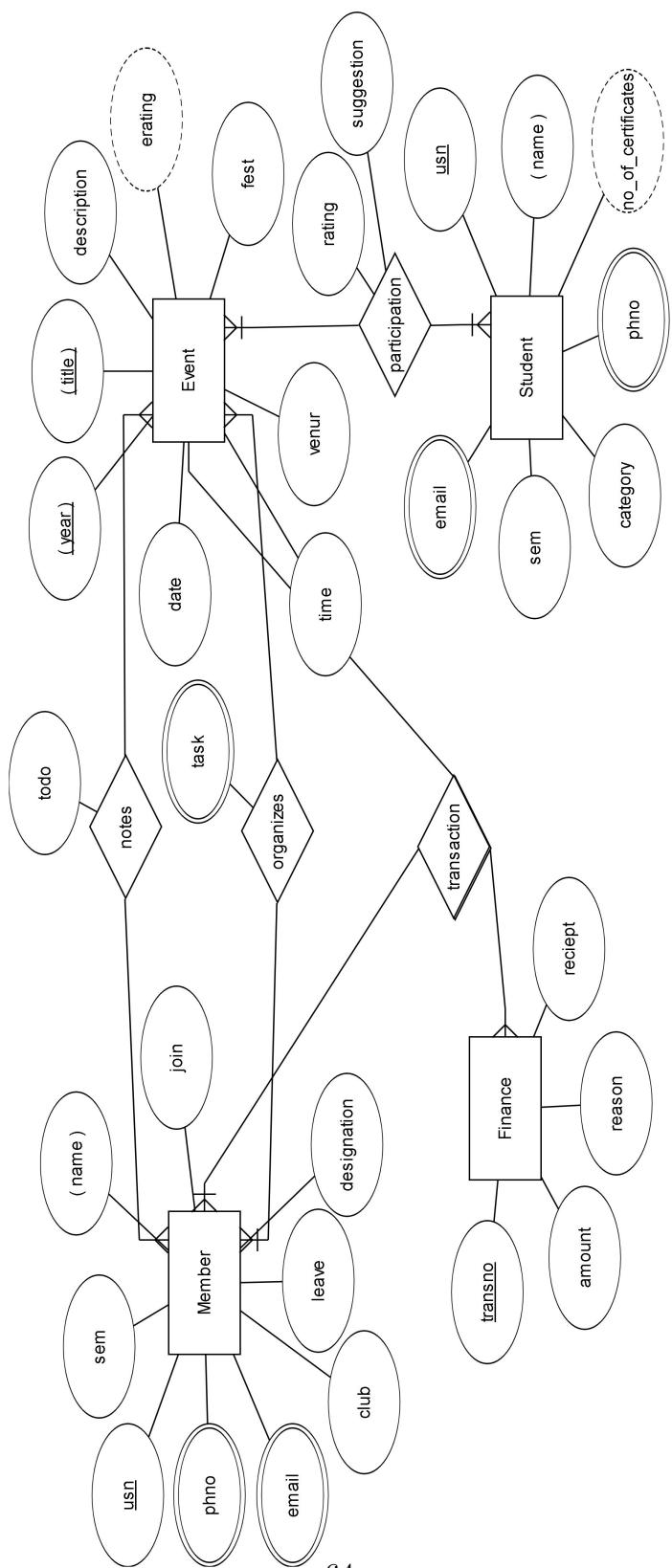


Figure 3.4: E-R Diagram

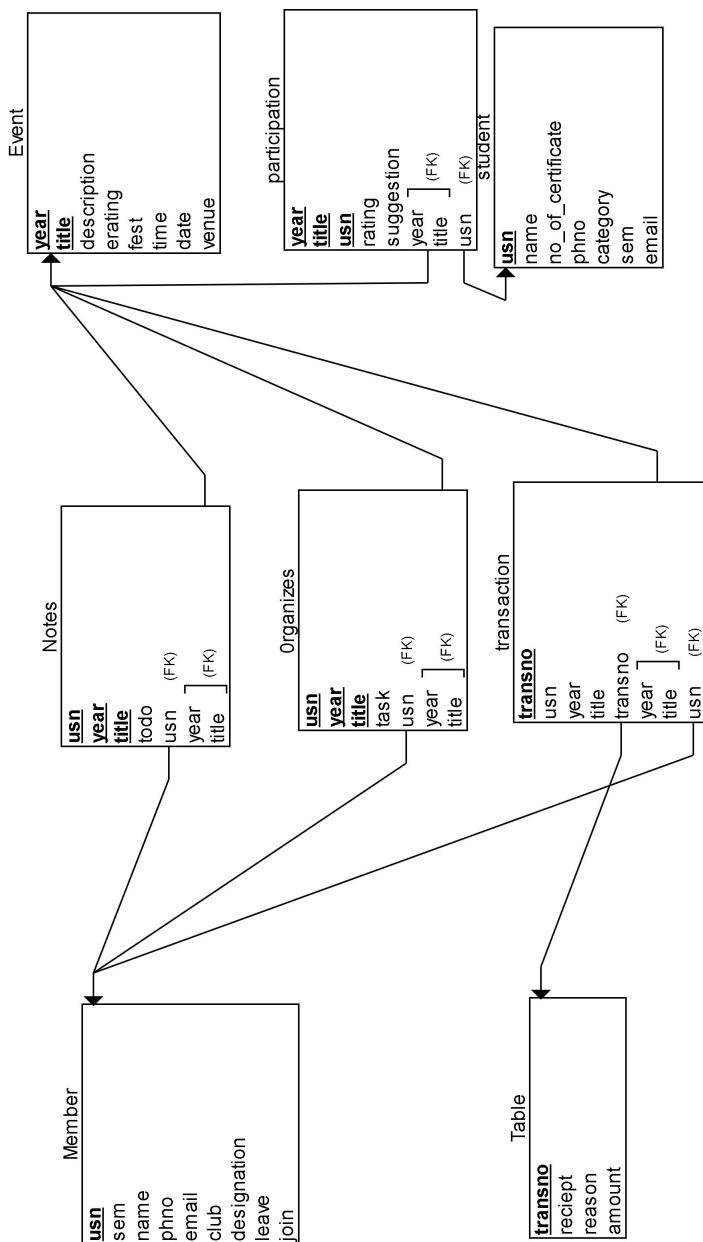


Figure 3.5: Club Management System

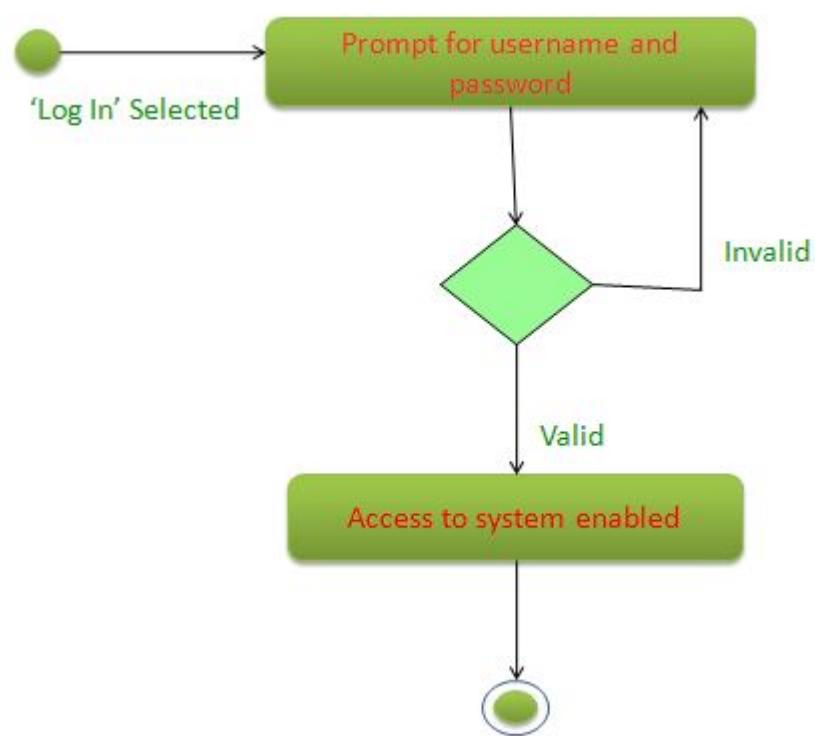


Figure 3.6:
presents 'Log in' Activity diagram for HOD, principal, member of the club.

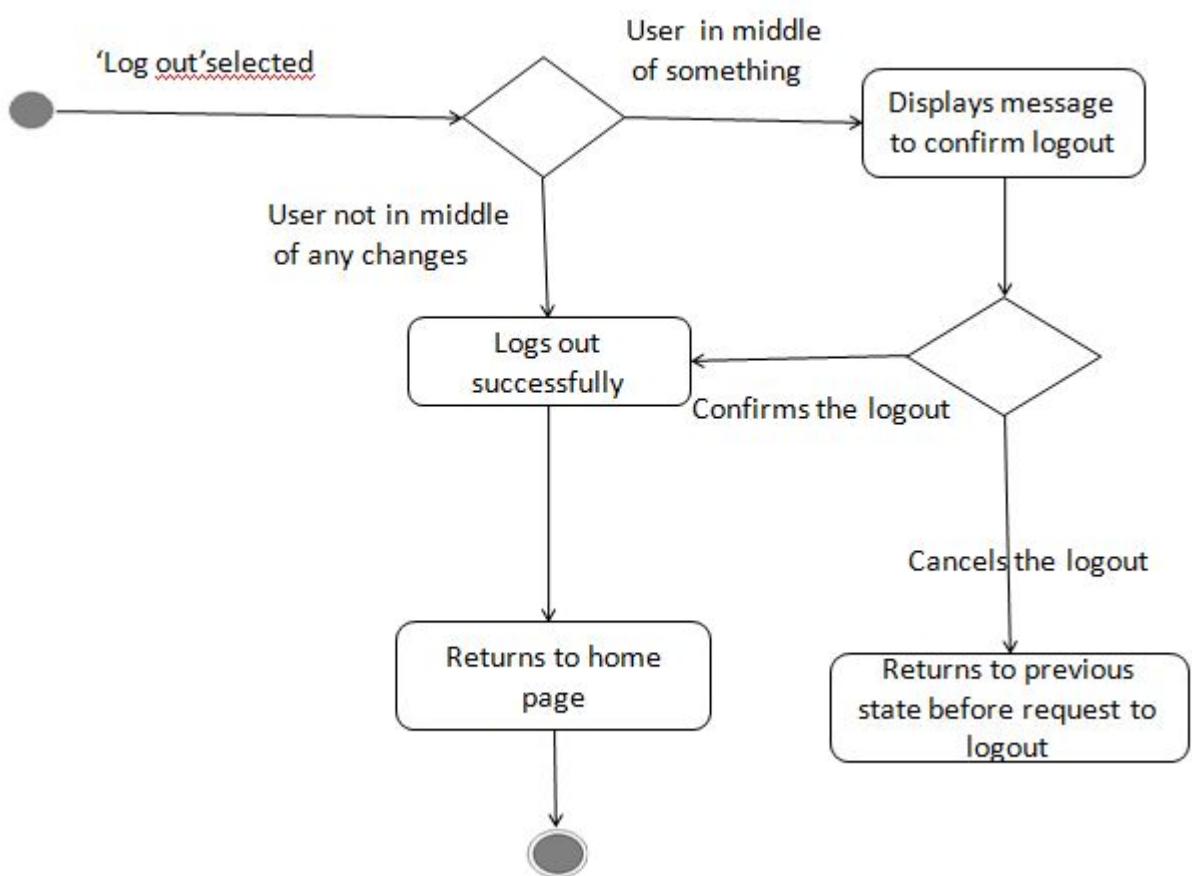


Figure 3.7: Presents 'Log out' Activity diagram

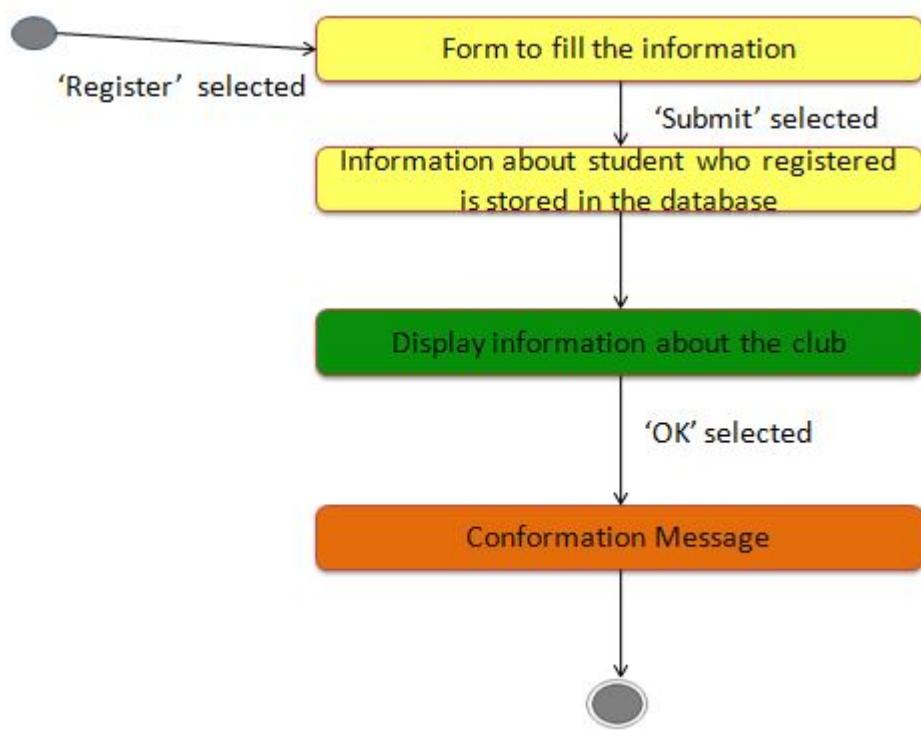


Figure 3.8: Presents 'Registration' Activity diagram

'VOLUNTEERS REGISTRATION' ACTIVITY DIAGRAM

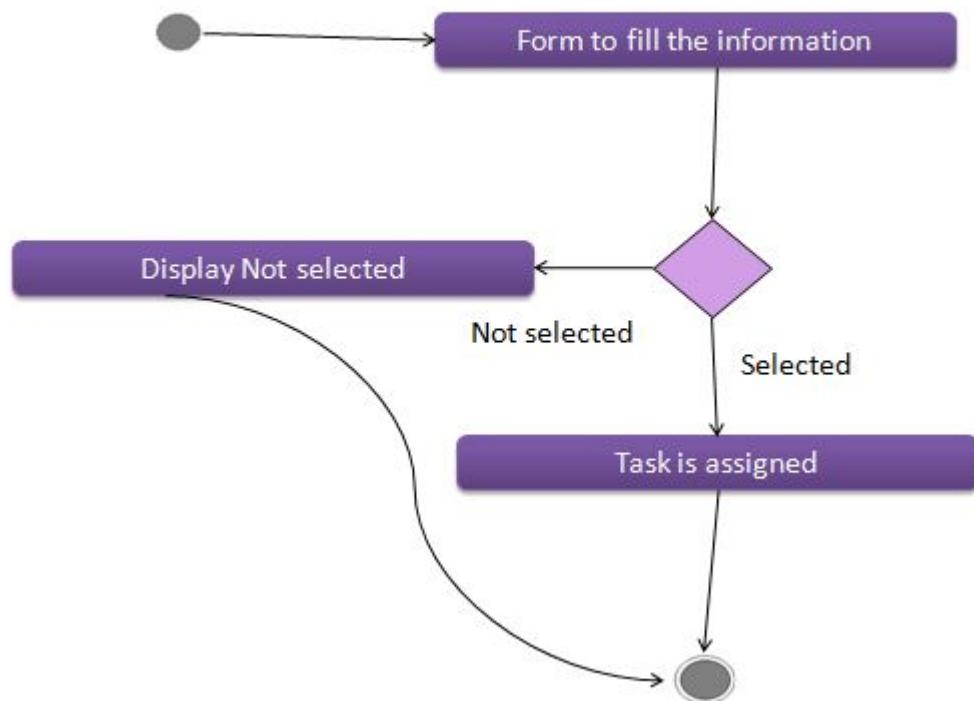


Figure 3.9: Presents 'Volunteers Registration' Activity diagram

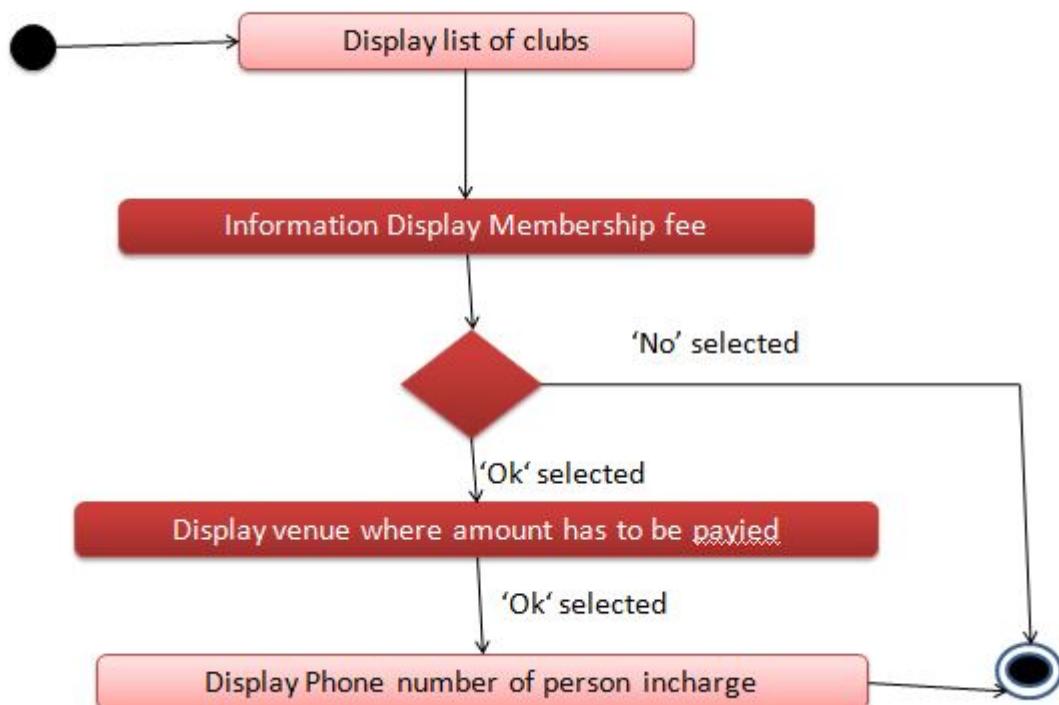


Figure 3.10: presents 'Membership selection' Activity diagram

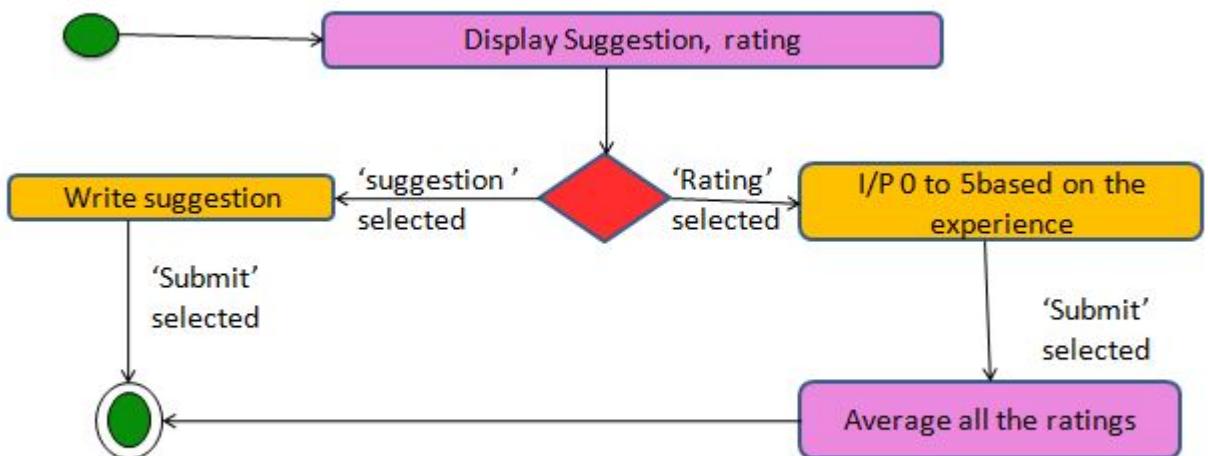


Figure 3.11: Presents 'Rating of an event' Activity diagram

Chapter 4

System Implementation

4.1 Introduction

4.1.1 Purpose

The purpose of this section is to detail the different ways in which the features of the product have been implemented. It includes descriptions in layman terms about each of the functions provided by the product without going into too much technical detail, so that the readers of the document are not burdened by unnecessary technical jargons.

4.1.2 Objective

This section attempts to describe the different aspects of the functionality provided by the product in order for the stakeholders to easily understand the working of the functions. This would allow them to efficiently use the product to suit thier specific needs. Here the stakeholders can also express thier requests for changes to be made once they understand the present functions.

4.2 Implementation

4.2.1 Event

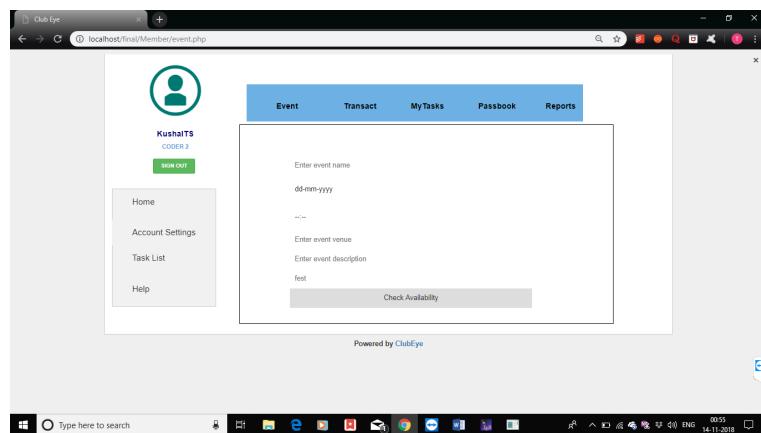


Figure 4.1: Event tab for member and student co-ordinator

This tab of the dashboard is common to both student co-ordinators and members. Through this page they can plan an event and add it to their list. Here an additional functionality is also provided so that the members who are planning events can check for clashes with events of other clubs. This function checks for clashes with respect to both time and venue i.e. if two clubs schedule events at the same time on the same day or if two clubs try to use the same venue on the same day respectively.

4.2.2 Transact

This tab allows the members or the student co-ordinators to store all the financial information related to their respective clubs. Here they have the functionality to enter the amount withdrawn or deposited for a specific event and then also add the receipt for that transaction. This provision for storing the receipts provides for a level of efficiency of financial management, as there is always a digital copy stored. The information about the member doing the transaction is also stored, this allows the club officials to keep track of the amount of funds spent a particular member for any future reference.

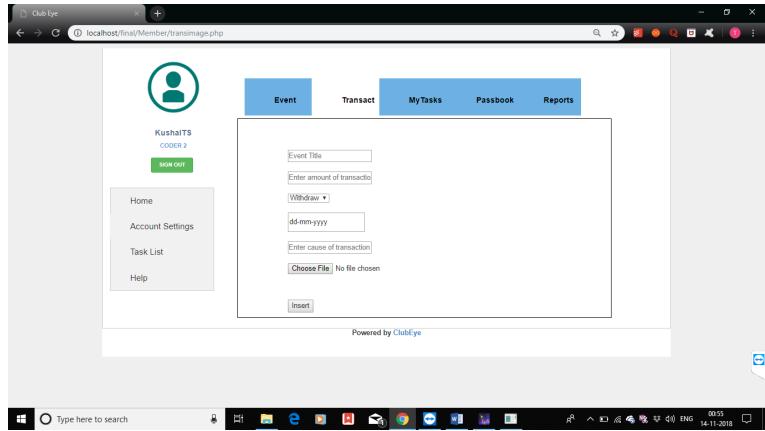


Figure 4.2: Transact tab for member and student co-ordinator

4.2.3 Tasks

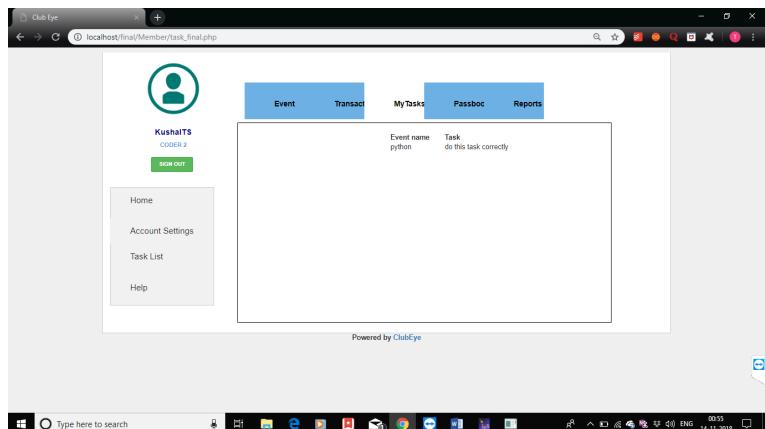


Figure 4.3: My Tasks tab for member and student co-ordinator

The above figure is a snapshot of one of the exciting features that was incorporated in the system and implemented for the Member point of view. This functionality would enable any member of a particular association or club to have a quick view of all the tasks assigned to him (by the student coordinator) for all the respective events. This would allow to reduce any communication gap or misinterpretation of information during the division of work among the members for any event that is to be conducted.

4.2.4 Passbook

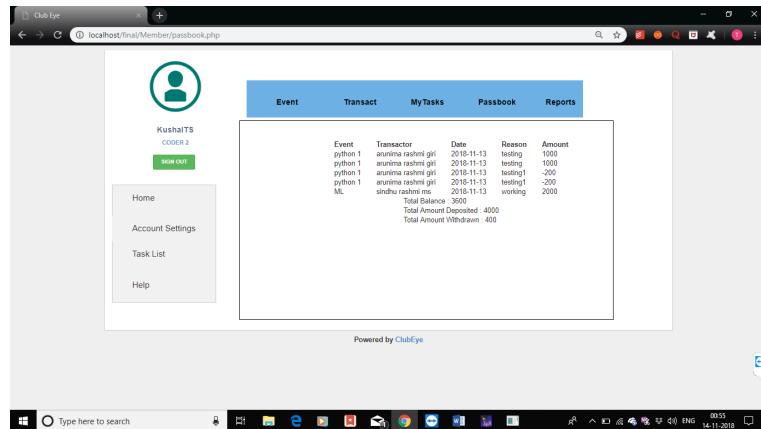


Figure 4.4: Passbook tab for member and student co-ordinator

The feature next up under taken for implementation was one of the most key features of the system. The passbook as the name suggests would hold all the transactions undertaken by a particular club or association .The visibility of it would be restricted to the Head of department in charge, student coordinators of the club and its members. It would hold details such as the person who transacted, the purpose, the event for which the transaction was made ,the amount withdrawn or deposited along with the total balance of the club wallet and the total expenditure and income. This feature would give a complete view of the financial activities of the club and would be used to account for every penny the club spends.

4.2.5 Reports

This tab generates different reports for the reference of the members or the student co-ordinators or the HOD of the club. The concerned personnel can generate reports based on different criteria and at different levels of detail. The different types of reports that can be generated are :

- **Event Info**

This option generates a report containing all of the information about

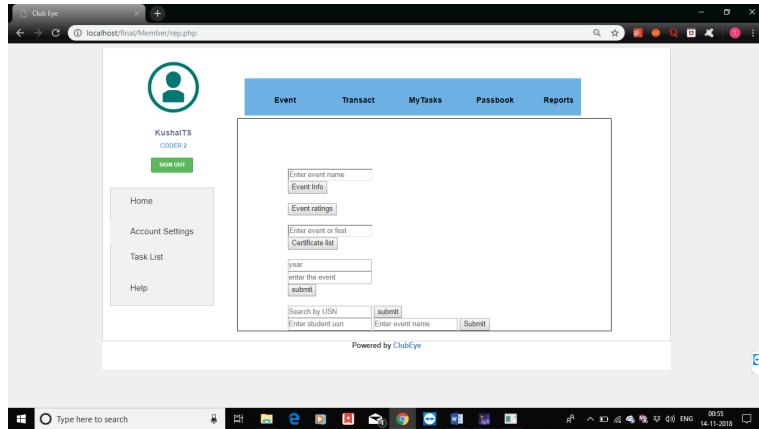


Figure 4.5: Report tab for member and student co-ordinator and HOD

the event for which a search is being done. This includes the event description, date, time, venue, rating and also all the students who have registered for that event as either volunteers or members.

- **Event Rating**

This option generates a report containing the ratings of all the events of the club. This allows for the members to analyse how each of the events are progressing and also to see how the entire club is functioning as a whole.

- **Certificate List**

Here certificates issued to all the students either as volunteers or participants for all the events hosted by the club are displayed, so that the members can keep track of the students to whom certificates are already given.

- **Financial Report**

Here financial reports are generated for either the year given or the combination of both year and event given. Hence financial reports for only selective transactions can be analysed.

- **Search by USN**

This option generates a report containing the details of the student whose USN is searched. This allows for the members to check on the progress of the students and decide on assigning designations for the future academic years.

- **Issue Certificate**

This option can be utilized by members to issue certificates to the students who have participated in the various events of a particular club as either a volunteer or participant.

4.2.6 Event Log

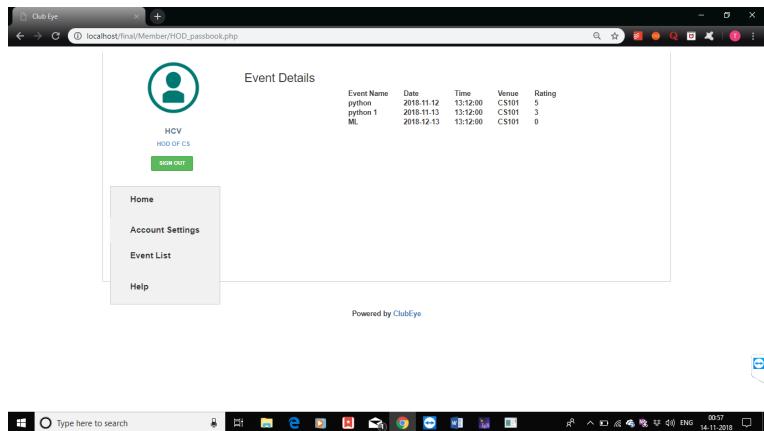


Figure 4.6: Event Log for HOD

The screen shot above shows the implementation adopted for a feature that allows the Head of the department to fetch a complete list of events that have already been organized or are to be organized by the club that is under her/his supervision. This to give the head of department the ability to view, approve, analyze the ratings of the events that a club or association that it has to undertake or has already done so. This also helps the HOD to keep track of all the activities of the club.

4.2.7 Power for Student Co-ordinator

The student co-ordinator has special privileges such as assigning a new task to a member for a particular event and changing the designation of a member of the same club. These are both restricted actions and hence can be performed only by the student co-ordinator.

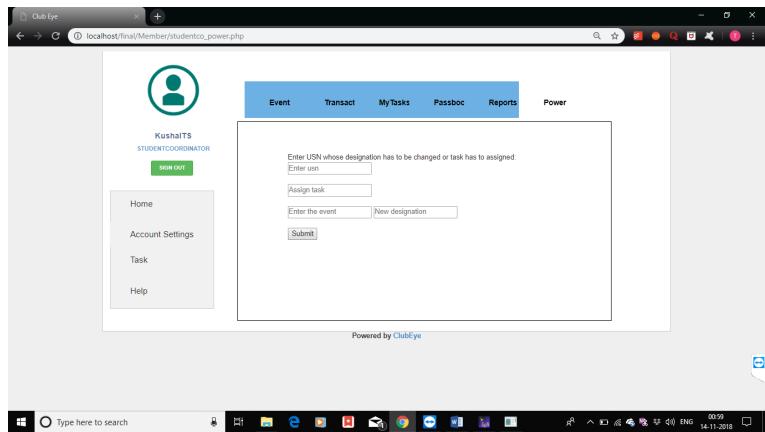


Figure 4.7: Power for Student Co-ordinator

4.2.8 Student Dashboard

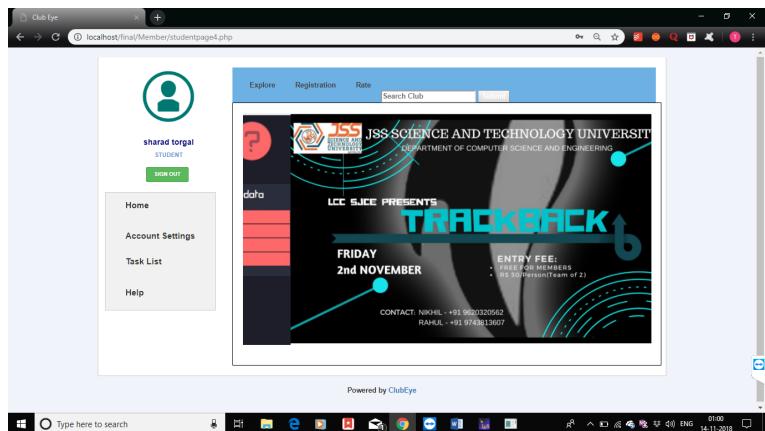


Figure 4.8: Student Dashboard

The student dashboard gives the students who log on details about events from every club which chooses to advertise its event through the product. The page also allows students to go to a registration and rating page for specific purposes.

4.2.9 Registration

This page allows the logged in students to register as either volunteers or participants for the event of their choice of any club registered with our product.

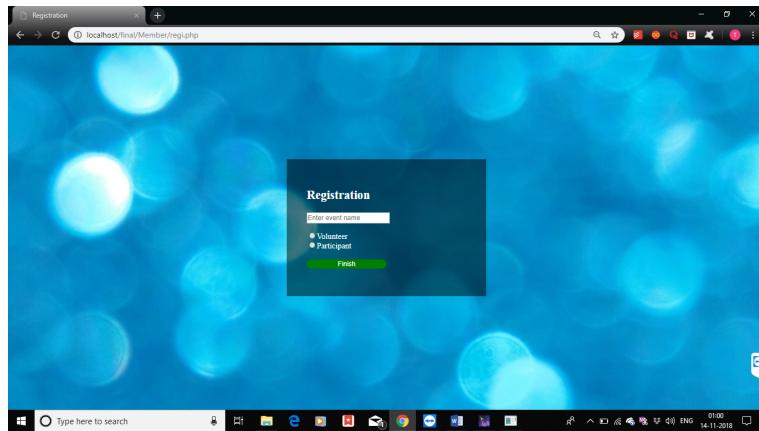


Figure 4.9: Registration for student

4.2.10 Rating

This page allows students to rate and also give suggestions for any event in which they have participated. This allows club personnel to analyse how different events are progressing and get feedback from students participating in them.

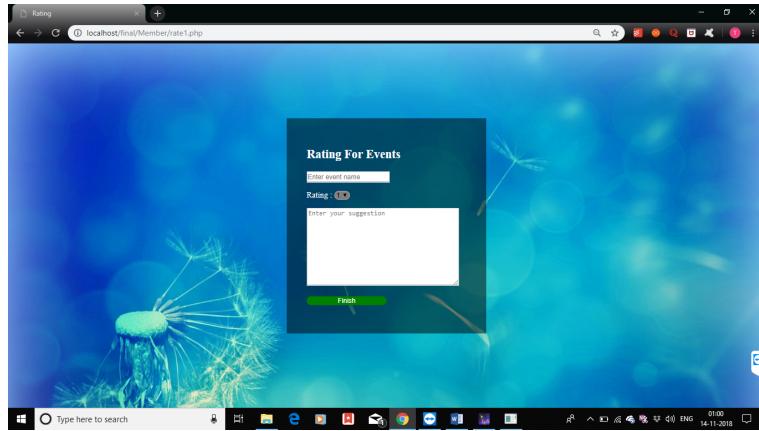


Figure 4.10: Rating for student

4.3 Coding

4.3.1 Introduction to SQL

Structure Query Language(SQL) is a programming language used for storing and managing data in RDBMS. SQL was the first commercial language introduced for E.F Codd's

Relational model. Today almost all RDBMS(MySql, Oracle, Infomix, Sybase, MS Access) uses SQL as the standard database language. SQL is used to perform all type of data operations in RDBMS . SQL is an ANSI (American National Standards Institute) standard. SQL is a data definition language. All DDL commands are auto-committed. That means it saves all the changes permanently in the database

- SQL can execute queries against a database.
- SQL can retrieve data from a database.
- SQL can insert records in a database.
- SQL can update records in a database.

- SQL can delete records from a database.
- SQL can create new databases.
- SQL can create new tables in a database.
- SQL can create stored procedures in a database.
- SQL can create views in a database.
- SQL can set permissions on tables, procedures, and views

4.3.2 Relational Algebraic Queries

Relational algebra is a procedural query language, which takes instances of relations as input and yields instances of relations as output. It uses operators to perform queries. An operator can be either unary or binary. They accept relations as their input and yield relations as their output. Relational algebra is performed recursively on a relation and intermediate results are also considered relations.

4.3.3 Queries designed using SQL Commands

- Selects name, usn, fest, title, category from certificate, student, event, participation entities.

```
SELECT name,S.usn as usn,fest,E.title,category FROM (((CERTIFICATE C JOIN STUDENT S ON S.usn=C.USN) JOIN EVENT E ON E.title=C.title) JOIN PARTICIPATION P ON P.usn=S.usn) WHERE fest =POST['Fest']
```

- To display name, semester, email, mobile number of the member.

```
SELECT name,sem,email,phno FROM ((MEMBER M JOIN MEMAIL E ON M.musn=E.musn) JOIN MPHNO P ON M.musn=P.musn) WHERE M.musn=SESSION['USN']
```

- Encryption of the password: We have used MD5. MD5 means a 128-bit encryption algorithm, generating a 32-character hexadecimal hash, whatever the captcha. This algorithms is not reversible, ie it is normally impossible to find the original word from the md5 hash.

```
SELECT * FROM MEMBER WHERE password= md5("oldpass.")  
AND musn="SESSION['USN']".
```

- To change password.

```
UPDATE MEMBER SET password = md5("newpass.") WHERE  
musr="SESSION['USN']";
```

- . To find the clash based on event date and event time.

```
"SELECT EVENT.title,clubname FROM (((CLUB JOIN MEMBER  
ON CLUB.clubid=MEMBER.clubid) JOIN ORGANIZES on ORGA-  
NIZES.musr=MEMBER.musr) JOIN EVENT ON ORGANIZES.title=EVENT.title)  
WHERE EVENT.Date="date." AND EVENT.Time = "time."";
```

- To find the clash based on event date and event venue.

```
"SELECT EVENT.Venue,clubname,EVENT.title FROM (((CLUB JOIN  
MEMBER ON CLUB.clubid=MEMBER.clubid) JOIN ORGANIZES  
on ORGANIZES.musr=MEMBER.musr) JOIN EVENT ON ORGA-  
NIZES.title=EVENT.title) WHERE EVENT.Date="date." AND EVENT.Venue  
= "venue."";
```

- . Inserting the records into the event.

```
"INSERT INTO EVENT (title,Date,Time,Venue,eventdesc,fest) values(" .title." , " .date." ,
```

- Inserting records into the organizes.

```
"INSERT INTO ORGANIZES (musr,title,year) VALUES (" .SESSION['USN']. " , " .title. " , " .year. " );
```

- . To fetch the description of event.

```
"SELECT eventdesc FROM EVENT WHERE title=" .SESSION['event']. " ";
```

- To display the financial status of the club.

```
"SELECT m.name,e.title,amt,reason,f.date FROM ((member m JOIN
```

```
transby t on m.musn=t.musn) JOIN event e on t.title=e.title) JOIN  
finance f on t.transno=f.transno);
```

- Aggregate functions.

```
"SELECT sum(amt) as sum FROM FINANCE"; "SELECT sum(amt)  
as sum FROM FINANCE WHERE amt<0"; "SELECT -1 * sum(amt)  
as sum FROM FINANCE WHERE amt>0";
```

- To display name, email, department of the HOD.

```
"SELECT name,HODemail,dept FROM HOD WHERE HODemail=)".SESSION['email']";
```

- Fetching information of the HOD from the password.

```
"SELECT * FROM HOD WHERE password= md5("oldpass.") AND  
HODemail=".SESSION['email']";
```

- Changing the password.

```
"UPDATE HOD SET password = md5("newpass.") WHERE HOD-  
mail=".SESSION['email']";
```

- To display title, date, time, venue, erating of the events. "SELECT
E.title as title,Date,Time,Venue,erating FROM (((HOD H JOIN CLUB
C ON H.HODemail=C.HODemail) JOIN MEMBER M ON C.clubid=M.clubid)
JOIN ORGANIZES O ON O.musn=M.musn) JOIN EVENT E ON
E.title=O.title) WHERE H.HODemail=".SESSION['email']";
- Fetching information about finance based on the transaction number.

```
"SELECT * FROM finance WHERE transno=".SESSION['transno'];
```

- To get information like date, time, venue, erating of the event.

```
"SELECT Date,Time,Venue,erating FROM EVENT WHERE title=".SESSION['event']";
```

- Checking whether member has registered or not.

```
"SELECT * FROM MEMBER WHERE musn = ".usn." AND password ='  
.password2."';
```

- Checking whether student has registered or not.

"SELECT * FROM STUDENT WHERE usn = "" .usn." AND password ='
".password2."";

- . Selecting student coordinator from the member entity.

"SELECT * FROM MEMBERE WHERE musn= "" .usn." AND password ='
".password." AND designation = 'studentcoordinator"';

- Select the members based on their usn.

"SELECT * FROM MEMBER WHERE musn= "" .usn."";

- . Select student based on their usn.

"SELECT * FROM STUDENT WHERE usn= "" .usn."";

- Select Club based on their club ID and Club Password.

"SELECT * FROM CLUB WHERE clubid= "" .clubid." AND clubpassword ='
".cp."";

- Inserting records into member.

"INSERT INTO MEMBER (musn, fname, mname, lname, sem, designation, joindate, password,
VALUES ('' .usn.'', '' .fn.'', '' .mn.'', '' .ln.'', '' .sem.'', '' .desc.'', '' .jd.'', '' .p.'', '' .cluid.'')"

- Inserting email ID to the MEMAIL entity.

"INSERT INTO MEMAIL (musn, email) VALUES ('' .usn.'', '' .email.'');
mysqliquery(con,sql1);

- Inserting phone number to the MPHNO entity.

"INSERT INTO MPHNO (musn, phno) VALUES ('' .usn.'', '' .phno.'');

- To display financial status.

"SELECT m.name, e.title, amt, reason, f.date
FROM (((member m JOIN
transby t ON m.musn=t.musn) JOIN event e ON t.title=e.title) JOIN
finance f ON t.transno=f.transno)";

- To display list of events with their ratings.

"SELECT title, rating FROM EVENT";

- To display information of the participating students in an event.

```
"SELECT name,S.usn as usn,category FROM PARTICIPATION P
JOIN STUDENT S ON P.usn=S.usn WHERE title="" .SESSION['event']."'";
```

- Inserting records into the student entity.

```
"INSERT INTO STUDENT (usn,fname,mname,lname,sem,password)
VALUES (''.usn.'','.fn.'','.mn.'','.ln.'','.sem.'','.p.'')";
```

- Inserting email ID's to the SEMAIL entity.

```
"INSERT INTO SEMAIL (USN,email) VALUES (''.usn.'','.email.'')";
```

- Inserting phone numbers into SPHNO entity.

```
"INSERT INTO SPHNO (USN,phno) VALUES (''.usn.'','.phno.'')";
```

- Inserting the information about the events conducted by the clubs.

```
"INSERT INTO EVENT (title,Date,Time,Venue,eventdesc,fest) values(''.title.'','.date.'','
```

- Assigning the task to the members.

```
"UPDATE ORGANIZES SET task=''.task.'' WHERE musn=''.usn.''
AND title=''.title.''";
```

- Student coordinator changing the designation of the member.

```
"UPDATE MEMBER SET designation=''.des.'' WHERE musn=''.usn.'''";
```

- To display the record in the organizes based on the usn.

```
"SELECT * FROM ORGANIZES WHERE musn=''.musn.''";
```

- Inserting the images of the receipts.

```
addslashes(filegetcontents(FILES["image"]["tmpname"]));
```

- Inserting records to the finance entity.

```
"INSERT INTO finance (amt,reason,date,receipt) VALUES (''.amt.'','.reason.'','.date.'''";
```

- Select records from the student entity based on the usn.

”SELECT * FROM STUDENT WHERE usn=’’.usn.’’’;

- Aggregate function: Counting the number of certificates.

”SELECT COUNT(*) as count FROM PARTICIPATION WHERE usn=’’.usn.’’’;

- To display event in which a particular student is participating.

”SELECT title FROM PARTICIPATION WHERE usn=’’.usn.’’’;

- Natural join: To display transactions of the events that is amount of money deposited or amount of money withdrawn.

”SELECT * FROM (TRANSBY T NATURAL JOIN FINANCE F) WHERE title=’’.title.’’ANDyear =’’.year.’’’;

Chapter 5

System testing and result analysis

5.1 Introduction

5.1.1 Purpose

This section holds a brief description of a test plan for Club Managerial System Testing, produced by the Testing team. It describes the testing strategy and approach to testing the team adopted to validate the correctness of the functionalities incorporated as specified in the requirement section.

5.1.2 Objective

- Meets the requirements, specifications and the Business rules.
- Supports the intended business functions and achieves the required software standards.
- Satisfies all functional requirement criteria, check effectiveness, efficiency and performance of the system .

5.2 Overall Strategy and Approach for Testing

Club Management System Testing will include testing of all functionalities that are in scope (Refer Functional Scope Section) identified. System testing activities will include the testing of new functionalities, modified functionalities, screen level validations, work flows, functionality access, testing of internal and external interfaces.

5.2.1 Usability Testing

User interface attributes, cosmetic presentation and content will be tested for accuracy and general usability. The goal of Usability Testing is to ensure that the User Interface is comfortable to use and provides the user with consistent and appropriate access and navigation through the functions of the application (e.g., access keys, consistent tab order, readable fonts etc.)

5.2.2 Functional Testing

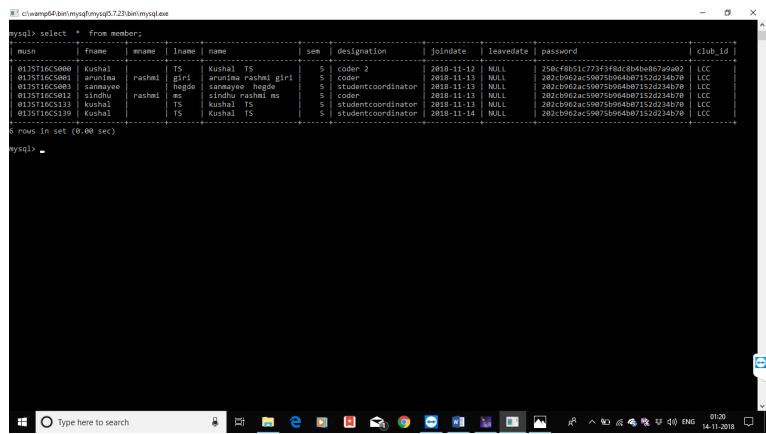
The objective of this test is to ensure that each element of the component meets the functional requirements of the business as outlined in the:

- Business / Functional Requirements
- Business rules or conditions
- Other functional documents produced during the course of the project i.e. resolution to issues/change requests/feedback

5.3 Testing Implementation

5.3.1 Unit Testing

The very first level of testing applied was by our developers ,in which they focused in checking the correctness of all the implemented code both from the GUI point of view as well as the physical storage point of view. In this layer of testing we had basically two of our developers working on the data end and the other two on the representational end of the software .Each pair after successful implementation of a particular module would test its correctness with a self-assessed bound condition input(maximum limit and minimum limit of users/accesses/data length etc) ,and then compare with the expected outcome with the least approximation possible. This half of testing of each module could be regarded as the white box testing . Once both the front and the back end halves were up and running we switched the modules amongst the pairs and began testing exhaustively the implementation as developed by the other pair, respectively. In this stage on encountering discrepancies with respect to any of the specified requirements we'd notify the pair(developer) of the same .This method of testing adopted by us was done by each pair with a mindset of breaking the code as put forward by the developing pair (Black box testing). Instances among many, when we stumbled upon these kinds of discrepancies are mentioned below.It was found that the password details which were initially stored as integer type might be secured away from the public access but would not be restricted from the developing/maintenance team which would be ethically and conceptually incorrect .To provide customers with a greater sense of security we then introduced the md5 type of encryption technique to hide all passwords from direct representation. Another instance was that the database end incorporated various features including deleting of a member when he/she exits the institution .But soon this was pointed out by the other pair that by doing so we lose hold of the member detail of any transaction made by the person leaving ,there compromising on the functionality that hold the transactor accounted by all transactions made in the significant past. So then the data front end restricted the deletion of any member information only in a bunch fashion dating 15-20 years back.



The screenshot shows a MySQL command-line interface window titled 'c:\wamp\mysql\mysql5.7.23\bin\mysql.exe'. The command entered is 'mysql> select * From member;'. The results show a table with columns: id, name, fname, lname, name, sem, designation, joindate, leave date, password, and club_id. The 'password' column contains hashed values starting with '25acfb8fb51c773f9fdcb0d8b867a9a02'. The 'club_id' column shows values 'LCC' or 'LCC' repeated. The command '0 rows in set (0.00 sec)' is displayed at the bottom.

Figure 5.1: Password encryption with md5 technique

5.3.2 Acceptance Testing

When the entire software product was ready and we had the confidence that it is working correctly and that it will satisfy all the requirements which are required to prove itself in the real world, our software product was exposed to Acceptance testing. This software whose primary purpose is to bring automation in manual management of the clubs which are functioning in our University, had the many stakeholders from those clubs. Our software product received a very positive response from these stakeholders who performed the acceptance testing of our product. They conducted a careful go through of our product to see whether it satisfies all the criterias mentioned by them during the requirement collection phase and it passed all the acceptance use cases. Operational acceptance testing of the product was carried out by them which also was a good feedback because our software is easy to maintain, it is reliable and also we have provided an efficient technical support. Our team efficiently clarified all their doubts and explained all the features that our product incorporates. Some of the functionalities demanded by them like solution to the problem of clash of events with other clubs, secured financial transaction management etc, which our product delivers efficiently made the customers extremely happy. After the entire validation was completed, we even listened to their suggestions regarding the functionalities provided by the product and we promise to help our stakeholders in future with any additional requirements that they desire.



Figure 5.2: Acceptance Testing by IEEE memeber Swathi S, EDS JOINT SECRETORY.

Chapter 6

Conclusion and future work

6.1 Conclusion

The main aim of our product is to provide services to the stakeholders which make their task of managing their clubs and associations more efficient and easier. The product provides an easy to use platform for different members to interact and plan their various events in a highly logical and conducive manner. The stakeholders would have easy access to all of their information regarding events, finance etc. with the help of our service. They would also be able to make changes brisk changes to any of the club related information, which would be immediately reflected in the data seen by the other members. The product provides high level of security in the case of financial transactions with end to end encryption by using methods such as OTPs, biometrics etc., this would ensure that all of the transactions are made only by the authorized personnel and would definitely decrease the chances of foul play. The club administrators would also have help in choosing future member and appointing heads od committees with the help of the data collected through the product, which would be available to them in a pictorial and easy to read manner such as activity graphs for each member. This would enable the higher-ups to make informed decisions based on the actual information acquired and filtered for this very purpose. The product could

also be configured to work seamlessly across different networks based on the resources available. We hope that by opting for product clubs would be able to forgo the huge amount of paperwork that they usually handle and as such avoid the human errors which occur when filling this paperwork. Through our services the need of the clubs to store all of their paper receipts for purchases made for the events of the club manually would be rendered moot, as our product has provisions to store the receipts digitally and hence these can be accessed easily and without hassle. The product encourages the clubs who use it to be completely digitized in their record keeping and avoid the unnecessary paperwork.

6.2 Future Work

We have fulfilled all the requirements of our stakeholders and we hope that our product can further be developed with enough resources and manpower so that it can handle the management of large organizations such as large-scale companies.

6.3 References

- ieee-collabratec.ieee.org
- [ieee latex documentation](http://www.latex-project.org)
- Software Engineering: A Practitioner's Approach by Roger S Pressman
- www.lccsjce.org