CU_NOC

Database Project Report

Group-08

Report submitted February, 2023

A project submitted to Dr. Rudra Pratap Deb Nath, Associate Professor, Department of Computer Science and Engineering, Chittagong University (CU) in partial fulfillment of the requirements for the Database Systems Lab course. The project is not submitted to any other organization at the same time.

Table 1: Details of Group-08

Roll Id	Name	Sigature	Date	Supervisor Approval
20701040	Md. Sabbir Hasan Bhuiyan			
20701052	Tapos Chandra Das			
20701067	Sourov Karmakar			
20701080	Farhana Sultana Ananna			

Contents

List of Figures

List of Tables

Listings

Abstract

Willing to fully automatize the CU_NOC application procedure , this is our first step to work with our Study Leave application system. Here, we are super exited to offer a computerized system to the applicants and related officials. The current manual procedure to collect, submit and evaluate the application is quite time consuming and costly where each of the individuals relating to this application system needs to undergo an avoidable pain. Basically , this is a common platform where the applicants can apply for Study Leave which will automatically be sent to the related officials. Subsequently , each of the related officials will receive an email notification to check for the validity of the application. After evaluation, an approval notification and related document will be sent to the applicant in pdf format which can be downloaded for further step. However, in this project we are dealing with Study Leave Application system only; the related other NOC application system will be added in future.

1 Introduction

Motivated to digitalize the current NOC application system of University of Chittagong , we , a team of enthusiastic learner , took this project as a challenge in our DBMS course. Although , we are willing to digitalize the complete NOC application system , here in this course we will work only on the part of Study leave application where the teachers and related official will be the chief beneficiary.

The objective of this course is to develop a database application system by applying the theories, methodologies, tools, and technologies we learnt in the theory course **Database Systems** with course id **CSE413** .

1.1 Background and Motivation

The present state reflects the chaotic manual way of handling the Study Leave Application System; the University of Chittagong did no exception to that. More elaborately, the whole application system is maintained hand on hand. Teachers need to submit all the required forms and hassle around all the required departments. Officials need a 3-4 week long manual process of evaluation. As an IT tech child, we found the motivation to change the world. Why not to start it from home ! Which is CSE CU.

Main aim of our platform is to computerize this system so that both applicants and officials can maintain this lengthy process within the least possible time.

1.2 Problem Statement

The problem we found to address in the Study Leave Application process is listed below from different perspective :

- In the beginning of the process, **applicants(teachers)** need to manually collect and fill up the application form ,and then submit at the **chairman** of respective department.
- Secondly, the form is sent to the **register** office and then to the **Higher Study branch** for respective evaluation. Respectively as a part of evaluation the Higher Study branch needs to contact with a number of departments(library, medical etc) , to check if their is any issues with the respective applicant.

- Subsequently , the form is send manually to the register for **VC approval**, which then backs to higher study branch through register.
- Finally,the respective **applicant** gets the final confirmation.

Here ,our system offers those manual process of form traversal will be automated, and there will be a tracking system for the applicant to check for the **current condition** of the application approval.

1.3 System Definition

"A computerized system used to manage CU_NOC application procedure , starting with the receiving of application from the the applicants to its evaluation at different levels by the officials. Controlling should be easy to learn, as the officials at different levels are untrained labor."

1.4 System Development Process

To develop the whole process the following steps needs to be done:

- **Requirement Gathering And Analysis:** We believe discussion and interviews can bring all possible requirements and data that are the core of this project.Collecting data from throw interview with our respected seniors and official is our first motive and then we will consult with our team members.
- **Conceptual Modeling:** The organized data will take us to the next step which is tracking all the entity types . finally it the shape of ER diagram.
- **Normalization:** After accomplishing the previous steps we have to confirm the data table is normalized.Our Target is to get 3NF.
- **System Architecture:** Afterwards we will achieve a user friendly system architecture.
- **Implementation:** We came to a decision to develop our front end with HTML,CSS and JAVASCRIPT . PHP and MYSQL will be our foundation of backend.
- **Validation:** Verifying our system from different end is our ultimate step.

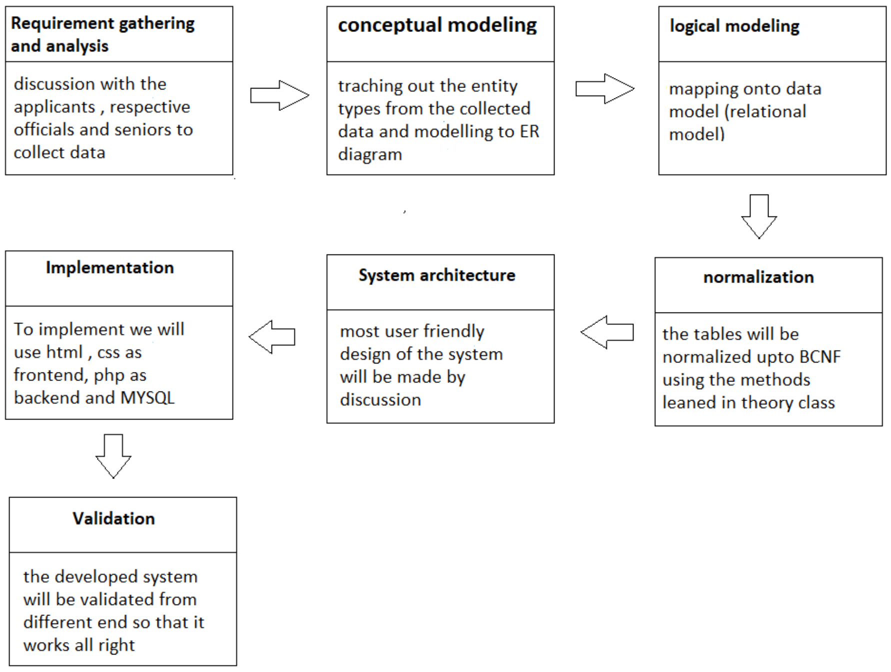

Requirement gathering and analysis
discussion with the applicants , respective officials and seniors to collect data
conceptual modeling
traching out the entity types from the collected data and modelling to ER diagram
logical modeling
mapping onto data model (relational model)
normalization
the tables will be normalized upto BCNF using the methods leaned in theory class
System architecture
most user friendly design of the system will be made by discussion
Implementation
To implement we will use html , css as frontend, php as backend and MYSQL
Validation
the developed system will be validated from different end so that it works all right

Figure 1: System Development Process

1.5 Organization

Organization of this document is illustrated in the following paragraph.

Section 1 gives the overview of the project, Section 2 describes how the project and the resources are managed , Section 3 tells how the requirements are gathered , Section 5 describes the logical model ,,, others will be written when done

2 Project Management

Organization and management of resources : Tolls like, **GitHub** and **Trello** are used mainly for better collaboration among the members. To illustrate, resources are shared among the members using the following GitHub repository https://github.com/SabbirHasanBhuiyan/CU_NOC. The repository mentioned here has 3 folders; namely, code, practice, and resources for dedicated purposes. On the other hand, Trello is used for assigning work to every individual and taking updates on their progress from them. Shortly, to explain our Trello board, cards are added to assign a task which undergoes through "doing"and "done"steps until complete accomplishment of the task.

Roles of Each member :

- **Md. Sabbir Hasan Bhiyan :** Requirement gathering , report writing , implementing database , implementing back end and helping to design front end .
- **Sourov Karmakar :** Modeling ER diagram using figma , implementing database , implementing back end .
- **Tapos Chandra Das :** implementing back end , implementing database and helping to design front end .
- **Farhana Sultana Ananna :** Implementing front end, implementing database, writing press release.

3 Requirement Gathering and analysis

To gather the information for our project one of our senior Atanu Kumar Dey helped us the most, who had completed such a project. We had a online and a offline session with him. In the online session he briefed us about flow Study Leave Application forom. Afterward, we had a hand written document of the interview with him in an offline session. Besides, our course instructor Associate Professor *Dr.Rudra Pratap Deb Nath*, guided us about the application procedure. Moreover, we collected a pdf document to look for the final approval document format from our university website noticeboard https://cu.ac.bd/noticeboard/.

After successful analysis of our gathered requirements, what conclusion we could reach is mentioned below:

- **Stakeholder:** The primary users of our project are applicants of the Study Leave Application (teachers). Besides, among secondary users are chairman of the department, registrar office, higher study branch and some other departments like library, medical etc.

- **Functional Requirements:** Functional requirements are product features or functions that developers must implement to enable users to accomplish their tasks.[1]The functional requirements found to implement in our system are:
  - enabling users to create account
  - user authentication during login
  - password recovery option
  - having a database with all the constant data of the applicant, so that the he needs to input some specific data relating to his current Study Leave Application only
  - creating views so that users can see his approved part only

- **Non-functional requirements:** Non-functional Requirements (NFRs) define system attributes such as security, reliability, performance, maintainability, scalability, and usability. [2]
  - hashing is used for security of the password
  - ensuring most optimized algorithm so that the user experience great performance

- making the platform in such a way that other NOC Application systems can be integreted easily
- try implementing modular code for the betterment of re-usability and main tainability

4 Conceptual Modelling

Conceptual modeling in database design is the process of creating a high-level representation of real-life objects and the relationships between them by translating user requirements. It defines the structure of a database and the relationships between different data elements. For Conceptual design, we are using a popular representation called **Entity-Relationship Model**. Using the entity-relationship model, the conceptual schema specifies the entities that are represented in the database, the attributes of the entities, the relationship among the entities, and the constraints on entities and relationships.

4.1 Entity-Relationship Diagram

Entity-Relationship diagram, also known as ER diagram is a graphical way of representing entities and their relationships with one another. It is a very useful technique for visualizing the structure of a database and identifying any potential problems or inconsistencies. The basic components of ER model include the following:

1. Entity Set
2. Attributes
3. Relationships

4.2 Entity, Attributes, and Relationship Finding Process

To find Entity Set from the requirement analysis, we have grouped all nouns and given a common name to them. Descriptive properties of these Entity Sets have been taken as Attributes. For Relationships, we analyzed verbs of our collected requirements.

After translating system requirements for the ER model, we found the following Entity Set and their attributes for our ER diagram :

1. **USER:** To store basic details of users for login, USER is a generalized entity set for all users using our system.
   Attributes: This includes ID, Name, Email, Password, and Dept.

2. **Applicant:** We also have a type of user that applies for study leave. These collections of applicants can be grouped as an Applicant entity

set.
Attributes: This inherits every attribute of USER and contains an extra attribute, Designation.

3. **Evaluator:** For evaluating applications, we can create an entity set Evaluator.
Attributes: This inherits all of the attributes of USER and adds Working_Role.

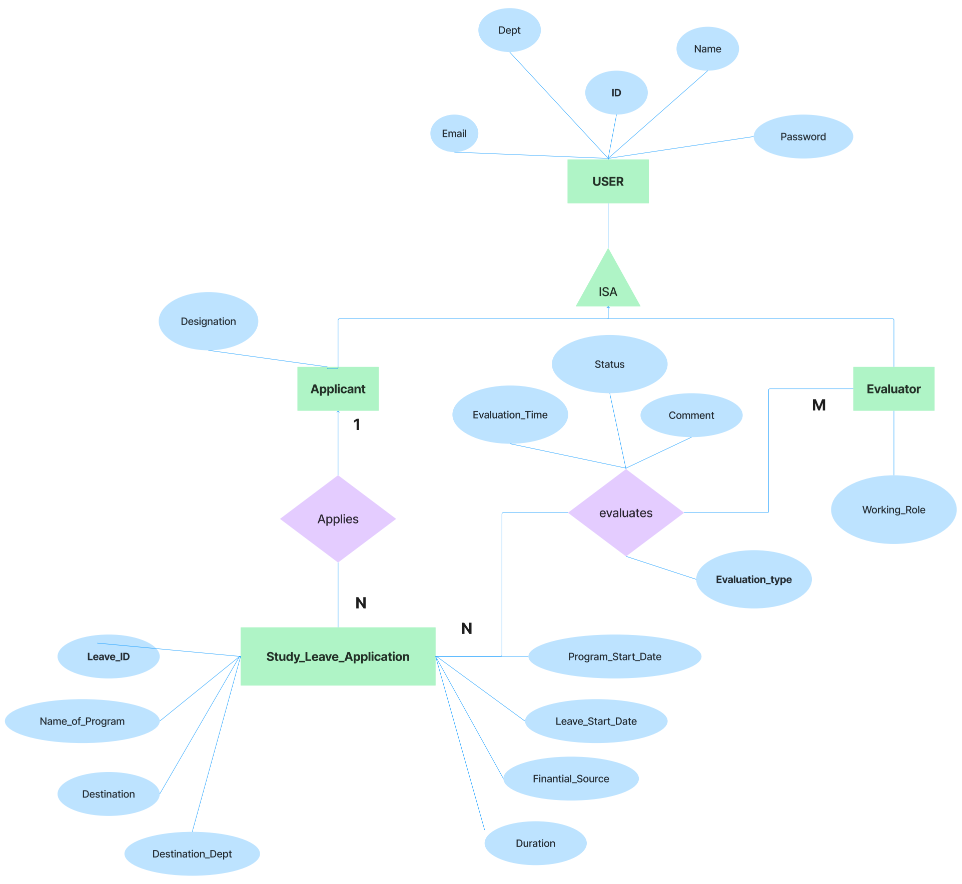
4. **Study_Leave_Application:** For collecting information and necessary documents about Study leave, we need an entity-set Study_Leave_Application.
Attributes: This includes Leave_ID, Name_of_Program, Destination, Destination_Dept, Program_Start_Date,Leave_Start_Date, Financial_Source and Duration.

From our analysis, we also found relationships between some of our entities. Such as:

- Applies: Applicant applies for Study_Leave_Application. This is a one-to-many relationship because one applicant can apply for more than one study leave.

- Evaluates: Evaluator evaluates Study_Leave_Application. This is a many-to-many relationship because one evaluator can evaluate many applications, also there is more than one evaluator who evaluates an application. Evaluations also include four attributes: Evaluation_type, Status, Evaluation_time, and Comment.

Also, there exist two special **ISA** relationships with USER and Applicant, USER and Evaluator. Both Applicant and Evaluator is a user and inherits all attributes of the USER entity.



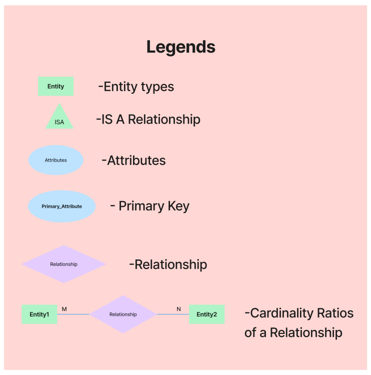


Figure 2: Entity Relationship Model of CU_NOC with Notation

5 Logical Modelling

After the Conceptual Design phase, we mapped our high-level conceptual schema onto the logical data model. This logical data model is typically the relational data model, and this step typically consists of mapping the conceptual schema defined using the entity-relationship model into a relational schema.

5.1 Relational Model

The Relational model is a way of organizing and representing data in a database as a collection of tables with rows and columns. Each table represents an entity, and the columns represent attributes of that entity. The relationships between entities are established through common columns, also known as keys. In the relational model, data is organized in a systematic and structured manner, making it easy to query, update, and manipulate.

5.2 Conversion from the ER model to the Relational model

Conversion from the ER model to the Relational model involves the following steps:

1. A table is assigned to each entity.
2. The attributes of each entity are the columns of that table.
3. After that, we define the primary key.
4. If there is a one-to-one relationship between two entities, we add the primary key of either side to the other side as a foreign key.
5. And if there is a one-to-many relationship between two entities, we add the primary key of the left side to the right side as a foreign key.
6. Also, if there is a many-to-many relationship between two entities, we create a separate table and add the primary key of both sides to it.
7. Finally, for Generalization(ISA relationship), we have four alternatives: **main classes, partitioning, full redundancy,** and **single relation**.

From our ER diagram, we have decomposed our **ISA** relationship by partitioning, which means we created three tables named USER, Applicant, and Evaluator. We used the ID of the USER as a foreign key, as well as the primary key of the Applicant and Evaluator.

After that, we have an entity named Study_Leave_Application. We have created a table for it and added all the attributes as columns. We have declared Leave_ID as the primary key of this table. It relates to the Applicant and there has a one-to-many relationship between them (one applicant can apply for many applications). So we have added the primary key of Applicant to the Study_Leave_Application table.

Lastly, there is a many-to-many relationship between Study_Leave_Application and Evaluator because we have many types of evaluators that evaluate many applications. So we have created a separate table called ”Evaluates” and added the primary key of Study_Leave_Application and Evaluator to it. Also, this consists of four other attributes: Evaluation_type, status, evaluation_time, and comment. The primary key of this table is taken as a combination of Evaluation_type, Leave_ID of Study_Leave_Application, and ID of Evaluator.

5.3 Final List of Relational Schema

- **USER:** {[<u>ID</u>, Name, Email, Password, Dept]}
- **Applicant:** {[<u>ID→USER</u>, Designation]}
- **Evaluator:** {[<u>ID →USER</u>, Working_Role]}
- **Study_Leave_Application:** {[<u>Leave_ID</u>,ID→Applicant, Name_of_Program, Destination, Destination_Dept, Program_Start_Date, Leave_Start_Date, Financial_Source, Duration]}
- **Evaluates:**{[<u>Evaluation_type,Leave_ID→Study_Leave_Application, ID→Evaluator</u>, status, evaluation_time, comment]}

6 Normalization

Normalization is a database design technique that reduces data redundancy and eliminates undesirable characteristics like Insertion, Update and Deletion Anomalies. Normalization rules divides larger tables into smaller tables and links them using relationships. The purpose of Normalisation in DBMS is to eliminate redundant (repetitive) data and ensure data is stored logically.

Normalization Rule: Normalization rules can be divided into following category

- First Normal Form:
  - It should only have single(atomic) valued attributes/columns.
  - Values stored in a column should be of the same domain
  - All the columns in a table should have unique names.
  - And the order in which data is stored does not matter.
- Second Normal Form:
  - It should be in the First Normal form.
  - it should not have Partial Dependency.
- Third Normal Form:
  - It should be in the Second Normal form.
  - it should not have Transitive Dependency.

6.1 Normalizing USER table

Relational Schema : $\{ID, Name, Email, password, dept\}$

FD : $\{ID- > Name, ID- > Email, ID- > Password, ID- > dept, Email- > Name, Email- > Id,\ Email- > Password, Email- > dept\}$

$\{Email\}^+ = \{ID, Name, Email, password, dept\}$

$\{ID\}^+ = \{ID, Name, Email, password, dept\}$

{Email} can Identify all attributes of the table, so this is super key for sure. Again as it contains a single attribute, this is the candidate key. Similarly, ID is also a candidate key. We can easily show that there is only two

candidate key in this table.
prime attributes:{ID,Email}
non-prime attributes:{Name, Password,dept}

1NF: table contains only atomic value.so it is in 1NF
2NF:No partial dependency exist.Because proper subset of candidate key doesn't exist.
3NF: No transitive dependency exist.Because for each non-trivial functional dependency left-hand side is the super key or the right-hand side is prime attribute. so the table is in 3NF.

6.2 Normalizing Study_Leave_Application table

Relational schema: $\{Leave\_ID, IDApplicant, Name\_of\_Program, Destination, Destination\_$
$Program\_Start_Date, Leave\_Start\_Date, Finantial\_Source, Duration\}$

FD : $\{Leave\_ID->ApplicantID, Leave\_ID->Name\_of\_Program, Leave\_ID->$
$Destination, Leave\_ID->Destination\_Dept, Leave\_ID->Program\_Start\_Date, Leave\_ID-$
$Leave\_Start\_Date, Leave\_ID->Finantial\_Source, Leave\_ID->Duration\}$

$\{Leave\_ID\}^+ = Leave_ID, ApplicantID, Name\_of\_Program, Destination,$
$Destination\_Dept, Program\_Start\_Date, Leave\_Start\_Date, Finantial\_Source, Duration\}$
{Leave_ID} can identify all attributes of this table. As it contains a single attribute.It is the candidate key.

prime attribute:{Leave_ID}
nonprime attributes:{ApplicantID, Name_of_Program, Destination, Destination_Dept, Program_Start_Date, Leave_Start_Date, Finantial_Source , Duration}

1NF: table contains only atomic value.so it is in 1NF
2NF:No partial dependency exist.Because proper subset of candidate key doesn't exist.
3NF: No transitive dependency exist.Because for each non-trivial functional dependency left-hand side is the super key or the right-hand side is prime attribute. so the table is in 3NF.

6.3 Normalizing applicant table

Relational schema:$\{ID->user, Designation\}$
FD:$\{ID->Designation\}$

$$\{\text{ID}\}^+ : \{ID, Designaton\}$$

ID is the only key in this schema.so it is the candidate key.
prime attribute:{ID}
non-prime attributes:{Designation}
1NF: table contains only atomic value.so it is in 1NF.
2NF:No partial dependency exist.Because proper subset of candidate key doesn't exist.
3NF: No transitive dependency exist.Because for each non-trivial functional dependency left-hand side is the super key or the right-hand side is prime attribute. so the table is in 3NF.

6.4 Normalizing Evaluator table

Relational schema:$\{ID- > user, Working\_Role\}$
FD:$\{ID- > Working\_Role\}$

$$\{\text{ID}\}^+ : \{ID, Working\_Role\}$$

ID is the only key in this schema.so it is the candidate key.
prime attribute:{ID}
non-prime attributes:{Working_Role}
1NF: table contains only atomic value. so it is in 1NF.
2NF:No partial dependency exist.Because proper subset of candidate key doesn't exist.
3NF: No transitive dependency exist.Because for each non-trivial functional dependency left-hand side is the super key or the right-hand side is prime attribute. so the table is in 3NF.

6.5 Normalizing Evaluates table

Relational Schema:$\{Leave\_ID- > Study\_Leave\_ApplicatioID > Evaluator$
, status, evaluation_time, comment}
FD: $\{\{Leave\_ID, EvaluatorID\}- > status, \{Leave\_ID, EvaluatorID\}- > comments, \{Leave\_ID, EvaluatorID\}- > Evaluation\_time\}$

$$\{Leave\_ID, Evaluator\_ID\}^+ = \{Leave\_ID, EvaluatorID, status, Evaluation\_time, commen$$

Leave_ID and Evaluator_ID can uniquely identify all attributes of the table. We can proof that Leave_ID and Evaluator_ID only set of attributes that can uniquely identify our table. Since we cannot discard at least one attribute from this set,This is the only candidate key.

prime attributes:{Leave_ID,Evaluator_ID}

non-prime attributes:{comment,status,Evaluation_time}

1NF: table contains only atomic value. so it is in 1NF.

2NF:No partial dependency exist.Because a proper subset of the candidate key cannot determine any other non-prime attribute.

3NF: No transitive dependency exist.Because for each non-trivial functional dependency left-hand side is the super key or the right-hand side is prime attribute. so the table is in 3NF.

7 System Architecture

Describe the architecture of your system using a figure: Describe how each component of the architecture communicate.

8 Implementation

Give some code snippet of each component you outlined in your System architecture. Some DDL query example. Use the listing environment for writing code. Listing 1 shows an SQL query.

```sql
select distinct name
from instructor
where salary > some( select salary
                        from instructor
                        where dept_name='CSE');
```

Listing 1: A SQL query example

9 Validation

Show that users are satisfied with your product. You can also give a user manual here describing how to use your system (process of completion of different tasks using your system) You can use some matrices (time, cost, resource etc.) to compare your system with the previous system.

10 Software Deployment

Describe how to install and configure your system so that a non-technical user can use your system.

11 Conclusion and Future Work

Write the conclusion of your project: what was the problem? what the developed solution offers, Significance of the project, limitations of the project and future work.

12 Bibliography

To add bibliography in your document, use the following steps:

1. First create a .bib file in the same directory where your .tex file is (in our case, the file name is references.bib). Also place the bibliography style file in the same directory. In our case, we are using the ios1.bst style file. We include the following commands in the .tex file for the style file and bib file:
   `\bibliographystyle{ios1}`
   `\bibliography{references}`

2. Import the BibTeX of your book or paper from Google Scholar or other sources into your .bib file. An example of BibTex is shown in Listings 2.

```
@article{kopka1995guide,
  title={A Guide to $\{$$\backslash$LaTeX$\}$--
    Document},
  author={Kopka, H and Daly, PW},
  year={1995},
  publisher={Citeseer}
}
```

Listing 2: A BibTeX example

3. Then, use the name of the BibTex (in Listing 2, the name is kopka1995guide) in the text of your .tex document where you want to refer it.

4. After saving your .tex document, execute the PDFLaTeX option one time; then execute the BibTeX option; then again execute the PDFLaTeX option for twice; finally, execute the QuickBuild option. Now your document refer the corresponding book or paper.

References

[1] F. Paetsch, A. Eberlein and F. Maurer, Requirements engineering and agile software development, in: *WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003.*, IEEE, 2003, pp. 308–313.

[2] S.S. Paradkar, A framework for modeling non-functional requirements for business-critical systems, *International Journal of Innovative Research in Computer Science & Technology (IJIRCST) ISSN* (2021), 2347–5552.