VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belgavi – 590014

A Project Report
on

"ACADEMIC PROGRESS SCORE"

Submitted in partial fulfillment of the requirement for the award of

BACHELOR OF ENGINEERING
in
INFORMATION SCIENCE & ENGINEERING
By

HARSHITH S **1AP18IS008**

Under the guidance of
Prof. Pallavi H B, Asst. Professor

2021-22

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

A.P.S. COLLEGE OF ENGINEERING

Anantha Gnana Gangothri Campus, Somanahalli
NH-209, Kanakapura Road, Bangalore-560082

A.P.S. COLLEGE OF ENGINEERING

(Affiliated to Visvesvaraya Technological University)
Anantha Gnana Gangothri,
NH-209, Kanakapura Road, Bangalore-560082

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

Certificate

This is to certify that the project work entitled

"ACADEMIC PROGRESS SCORE"

Is a bonafide work carried out by

HARSHITH S **1AP18IS008**

In partial fulfillment for the award of the degree of ***Bachelor of Engineering*** *in* ***Information Science and Engineering*** *of the* ***Visvesvaraya Technological University****, Belgaum during the year* ***2021-2022****. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The Project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the Bachelor of Engineering Degree.*

Prof. Pallavi H B Asst. Professor	**Dr. Kumar B I D** Professor & HOD	**Dr. A G Nataraj** Principal

External Viva

Name of the Examiners	**Signature with Date**
1.	
2.	

DECLARATION

I hereby declare that the entire project work entitled **"Academic Progress Score"** done at **A.P.S College of Engineering, Bangalore-82,** submitted to **Visvesvaraya Technological University,** in partial fulfillment of requirements for the degree of **Bachelor of Engineering** in **Information Science & Engineering** is a record of original work done by me and no part of it has been submitted for any degree or diploma of any institution previously.

Place: Bangalore **Signature of the Student**

Date: **HARSHITH S** **1AP18IS008**

ACKNOWLEDGEMENT

Salutations to our beloved and highly esteemed institute **"APS College of Engineering Bangalore-82"**, for having well qualified staff and labs furnished with necessary equipment and computers for growing up us to become true graduates.

I express my sincere gratitude to **Dr. A. G. Nataraj**, Principal, and to **Dr. Kumar B I D**, Professor & Head of the Department of Information Science & Engineering, APS College of Engineering, for their help and encouragement.

I would like to express my immense gratitude to my guide **Prof. Pallavi H B,** Assistant Professor, Dept. of ISE, for her guidance and assistance throughout this project.

I would also like to thank **Prof. Nandeeswar S B,** Associate Professor & Project Coordinator, Dept. of ISE, for his assistance and encouragement.

I take this opportunity to express my gratitude to all those individuals who have helped me in this project. In few words I cannot express adequately my indebtedness to all my friends. Their constant encouragement and cooperation have been the essential source of inspiration to me.

Last, but not the least, my heart full thanks to my parents for unrelenting support and being there always with me forever cooperation and goodwill.

HARSHITH S **1AP18IS008**

ABSTRACT

Academic Progress Score is an android application that help students to give their honest feedback directly to the faculty. This application is user friendly, effective and secured rating system for all the students, teaching faculty and even for the institution. Students can view the respective teaching faculty profile and provide the feedback score according to each parameter specified. This feedback score is considered for calculating the faculty's academic progress score. Student need not login to the application to provide the feedback score. Students are provided with list of the faculty and students need to select the faculty which they want to provide the feedback.

Each faculty can use valid credentials to login to the system and view their respective academic progress score along with their qualification profile. Faculty can utilize this feedback score to improve their teaching, skills and for overall development. The institutions can use the application to view academic progress score of faculties and improve the facilities provided by them. This application helps to better the understandingand relationship between the faculty, students and the institution. It provides a way for students to keep up their opinion in front of the institution and using this review of students the institution can adopt the various methods to improve the quality of the education. This application helps the faculty and institution to improve.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

CHAPTER 1

INTRODUCTION

Academic Progress Score is an application that provides students a platform to provide feedback directly to the faculty using their mobile phones. This application is designed in such a way it will be a user friendly, effective and secured rating system for all the students, teaching faculty and even for the institution. This application contains two main login pages- admin login page and faculty login page. Only an authorized admin can login to the system to make changes or update the faculty profile.

Students can view the respective teaching faculty profile and provide the feedback score according to each parameter specified. This feedback score is considered for calculating the faculty's academic progress score. Student need not login to the application to provide the feedback score. Students are provided with list of the faculty and students need to select the faculty which they want to provide the feedback.

Each faculty can use valid credentials to login to the system and view their respective academic progress score along with their qualification profile. Faculty can utilize this feedback score to improve their teaching, skills and for overall development.

The academic progress score is calculated by the application considering both the feedback given by the students about each faculty and achievements, qualification and results achieved by the faculty.

The institutions can use the application to view academic progress score of faculties and improve the facilities provided by them. This application helps to better the understanding and relationship between the faculty, students and the institution. It provides a way for students to keep up their opinion in front of the institution and using this review of students the institution can adopt the various methods to improve the quality of the education. This application helps the faculty and institution to improve.

1.1 FEEDBACK SYSTEM

Feedback is a reaction or information that occurs as a result of actions or behavior under taken by an individual or group. In a Learning & Development context, both positive and negative feedback is crucial. Feedback provides a sense of engagement and interactivity, and allows learners to take ownership of their learning. Effective feedback shows learners their current level of performance, and lets them know what they need to do to reach a higher level. Feedback occurs when outputs of a system are routed back as inputs as part of a chain of cause-and-effect that forms a circuit or loop. The system can then be said to feedback into itself. The notion of cause-and-effect has to be handled carefully when applied to feedback systems.

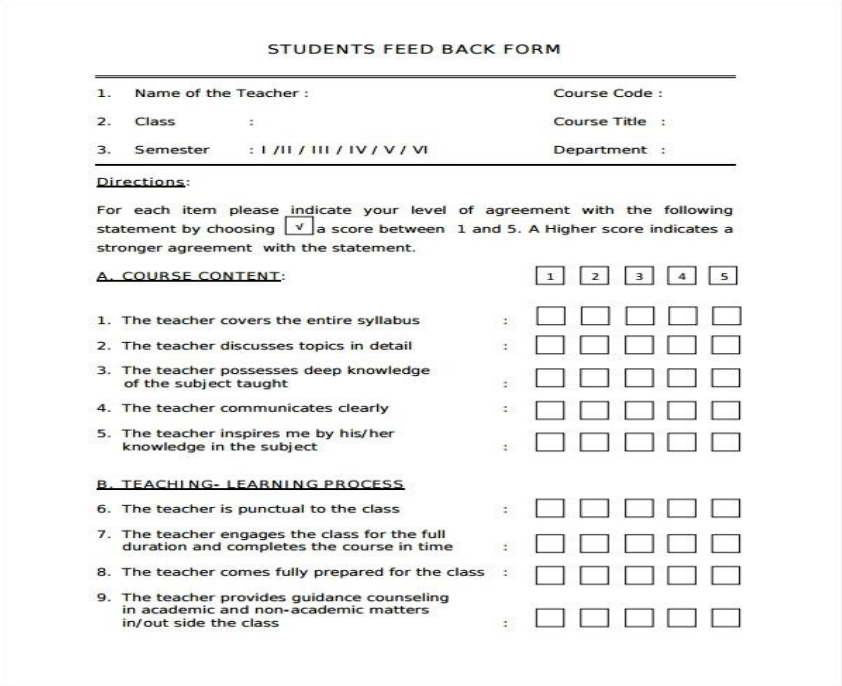
1.1.1 STUDENT FEEDBACK SYSTEM

Student feedback is usually collected through both formal mechanisms (for example, awritten questionnaire, student participation on committees) and informal means. One of the challenges to both structured and unstructured feedback is the impact of cultural and societal norms. Institutions are expected to ensure that their student feedback processes are understood by the students and are appropriate for the students and the institution. This means that institutions formulate their own coherent approach to student feedback, which is clearly understood and accepted by students, academic staff and their managers. Above all, the processes for collecting feedback from students are expected to be robust.

The Online Student Feedback System is an automatic feedback generation system that provides the proper feedback to the lecturers. In the existing system students can give feedback about the lecturers by doing manually. By this process student can give feedback in online system without wasting his time in writing. After giving feedback by every student papers are collected by the faculty and calculated the overall percentage for each subject and each lecturer. After that those all-percentage results is viewed by the faculty. Hence estimating the performance of lecturers and giving feedback to college staff. So, the existing system carries more time to do a piece of work for this reason the online system feedback is implemented. This is the main disadvantage of the existing system for giving feedback about the lecturers and viewing report of lecturers manually. Student feedback on courses is an essential element in quality assurance. The aim of this is to save time for staff in academic departments.

1.2 EXISTING FEEDBACK SYSTEM

In the present system, the student feedback is collected by the institution through pen and paper mode in a classroom. All feedback is collected through printed feedback forms which are manually filled by the students for each course attended by the student. The other way of collecting student feedback in online mode is through Google forms as shown in the fig 1.1.



STUDENTS FEED BACK FORM

1. Name of the Teacher : Course Code :
2. Class : Course Title :
3. Semester : I /II / III / IV / V / VI Department :

Directions:

For each item please indicate your level of agreement with the following statement by choosing [√] a score between 1 and 5. A Higher score indicates a stronger agreement with the statement.

A. COURSE CONTENT:	1	2	3	4	5
1. The teacher covers the entire syllabus :	☐	☐	☐	☐	☐
2. The teacher discusses topics in detail :	☐	☐	☐	☐	☐
3. The teacher possesses deep knowledge of the subject taught :	☐	☐	☐	☐	☐
4. The teacher communicates clearly :	☐	☐	☐	☐	☐
5. The teacher inspires me by his/her knowledge in the subject :	☐	☐	☐	☐	☐
B. TEACHING- LEARNING PROCESS					
6. The teacher is punctual to the class :	☐	☐	☐	☐	☐
7. The teacher engages the class for the full duration and completes the course in time :	☐	☐	☐	☐	☐
8. The teacher comes fully prepared for the class :	☐	☐	☐	☐	☐
9. The teacher provides guidance counseling in academic and non-academic matters in/out side the class :	☐	☐	☐	☐	☐

Fig.1.1: Student Feedback Form

1.3 PROBLEM IDENTIFICATION

- In the present system, the student feedback is collected by the institution through pen and paper mode in a classroom.
- In this case, it is difficult for the students to give an honest and unbiased feedback for each teaching faculty.
- This method is time consuming, and it also reveals the identity of the students who gave the feedback.
- Here the feedback data can be lost, damaged or be mishandled. There is no structured designed to store and manage all the student feedback data in one place in an effective manner.
- The data cannot be reused or accessed by the institution in order to improve student and management relationship.

1.4 OBJECTIVE OF THE PROJECT

- The main objective is to manage and displays the faculty profile along with the score calculated through the student feedback.
- To collect feedback in digital form from the students and entered the system by the authorized admin.
- To store data in the database, which enables admin to retrieve, update, delete or insert faculty. These data can be accessed and viewed anywhere at any time.
- The system can be used by the faculty also and they can view their respective score through this application.
- To provide a secure environment to maintain and display the score and profile of the faculty.

1.5 THE PROPOSED PROJECT

- "Academic Progress Score" allows the institution to provide a secure environment to maintain and display the score and profile of the faculty. Overall proposed project architecture is shown in the fig 1.2.
- This system contains two main pages, faculty login and admin login.
- Only an authorized admin can enter the score of the faculty according to the student feedback.
- The entered score will be reflected in the application along with the faculty details as uploaded by the admin.
- The faculties can also view their profile and score given by the students by entering a valid login detail in the system.

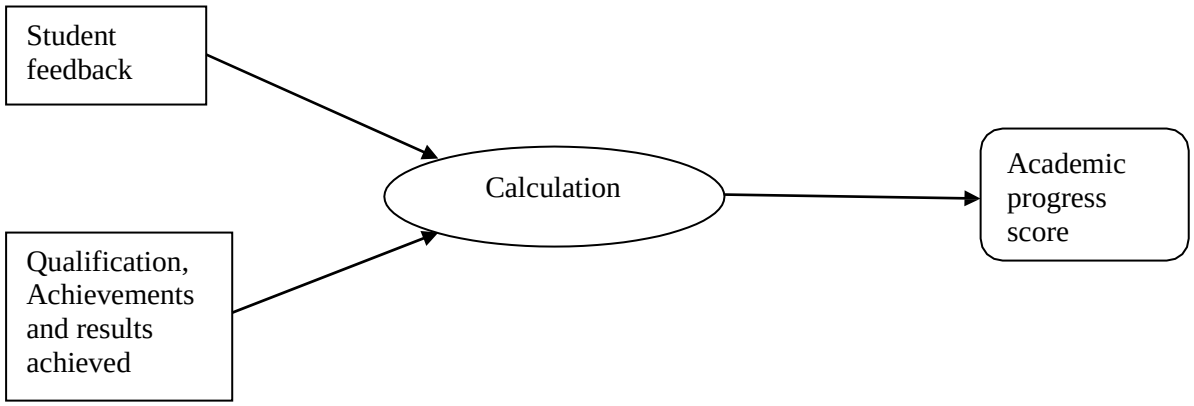


Fig.1.2: Proposed System Architecture

1.6 ORGANIZATION OF THE REPORT

- Chapter-1: **Introduction-** This chapter talks about the introduction to online social network, problem statement, Aim of the project and motivation factors.
- Chapter-2: **Literature Survey-** This chapter majorly deals with existing systems and proposed systems and related research works.
- Chapter-3: **Software Requirement Specification-** This chapter speaks about the product perspective, user characteristics, its assumptions and dependencies, specific requirements, non- functional requirements and functional requirements.
- Chapter-4: **System Design-** This chapter deals with the advanced software engineering
  wheretheentireflowoftheprojectisrepresentedbyaprofessionaldataflowdiagram starting from level 0 till 3. This chapter mainly deals with use case diagram for the project representation
- Chapter-5: **Implementation-** This chapter deals with the steps involved in the creation of the project work. It is defined with the assistant of code explanation for the ease of reader.
- Chapter-6: **Testing-** This chapter mainly deals with the various types of the test cases to prove the validity of the project.
- Chapter-7: **Snapshots-** This chapter mainly deals with the output of the application.
- Chapter-8: **Conclusion and Future Work-** This chapter is mainly the summary of the entire project development and it also suggests some of the enhancement ideas which couldn't be covered up due to constraint of time and resources.
- **References-** This section mainly highlights all the journal and IEEE papers being referred for the development of the project.

LITERATURE SURVEY

Chapter 2

LITERATURE SURVEY

2.1 EFFECT OF STUDENT FEEDBACK ON MOTIVATION OF INDIAN UNIVERSITY TEACHERS

The purpose of the study is to measure the motivation of the teachers of higher education towards students' feedback policy of National Assessment and Accreditation Council (NAAC) established by Internal Quality Assurance Cell (IQAC) for different Universities. By the help of questionnaires, the data were gathered, which were earlier sent to the participants, via e-mail for deep analysis and it was directed towards broad generalization of the population. One hundred University teachers were randomly selected as the sample from twelve departments of a university. These departments were: Ecology and Environmental science, Life Science, Chemistry, Physics, Mass Communication, Fine Arts, English, Sanskrit, Bengali, Commerce, Education, Social work. Teachers' Motivation Towards Student Feedback Inventory (TMTSFI) was administered among the samples and all the data sheets were scored and analyzed. It was found the overall motivation towards student feedback was strongly positive among university teachers and it significantly affected the teachers' classroom performance.

Now different countries of the world have initiated quality assurance mechanism of higher education, and more are in the process of developing the quality assurance strategy. Both at national and international levels, different countries are trying to provide quality education to the students (Stella, 2002). Recently, parents, educators, employers and the entire societyhave expressed serious concern about the quality of education. Especially in the university, students are blaming the poor quality of instruction and teachers' lack of interest and attitude towards teaching profession. Possibly, there is somewhere problem in the instruction, and teachers' motivation towards teaching learning process (Schmelkin, Spencer & Gellman, 1997). At present primary, secondary school, colleges and even in university levels, students are blaming the teachers (Shannon, Twale, & Hancock, 1996). They have doubts about the knowledge base, communication skills, sincerity and commitment, interest generation, abilityto integrate content, accessibility both in and out of the classroom, ability to design test and their provision to give feedback and sufficient time to their students.

It is in debate and if they do so; what extent, and how frequently they are giving feedback to their students, how much depth of knowledge is within them, how many times they maintain sincerity are the questions (Renaud, & Murray, 2005). Quality education to the students, is now a day a, challenge (McDonald, 2001). University Grants Commission (UGC) has given an opportunity to the students to evaluate their own teachers through the student feedback system. The table 2.1 gives the description about the teachers motivation towards student feedback in percentage.

Variables:	Teachers' motivation			
	Strongly Agree(S+)	Agree(+)	Disagree(-)	Strongly Disagree(S-)
Teachers' depth of content knowledge	50(50%)	20(20%)	5(5%)	25(25%)
Teacher's sincerity and commitment	15(15%)	45(45%)	10(10%)	30(30%)
Way of interest, motivation, Creation among students towards teaching learning process	10(10%)	10(10%)	40(12%)	40(40%)
Teacher's mood of integration of course material and learning environment	05(05%)	05 (05%)	40(40%)	50(50%)
Teacher's accessibility to the student and giving sufficient time for feedback to the student	40(40%)	35(35%)	15(15%)	10(10%)
Evaluate students level of understanding	40(40%)	25(25%)	10(10%)	25(25%)
Teacher's manual skill and analytical ability	45(45%)	20(20%)	15(15%)	20(20%)

Table.2.1: Teachers Motivation Towards Student Feedback in Percentage

Therefore, National Assessment and Accreditation Council (NAAC) has established the IQAC (Internal Quality Assurance Cell) in every university and colleges to ensure the quality of teachers i.e., API (Academic performance Index). IQAC is responsible to conduct time to time student feedback i.e., twice in a year Wegener (1995) stated that well over 90% of universities currently use student feedback to assess the teaching staff. Still this system is in question and often becomes an issue of debate.

The Indian Higher education system is one of the largest in the world. The deterioration inthe quality of higher education is a serious anxiety for all stakeholders. In 1994, the nationwide NAAC was established to ensure and enhance the quality of Indian higher education. However, continuous feedback from the beneficiaries of education determines the effectiveness of the council.

The literature found, in the last 10 years of its reality, that NAAC has earned a lot of generosity and admiration from the learned community (Pillai & Srinivas, 2006). Internal IQAC has established much strategy to consider the quality of the educators in higher education institution. Amidst these schemes, scholars' response is the recent issue in India. Dialogue with numerous universities found that student feedback has a negative impact on the educator's presentation and motivation and mind-set (Mizoram University 2007; MANUU, 2010). Nevertheless, Assam University & Tezpur Universities, scholars have an affirmative response and mind-set towards their response to the teachers. Martin and Rich (1971) investigated on school motivation and attitude in the direction of scholar evaluation, and discovered student feedback help for advancement to the enhancement of teaching learning method.

The abilities of different university have a favorable disposition in the direction of the use of student feedback and the teacher's feedback (Rich, 1976). Ryan, Anderson and Birchler (1980) found that introduction of mandatory use of student feedback directs to an important reduction of school lesson and job Satisfaction. Student feedback is a helpful method and it is a valid and dependable method to identify distinct issues in teaching-learning process (Ballantyne, 1999). However, Avi-Itzhak and Kremer (1986) found that older teachers strongly fight against the use of student feedback and it had no worth for school improvement.

But, Gillmore (1984) found that student response was significant and positive and if administered to the direction, aided the teachers to present better. Guichon, Betrancourt & Prié, (2012) investigated and found that most of the faculty did not seem to think that students ranking them have a contradictory effect on school moral and believe and they also examined that student feedback of instruction tend to have a more affirmative attitude about their use of direction. Gordon (1990) argued student feedback for educators is not a good practice.

The author also agreed that students are unable to assess the deepness of a teacher's knowledge and it should play an awful impact on the rapport between teachers and scholar. Tang, Jinlan; Harrison, Colin (2011) discovered that 23% of school answered the review, and the outcome is that scholar evaluation alters the production habit and the assignment of teachers. Adams 1997; Nasser & Fresko (2002) argued that student feedback to their teacher is an awful custom in higher education because learners have no dependable attitude and motivation in the direction of their teachers.

Significance of the Study

The significance of the study has some parameters as described in the below table 2.2. The present study was undertaken to determine the motivation of teachers after the student feedback on the following head:

- Teacher's depth of content knowledge
- Teacher's sincerity
- Teacher's way of interest, motivation
- Teacher's mood of integration of course material and learning environment
- Teacher's accessibility to the student and giving sufficient time for feedback to the student
- Teacher's assignment and evaluation to know the students' level of understanding and analytical ability
- Teacher's skill and analytical ability

Sl	Parameters		Sum of squares	df	Mean square	F	p
1	Teachers' depth of content knowledge	Between Groups	108.68	11	9.88	3.68	P<.01
		Within Groups	236.22	88	2.68		
		Total	344.91	99			
2	Teachers' sincerity and commitment	Between Groups	27.35	11	2.48	1.97	P<.01
		Within Groups	110.64	88	1.25		
		Total	138.00	99			
3	Interest and motivation creation among students	Between Groups	19.72	11	1.79	1.22	p>.01
		Within Groups	128.38	88	1.45		
		Total	148.11	99			
4	Integration of course material with learning environment	Between Groups	26.10	11	2.37	2.78	P>.01
		Within Groups	74.94	88	.85		
		Total	101.04	99			
5	Accessibility to the students at both inside and outside of the department	Between Groups	101.07	11	9.18	4.38	P<.01
		Within Groups	184.23	88	2.09		
		Total	285.31	99			
6	Assignment to evaluate student's level of understanding	Between Groups	5.70	11	.51	.68	P<.01
		Within Groups	66.48	88	.75		
		Total	72.19	99			
7	Teachers' manual skill and analytical ability	Between Groups	25.69	11	2.33	2.30	p>.01
		Within Groups	89.21	88	1.01		
		Total	114.91	99			

Table.2.2: ANOVA of Pre-Test Post-Test Score of Teachers MotivationTowards Student Feedback Policy of IQAC

Teacher’s overall degree of motivation towards student feedback was strongly positive and Martin, and Rich (1971) supported this result. It was also found that there existed a significant difference among the motivation of teachers after the students' feedback on the teacher's depth of knowledge. The same result was concluded by (Haskell, 1997; Scheeler, McKinnon, Kathleen & Stout 2012). There existed a significant difference in the degree of motivation of teachers towards students' feedback on teacher's sincerity and commitment to the teaching learning process (Chen & Hoshower, 2003).

The study conducted by Ballentyne, 1999; Nasser and Fresko, 2002 earlier resulted that faculty attitude and their carrier increases by their sincerity and their commitment and this was the literature in support to the above result of the recent study. The present study also resulted that there existed no significant difference in the motivation of teachers towards the student feedback on the teacher's way of interest and motivation creation for the students learning process and the similar result was found by Tang & Harrison, 2011.

Studies conducted by Haskell (1997) found that student feedback helps the teachers to integrate the learning material in the classroom. The same finding also has been drawn from the present study that there existed no significant difference in the motivation of teachers towards the students' feedback on the teacher's mode of integration of course material with the learning environment.

It was also found that there existed a significant difference among the motivation of teachers towards the student feedback on teacher's accessibility to the students at both inside and outside of the department. In support of this result there was no study conducted both in India and abroad. That is why the result needs further investigation. The researchers found that there existed a significant difference among the motivation of teachers after the students' feedback on teacher's assignment to know the level of understanding.

Moreover, there existed no significant difference among the University teachers towards the student feedback on teacher's manual skill and analytical ability. Similar to the above result, there was no significant evidence. It was recommended to study whether the students are able to evaluate their teachers' depth of knowledge or not. If so, a comparative study is necessary to study the relationship between sincerity and commitment of the teacher after student feedback.

2.2 IDENTIFY THE INFLUENCE OF VARIOUS FACTOR OF APPS ON THE GOOGLE PLAY APPS RATINGS

Google Play Store contains millions of Apps. These apps are downloaded and been used by millions and billions of users. Whenever a user browses or search for apps on Play store, a list of apps is shown to the user in which each app contains the app name along with its rating. Usually, the user prefers to download highly rated apps because highly rated apps reflect users' satisfaction. In order to gain high ratings, app developer uses different techniques and tweaks other than the app quality its self. Developers use attractive app titles, demanding icons, and other things to gain better ratings for their apps. However, there is no scientific approach to find the real impact of using attractive titles or any other such thing in order to gain higher ratings. Therefore, in this paper, we examine a number of factors of google play store apps and identify the influence of these factors using variable importance. For this purpose, real-world Google Play store apps dataset is used in this paper to identify the importance of these factors. For identification of important variables, Random Forest, Linear Regression Model and Support Vector Regression are used. The performance of the model is evaluated using standard performance evaluation techniques. The results show that some factors have higher significance and influence the app ratings. Moreover, keyword analysis has taken place to find the important words used in app title those results in higher and lower ratings.

The mobile apps are continually becoming more and more popular. Mobile app stores such as Google Play store contains millions of apps. Among those apps, a number of apps have billions of downloads and active users. These apps are related to different categories including games, communication, books, business, news, sports, and many others. Each category contains a huge number of apps. In Google Play store, each app has usually been identified by its name and its rating. The app rating is the average rating of all the ratings given by the users. Before downloading an app, people usually prefer to download apps with high ratings because high rating apps usually have higher quality than the rest of the apps. In this regard, Hsu and Lin (2015) performed a detailed analysis of the intentions of mobile phone users. This study shows that app ratings have a valuable effect on the user's intention to download and use an app. Therefore, in order to gain better ratings, companies use different techniques to increase the ratings of their apps on the Google Play store. For this purpose, many companies even use fraudulent and deceptive activities to gain more ratings for their apps, so their apps can be more visible in the play store (Zhu et al. 2015).

However, to the best of our knowledge, there is no scientific approach to find a relationship between ratings and other factors of the app. In order to fill the research gap, it is of great importance to analyze the various factors of Play store apps and find out if there is any real connection between these factors and the app rating. For this purpose, it is worthy to test these factors of Play store apps by the means of Machine Learning and data analysis. This will help finding a real connection between these factors and apps rating. In this paper, our sole objective is to identify the influence of various factors of Google Play store apps on the apps rating. Although, some researchers have worked on different app variables and their relationship with the app rating, none of the researcher focused on the variables proposed in this research. Also, in this research different linear and non-linear regression models are used along with the standard evaluation techniques for performance evaluation as shown table 2.3.

Variable name	%IncMSE	IncNodePurity
category	41.07	136.86
no_of_reviews	54.1	393.37
app_size	37.47	183.56
no_of_installs	30.74	161.6
type_of_app	27.21	22.09
content_rating	32.22	37.75
genre	43.72	155.99
android_version	26.12	143.67
word_count_in_name	31.27	98.49
character_count_in_name	38.11	204.86
Symbol_Count_In_ Name	24.19	29.69
category_related	19.39	13.81
free_in_title	9.65	1.95
genre_related	16.33	11.73
digits_in_title	12.29	18.73
year_in_title	4.94	5.2

Table.2.3: Random Forest Regression Results

Moreover, this research focuses on detailed keyword analysis that finds the important keywords that are helpful in gaining better ratings and vice versa. For this purpose, we take real-world data of Google Play store apps and use various Machine Learning models to find a real relationship between these variables and app ratings.

We also perform detailed statistical analysis in terms of app categories, app names, keywords, app size, number of installs, app types, content ratings, etc. with the overall app rating to find out a correlation between these factors and their value in terms of app ratings. The below figure 2.1 shows the variable identification and computations.

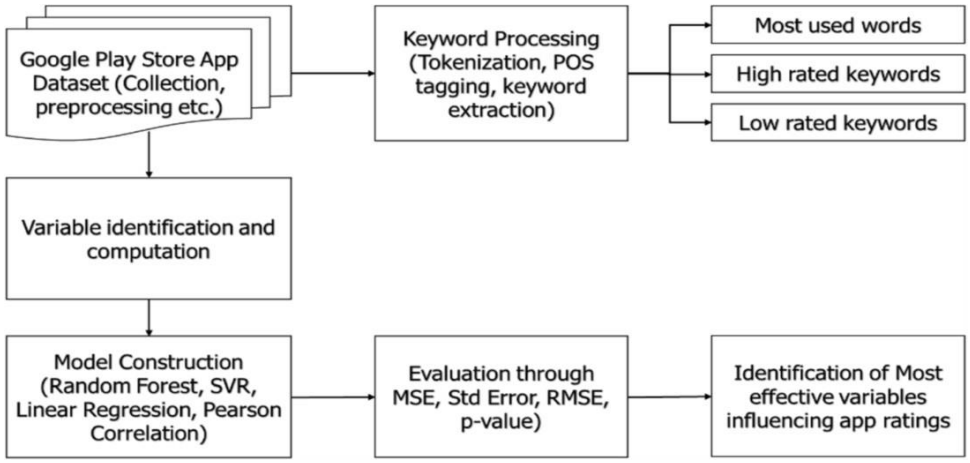


Fig.2.1: Variable Identification and Computations

After computing all the variables and applying the models, the results are computed from different models. Each model has its own significance and importance. Random forest regression and variable importance is the most commonly used regression models to find the important variables in a dataset. The results of random forest regression are shown in Table:2.3.

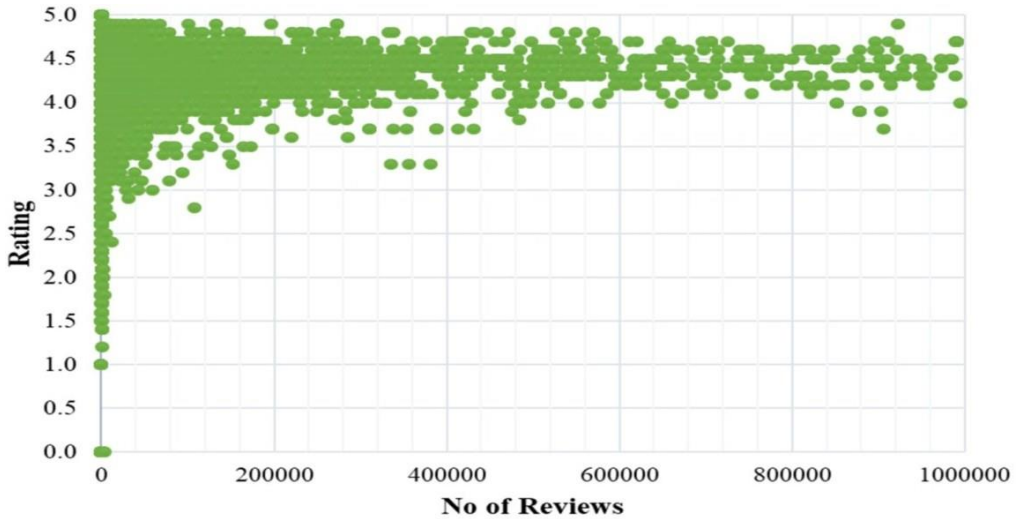


Fig.2.2: Scatter Chart of Apps Rating for Different Number of Reviews of Apps

The Results show that no_of_reviews, genre, character_count_in_name, app_size are themost influential variables and have high impact on the ratings of the apps. Similarly, year_in_title, free_in_title, digits_in_title have little importance in terms of predicting the ratings of an app. In order to analyze the importance of a number of reviews on the app ratings, a scatter plot of number of reviews against app rating is shown in Fig. 2.2.

The graph clearly shows that apps with higher number of reviews usually have higher ratings. These are usually the apps which have a very high number of installs and are owned by bigger companies who try to improve upon their apps. These apps usually have higher ratings. Similarly, some genres are ranked higher and are highly liked by the users. Therefore, apps in those ranks are highly rated by the users. On the other hand, some genres are highly criticized by the users, users expect much more from the app, or the users are highly diverse from different backgrounds which lead in mixed or lower sort of ratings. Other important attribute is character_count_in_name which clearly shows that character count really matters in the app.

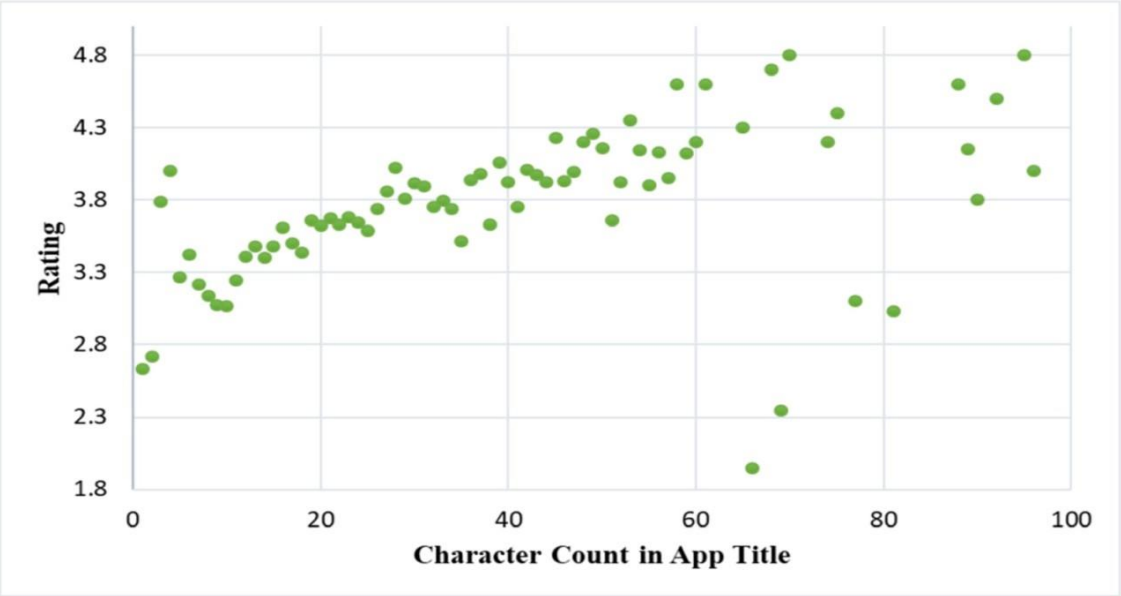


Fig.2.3: Scatter Chart of Apps Rating for Different Character Counts in App Title

In order to show the better demonstration of character counts with mean app rating, a scatter graph is shown in Fig. 2.3. The Figure shows that for most of the times, when the character count in the App title is low, the ratings are usually low and when the Character counts are higher the ratings are higher. Although there is no perfect correlation between these two attributes, but the correlation is present.

2.3 FEATURE-LEVEL THE RATING SYSTEM USING CUSTOMER REVIEWS AND REVIEW VOTES

With the rise of the Internet and the kind of busy lifestyles people have today, online shopping has become a norm. Customers often rely on the online ratings of the previous customers to make their decisions. However, most of these ratings on the online websites are product-level ratings and lack specificity. Although products can be compared based on the product-level ratings available, there is always a class of people who prefer buying the items based on particular features. Such people have to generally go through the entire comments section to know previous customers' perceptions, of the product's features in which they are interested. Considering the number of products present for an item (such as mobile), it becomes a tedious job for a customer to arrive at the best product for himself. Moreover, from a manufacturer's perspective, such product-level ratings hardly specify what is good or bad about the product. Thus, if feature-level ratings are available, it gives more clarity to the manufacturer on how to improve the product. Given all these benefits, our goal is to develop a feature-level rating system. Although feature-level ratings can also be requested from the customers just like the product-level ratings, it is not a good proposal, for there could be too many features. Instead, it is much more practical to leverage whatever reviews and review votes that are being already given by customers to provide feature-level ratings. The reviews are made up of sentences, and every sentence has some sentiment associated with it, namely, positive, neutral, or negative. Also, since they can be separated, we can always extract the sentences describing a particular feature of the product and, subsequently, obtain sentiment scores over such sentences.

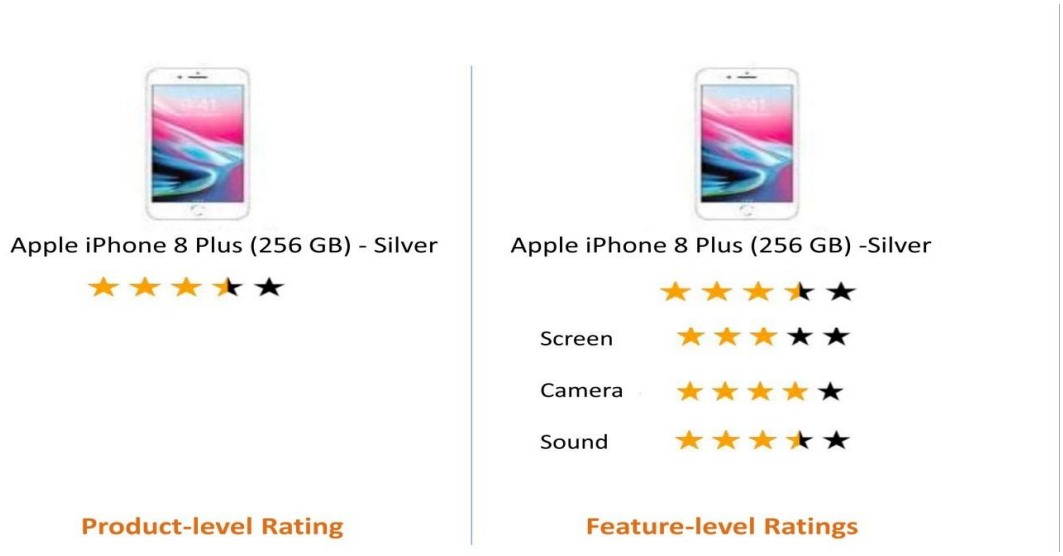


Fig.2.4: Product Level Rating v/s Feature Level Rating System

By utilizing these sentiment scores as the basis and the review votes as a support, we can build a feature-level rating system that can yield feature-level ratings, as shown in Fig. 2.4. However, there are few challenges in building such a feature-level rating system. First, we need to determine which features to look for in an item. Another challenge is that there could be many words relating to the same feature; they all need to be clubbed into one feature. Second, we have to preprocess the data as some of the review comments may contain non-English languages, one word, spelling mistakes, andso on. Third, we have to devise a way to transform the extracted sentiment scores into an appropriate rating for a feature of a product while incorporating the review votes. As far as feature identification of an item (such as mobile) is concerned, we go through the word frequency table of the entire customer review data for that item, at least up to a particular frequency. Next, all the related words coming under the same feature are grouped, and the most frequent one is chosen as a representative. We call such representatives as feature keywords. Then, we perform a series of preprocessing steps onthe customer review data to filter out the unnecessary data, correct the remaining, and turn it into structured data. Each review is broken into sentences, and only relevant sentences are retained. The relevant sentences are passed through sentiment analyzer to generate sentiment scores, which are then adjusted to the ratings. Scores within a particular range are given a specific rating. The ratings of the relevant sentences containing a particular feature are combined using the weighted-average to obtain the final ratings since all opinions are not equally valuable. We leverage review votes to assign the required weights. Our contributions are as follows. We develop a feature-level rating system that takes customer reviews and review votes as input and then outputs feature-level ratings. We obtain such ratings for as many as 4000+ mobiles sold online in terms of as many as 108 features. We propose vote-aware cumulative rating and vote-aware final rating measures, a new way of accumulating and finalizing the sentiment scores. Although there are no ground truths available, we still manage to evaluate our approach by comparing the final ratings of our phone feature against overall ratings of the phone given by the customers themselves, which leads to remarkable results, demonstrating the effectiveness of our method.

A. Feature Selection

Let us say that we collect a data set of N feature-related words, denoted by W = {w1,...,wN }, by manually going through word frequency table of the entire customer review data on an item (mobile, in our case). In this way, we identify the features in which people are generally interested. Note that we neglect the words having their frequency less than 0.02% of the total

number of reviews in the review data, which means they are rarely discussed feature-related words and can, therefore, be neglected. Let us say that the corresponding frequencies of the feature-related words form another set denoted as Z = {z1,...,zN }. Since the feature-related words related to a particular feature should be clubbed into one feature, we define a relationizer function denoted as R(W, wi), which returns a set of all the related words of wi in W, including itself.

We give the information regarding what all characters are retained. After that, we remove any entries that are left empty because we have no use of them in our feature-level rating system. Let us consider that, for a product (not an item), we denote product review data as D = {C1,C2,...,Cm}, comprising of m useful review comments.

Note that the relationizer function discussed here as a matter of notation is manual. We now define our feature data set, denoted by F, as a set of such distinct sets of related feature words, as defined in the following: F = {R(W, wi)}i∈{1,...,N} (1) where we iterate through all the words in W and form distinct sets of the related words using the relationizer function. Since different related words will form the same sets, the duplicates will be removed to make the left sets distinct. Now, let Fk be thekth feature words set in the F feature data set. In any Fk , a representative feature word is selected to identify the whole feature words set. Let us call such representatives as feature keywords. The most frequent feature word in the set is chosen as the representative or feature keyword to assign its nameto the set, as follows:

$$Fk \leftarrow wi|i = \max(\{Z(i)|wi \in Fk\})$$

Where the feature set is assigned a name with the word that has the maximum frequency in the set. From now on, abusing notations a bit, Fk can mean both the kth feature set (a set of related feature words) and its keyword (or set's name), according to the context.

B. Preprocessing

The review comments' data is generally unstructured, as it is written by the customers online. Our goal now is to convert this unstructured data into structured data in our preprocessing steps, which means that the useful data is extracted, disintegrated, and corrected. Note that the data retained after removing the characters that are useless for our purpose are what we mean as the useful data.

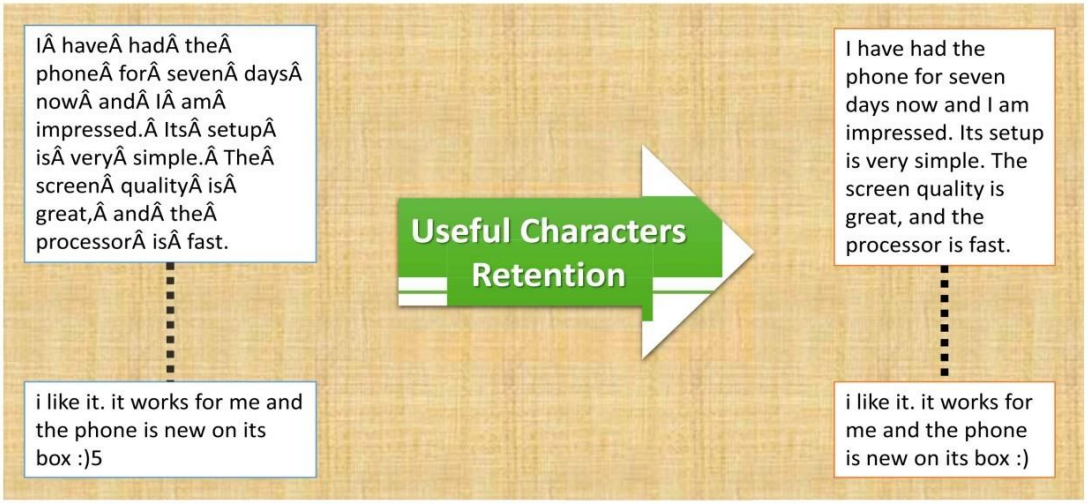


Fig.2.5: Preprocessing: Useful Characters Retention, where Unnecessary Characters are Removed in this Illustration.

While inspecting the reviews for figuring out the features to work with, we also noticed how people praise or criticize. People often use the characters required for adjective words, punctuation, or emoticons. While we retain the characters required for word formation, punctuation, and emoticons, we remove all other characters, including numbers; we give the information regarding what all characters are retained as displayed in the above figure 2.5. After that, we remove any entries that are left empty because we have no use of them in our feature-level rating system. Let us consider that, for a product (not an item), we denote product review data as D={C1,C2,...,Cm}, comprising of m useful review comments.

Note that when we said customer review data of the item in the last section for feature selection, we meant review data of all the mobiles. In contrast, D is the review data of just the mobile product under consideration. By corrected data, we mean the data obtained after correcting the related feature words issue and after spelling correction in the useful data just extracted. To correct the data in such a manner, we need to disintegrate the reviews into words and process them separately. We use the NLTK package of python for this purpose. It helps in disintegrating the reviews into words as tokens while neglecting the spaces. It considers even period (.) as a token, which becomes useful later while breaking the comments into sentences. Each comment can now be represented as a set of tokens, i.e., Cj = {t 1 j ,t2 j,...,t |Cj | j }, where |Cj| denotes the number of tokens obtained in Cj and t i j represents the ith token of the jth comment.

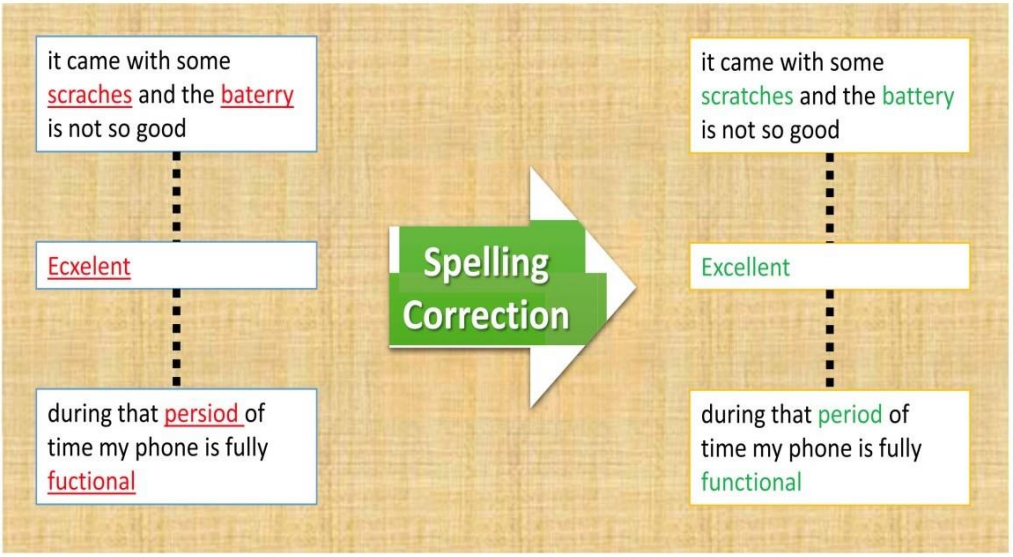


Fig.2.6: Preprocessing: Spelling Correction, where Misspelled Words are Corrected in this Illustration.

We correct any token t i j of the useful data in the following manner: t i j = Fk , if S(t i j) ∈ Fk or ti ∈ Fk , ∀Fk ∈ F S(t i j), otherwise where S(·) represents spelling correcting function (using autocorrect package of python).

If a token before or after spelling correction matches with any of the members in any of our feature words sets, we replace it with the feature keyword of that set; otherwise, we replace it with the corrected token. In this way, we take care of both the related words issue (by replacing with keywords) and the spelling correction issue simultaneously. The illustrations of spelling and keywords correction are given in Figs. 2.6 and 2.7. Thus, with the useful data extracted, disintegrated, and corrected, our reviews data for a product becomes structured. Now, we can say that t i j is the ith token of the jth comment.

Fig.2.7: Preprocessing: Keyword Correction, where Feature Words are Corrected to their Respective Feature Keywords in this Illustration.

C. Relevant Sentence

Extraction Since customer review data is now structured, we can group the continuous tokens present in a review as a sentence, as shown in the following definition of a set of sentences Xj derived from Cj X j = {(t u j ,...,t v j)|(t v j ,t u−1 j) = '.', (t u j ,...,t v−1 j) = '.', (u,v) ∈ {1,...,|Cj|} and v > u}. where we call a group of continuous tokens as a sentence if the last token and the token just before the first one are periods (.) and if all other tokens in that group are not periods. However, not all the sentences are relevant for feature-based rating. We define if a sentence Xl j , the lth sentence in X j , is relevant or not in the following way: ρ(Xl j) = 1, if Fk ∈ Xl j for any Fk ∈ F 0, otherwise where ρ(·) is relevance function for a sentence that outputs 1 if any of our feature keywords are present in the sentence. In this way, we extract only relevant sentences. See Fig. 2.8 for an example.

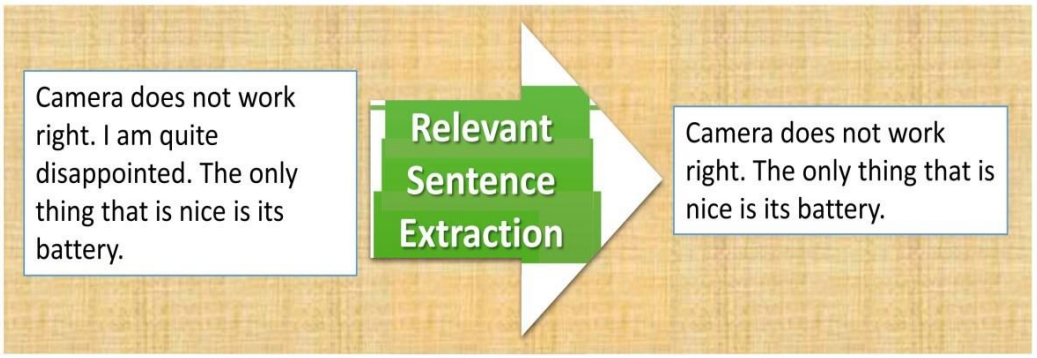


Fig.2.8: Relevant sentence extraction: only sentences containingfeature keywords are retained.

D. Feature-Based Ratings

Generation Having extracted relevant sentences, we can go through each sentence to figure out if it mentions a particular feature, say Fk . If yes, we can extract the emotion of the sentence to score it. For this purpose, we extract sentiment analysis scores for each of these sentences. We use because it accommodates emoticons also while performing the analysis. We use their compound score as the required sentiment analysis score.

It ranges between −1 and 1. Let the function that computes the sentiment score and assigns the appropriate rating be ψ(·). Then, we compute cumulative rating (Q(·)) for each feature over the entire product review data, i.e., D, in the following manner:

$$Q(Fk) = \sum_{Cj \in D} \rho(Xl\,j) = 1 \sum_{Xl\,j \in X\,j} \psi(Xl\,j) \times \delta(Fk \in Xl\,j) \times (\varphi(Cj) + 1)$$

Where δ(·) denotes logical function to check if a sentence consists of the concerned feature or not. During our accumulation, we also consider the number of votes received for a review to which the sentence belongs.

These votes inform us about the strength of the opinion associated with the reviews. Let φ(Cj) denote the number of votes received for Cj . Here, we are assuming that any sentence equally contributes to the strength of the opinion. We adjust the votes by adding 1 to account for self-votes of the customers who originally wrote the reviews. This accumulation is illustrated in Fig. 2.9. Then, we compute our final rating (A(·)) for a feature Fk using the following equation:

$$A(Fk)=Q(Fk)Cj\in D \ / \ \rho(Xl\ j)=1\ Xl\ j\in X\ j\ \delta(Fk \in Xl\ j) \times (\varphi(Cj) + 1)$$

We have developed a system to rate mobile phones in terms of 108 features based on customer reviews and review votes. We could rate 4000+ phones; this can help make personalized buying decisions and improve the products. We accomplish this by first converting the unstructured data into structured data; then, we extract the sentences comprising our feature keywords; then, we were able to provide the feature-level ratings through sentiment analysis of these sentences.

We rank the phones based on the number of features that they are best at, and accordingly, we were able to recommend the best phones for a feature.

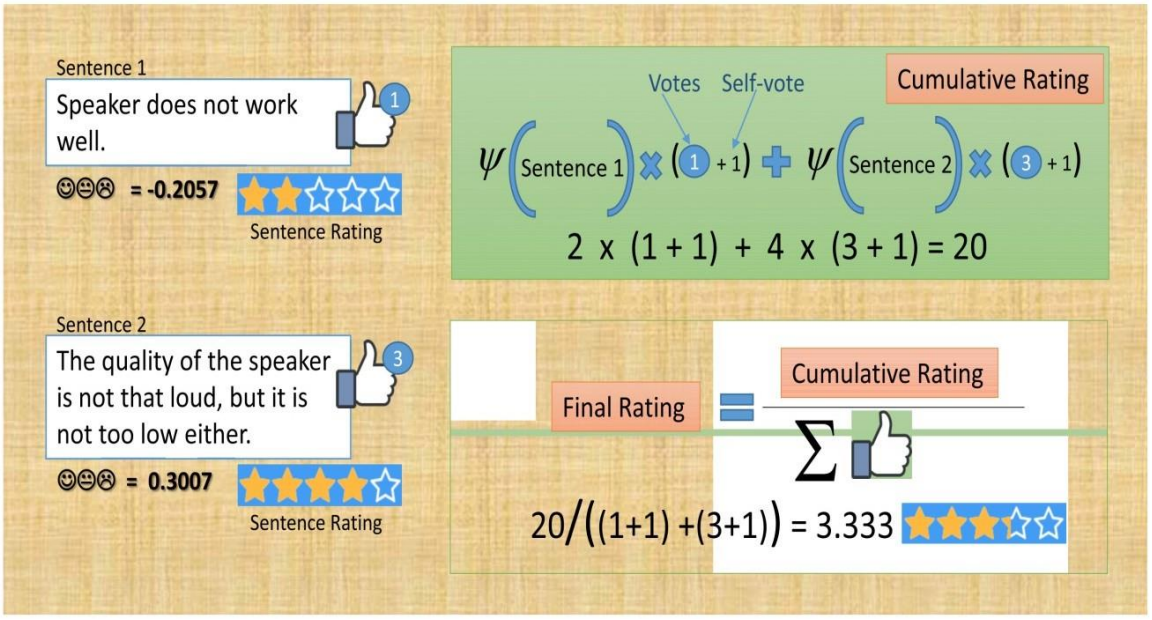


Fig.2.9: First, to compute cumulative rating of a feature, we accumulate sentence rating of all sentences having the feature word along with their total votes. Second, to compute the final rating, we divide the cumulative rating by the total number of votes.

We tested our methodology on the "phone" named feature by considering the overall customer ratings as ground-truth ratings. The performance of our method is found to be decent. We obtain MAE of only 0.555, i.e., approximately just half a star. We get 52.3% accuracy if exact integer ratings have to be predicted. However, if we can tolerate the one-star integer rating error, the accuracy jumps to 93.8%. The proposed approach is unsupervised. As an extension, we will work on improving the performance by taking a weakly supervised or supervised approach to this problem, for which we will have to annotate the available data in terms of all our 108 features.

2.4 RATING AND MATCHING IN PEER REVIEW SYSTEM

Peer review (e.g., review of research papers) is essential for the success of the scientific community. In peer review, the reviewers voluntarily exert costly effort in reviewing papers. Hence, it is important to design mechanisms to elicit high effort from reviewers. Exploiting the fact that the researchers interact with each other repeatedly (e.g., by submitting and reviewing papers over years), we propose a rating and matching mechanism to elicit high effort from reviewers. Our proposed mechanism overcomes two major difficulties, namely adverse selection (i.e., the unidentifiable quality of heterogeneous reviewers) and moralhazard (i.e., the unobservable effort levels from reviewers). Specifically, our proposed mechanism assigns and updates ratings for the researchers, and matches researchers' papersto reviewers with similar ratings.

In this way, the mechanism identifies different types of reviewers by their ratings, and incentivizes different reviewers to exert high effort. Focusing on the matching rule, we first provide design guidelines for a general matching rule that leads the system to equilibrium, where the reviewers' types are identified and their high efforts are elicited. Then we study in detail a baseline matching rule that assigns each researcher's paper to one of the two reviewers with the closest ratings, provide guidelines of how to choose the initial ratings, and analyze equilibrium review quality and equilibrium ratings.

Peer review (e.g., review of research papers) is an integral and essential part of the academia. The efficiency of the peer review system has a critical impact on the quality of a research community. A key problem in the peer review system is that the reviewers voluntarily exert effort in reviewing papers, and that the efforts are costly for the reviewers. Hence, it is important to design mechanisms in which the reviewers have incentives to review papers with high effort, such that the efficiency of the peer review system is improved. Based on the fact that the researchers interact with each other repeatedly (e.g., by submitting

and reviewingpapers over time), we propose a rating and matching mechanism to elicit high effort from the reviewers.

The basic idea of our proposed mechanism is to assign a rating for each researcher and update their ratings based on their past review quality reported by the authors. Next we match the papers from high-rating researchers more likely to high-rating reviewers. Hence, the researchers have incentives to exert high effort levels to improve their ratings and the chances of getting high-quality reviewers.

There are two major challenges in designing an efficient rating and matching mechanism. First, even though the quality of a review can be assessed by the authors, there is no way to observe the effort level of the reviewer. This is because the quality of the review is an unknown function of the effort level exerted by the reviewer. The problem of unobservable effort levels is called moral hazard problems in economics literature. Second, since the reviewers are heterogeneous, their review quality can be different even if they exert the same amount of effort. This problem of unidentifiable quality (i.e., the mapping from a reviewer's effort level to the review quality) of a reviewer is called adverse selection problems in economics. In the presence of both moral hazard and adverse selection problems, it isdifficult to identify high-quality and low-quality reviewers, and the effort levels exerted by them. In this paper, we design the rating update rule and the matching rule, such that we can identify different types of reviewers and incentivize them to choose high effort levels in the equilibrium.

The rating update rule ensures that the ratings of the reviewers truly reflect their review quality. The matching rule incentivizes different types of reviewers to exert high effort. In the equilibrium of the system, more capable reviewers with higher quality and lower cost in reviewing provide higher-quality review, and hence have higher ratings which in turn confirm their quality. We first propose a baseline matching rule, which simply matches a researcher's paper to reviewers with closest (higher and lower) ratings.

The probability of being matched to a higher-rating or lower-rating reviewer depends on the distance from the researcher's rating to the reviewers' ratings. We provide design guidelines of how to choose the initial ratings, prove the convergence to the equilibrium, and analyze the equilibrium review quality and rating of researchers with different quality. We then extend the baseline matching rule in two directions. Both extensions enable us to construct a class of matching rules and to tune the extent to which the matching rules reward and/or punish reviewers. The first extension provides extra reward and/or punishment by adjusting the

probabilities of matching each researcher's paper to its neighbors. The second extension provides extra reward and/or punishment by allowing matching each researcher's paper to reviewers other than its neighbors.

Our interesting finding shows that it is beneficial (in the sense that the optimal equilibrium review quality is higher) to reward reviewers in the first extension, and to punish reviewers in the second extension, due to the different ways the reward and punishment are carried out. Our results show the importance of designing the correct reward and punishment mechanisms for different types of matching rules.

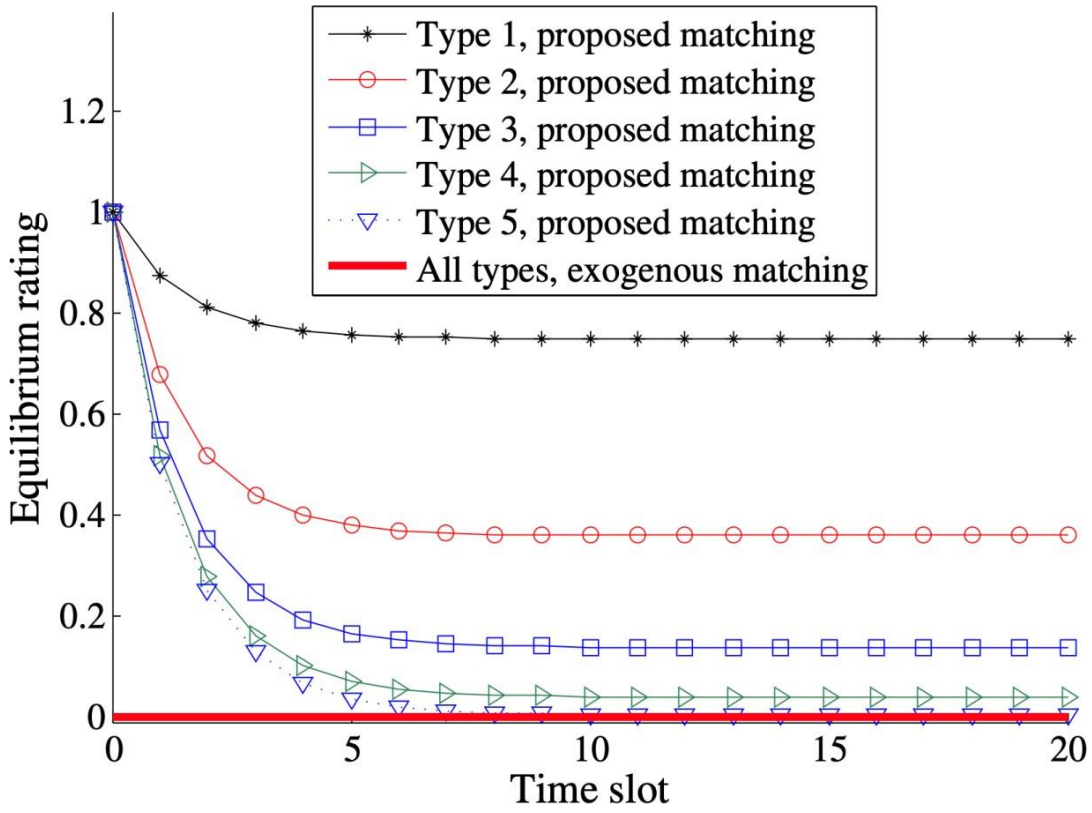


Fig.2.10: Convergence to the conjectural equilibrium.

1) Convergence and Performance Improvement

Fig. 2.10 shows the convergence to the conjectured equilibrium from the initial rating 1.0. As predicted, the equilibrium rating and review quality is ordered according to the types of the researchers. Fig. 2.10 also shows that under exogenous matching rules that do not depend on researchers' ratings, the reviewers exert lowest efforts all the time. Hence, the proposed endogenous matching greatly improves the performance of the review system.

2) Impact of Initial Ratings

In Fig. 2.11, we examine the impact of the initial rating. We can see that when the initial rating is high enough (1.2 and 0.8), the system converges to the same equilibrium. When the initial rating is low (0.5), some types may stay at the initial rating. This is consistent with Proposition 1, which indicates that when the initial rating is sufficiently small, the initial rating is the equilibrium rating for all the researchers.

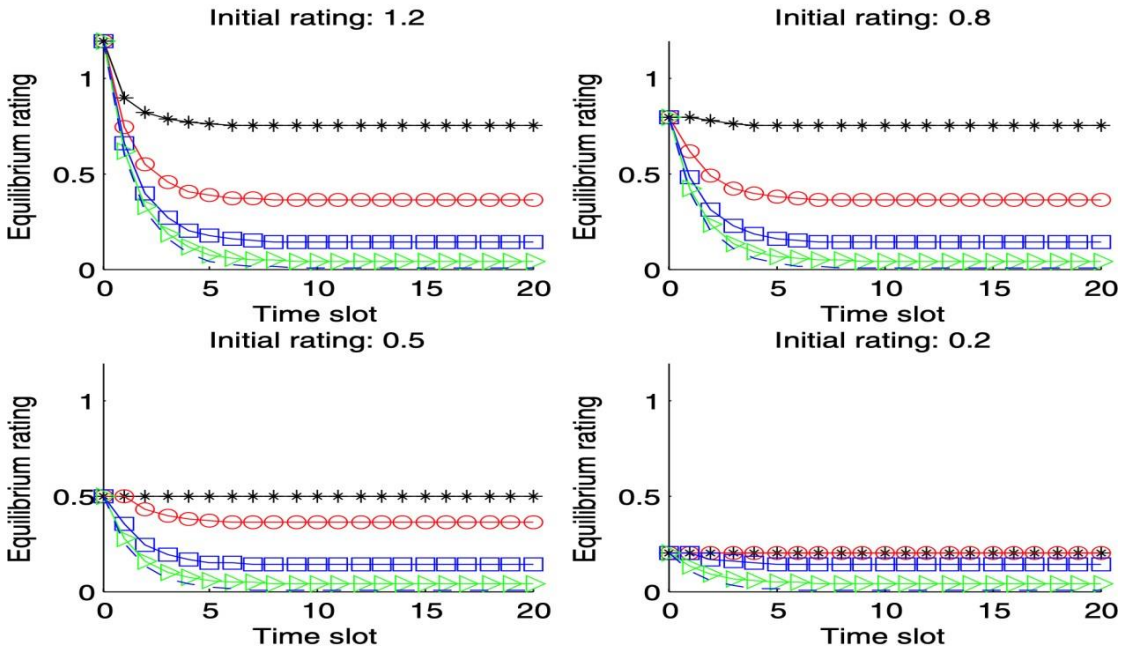


Fig.2.11: Equilibria under Differential Initial Ratings.

3) Different Matching Rules

We compare the sum review quality and the social welfare (i.e., the total benefit minus cost) at the equilibrium under different matching rules. under different parameters γ. We can see that in our setting, the optimal γ should be 0.1, which results in the highest sum review quality.

This is consistent with our theoretical results: we can find a rewarding matching rule that outperforms the baseline rule. It is worth mentioning that the matching rule that maximizes the sum review quality may not be the one that maximizes the social welfare. This is reasonable, because higher review quality also results in higher cost. In Table II, we evaluate the second extension of matching rules under different parameters γr and γp. We can see that the optimal sum review quality is achieved when γr = 0 and γp = 1.

The threat of being matched to an even lower rating reviewer provides more incentive for researchers to exert high effort. Again, such a matching rule does not result in the optimal social welfare. The optimal social welfare is achieved when $\gamma r = 0.5$ and $\gamma p = 0.5$, where the researchers are also rewarded for good behaviors.

We studied the problem of effort elicitation in peer review systems. We modeled the two key features in such systems: namely moral hazard (i.e., the unobservable effort by the reviewers) and adverse selection (i.e., the unidentifiable quality of the reviewers).

We proposed a rating and matching mechanism to identify the reviewers of different types, and elicit the appropriate amount of effort from different reviewers. We extensively studied the design of matching rules, in terms of the initial ratings, the convergence, and the equilibrium ratings and review effort. We also studied the extensions to different classes of matching rules, and proved the efficiency of different reward and punishment mechanisms under different matching rules.

SOFTWARE REQUIREMENT SPECIFICATION

CHAPTER 3

SYSTEM REQUIREMENT SPECIFICATION

The following document details the requirements for the Management System for Distributed Environment. The software requirements shall be specified for all the phases of the Management System. The purpose of this Software Requirement Specification (SRS) Document is to specify the user goals and tasks that need to be achieved. SRS forms the basis of software development. Software Requirement Specification acts as a reference for validation of the final product. It helps to check if the software has met the requirements. Hence a high-quality SRS is a prerequisite to high quality software. Here, the chapter gives an overview on the following information:

1. System Overview.
2. Specific Requirements.
3. System requirements.

3.1 SYSTEM OVERVIEW

1. Purpose

The purpose of proposed project is to better the feedback system and enable students to provide feedback in online mode without revealing the identity of individual and to display the respective feedback score under the faculty profile along with their qualification.

2. Input

The 50% of input provided to the system are the feedback scores entered by students that are provided through forms for each teaching faculty and other 50% of input from different parameters with respect to the faculty.

3. Output

An academic progress score is provided under each faculty profile.

3.2 SPECIFICATION REQUIREMENTS

To properly satisfy the basic goals, an SRS should have certain properties and should contain different types of requirements and below stated are some of the important requirements involved in developing software. System requirements should simply describe the external behavior of the system and its operational constraints.

3.2.1 FUNCTIONAL REQUIREMENTS

These are statements of services the system should provide, how the system should react to inputs and how the system should behave in particular situations.

a. 50% of data is acquired from the student's feedback score.

b. Other 50% is taken from the results achieved, qualification and other achievements achieved by the faculty.

3.2.2 NON-FUNCTIONAL REQUIREMENTS

The major non-functional Requirements of the system are as follows:

1. **Usability:**

The system is designed with a completely automated process of taking studentfeedback from students in online mode.

2. **Reliability:**

The system is more reliable because of the qualities of Android studio as it has extensive support libraries and it is portable and interactive.

3. **Performance:**

The system does not require any training data to be fed to the application and hence gives the academic progress score faster under each faculty profile.

4. **Scalability:**

The accuracy of the system could be improved with an increase in the number of student feedback score.

5. **Readability:**

The system is developed in Java, which is considered to be the most English like programming language and hence is readable by any user.

3.3 SYSTEM REQUIREMENTS

The hardware and software components of a computer system that are required to install and use the software efficiently are specified. The minimum system requirements need to be met for the program to run on the system.

3.3.1 HARDWARE REQUIREMENT

The hardware requirements specify the necessary hardware which provides us the platform to implement our programs.

- Processor : AMD Ryzen 5
- Hard Disk : 1 TB
- RAM : 8 GB
- Clock speed : 2.40 GHz

3.3.2 SOFTWARE REQUIREMENT

The software requirements specify the pre-installed software needed to run the code being implemented in this project.

- Front end design : XML, Java
- Back-end design : PHP
- Server : XAMPP Server
- Database ; MYSQL
- IDE : Android Studio, Visual Studio Code

3.3.3 FRONT-END DESIGN

- **XML**

Extensible Markup Language (XML) is a markup language and file format for storing, transmitting, and reconstructing arbitrary data. It defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.The World Wide Web Consortium's XML 1.0 Specification of 1998 and several other related specifications - all of them free open standards define XML. XML is an application profile of SGML (ISO 8879).

The versatility of SGML for dynamic information display was understood by early digital media publishers in the late 1980s prior to the rise of the Internet. By the mid-1990s some practitioners of SGML had gained experience with the then-new World Wide Web, and believed that SGML offered solutions to some of the problems the Web was likely to faceas it grew. Dan Connolly added SGML to the list of W3C's activities when he joined the staff in 1995; work began in mid-1996 when Sun Microsystems engineer Jon Bosak developed a charter and recruited collaborators. Bosak was well connected in the small community of people who had experience both in SGML and the Web.

The XML Working Group never met face-to-face; the design was accomplished using a combination of email and weekly teleconferences. The major design decisions werereached in a short burst of intense work between August and November 1996, when the first Working Draft of an XML specification was published.

Further design work continued through 1997, and XML 1.0 becamea W3C Recommendation on February 10, 1998. The design goals of XML emphasize simplicity, generality, and usability across the Internet. It is a textual data format with strong support via Unicode for different human languages. Although the design of XML focuses on documents, the language is widely used for the representation of arbitrary data structures such as those used in web services.

The main purpose of XML is serialization, i.e. storing, transmitting, and reconstructing arbitrary data. For two disparate systems to exchange information, they need to agree upon a file format. XML standardizes this process. XML is analogous to a lingua franca for representing information. As a markup language, XML labels, categorizes, and structurally organizes information.

XML tags represent the data structure and contain metadata. What's within the tags is data, encoded in the way the XML standard specifies. An additional XML schema (XSD) defines the necessary metadata for interpreting and validating XML. (This is also referredto as the canonical schema.) XML allows the use of any of the Unicode-defined encodings and any other encodings whose characters also appear in Unicode. XML also provides a mechanism whereby an XML processor can reliably, without any prior knowledge, determine which encoding is being used. Encodings other than UTF-8 and UTF-16 are not necessarily recognized by every XML parser.

- **JAVA**

Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general- purpose programming language intended to let programmers write once, run anywhere (WORA), meaning that compiled Java code can run on all platforms that support Java without the need to recompile. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but has fewer low-level facilities than either of them. The Java runtime provides dynamic capabilities (such as reflection and runtime code modification) that are typically not available in traditional compiled languages. As of 2019, Java was one of the most popular programming languages in use according to GitHub, particularly for client–server web applications, with a reported 9 million developers.

Java was originally developed by James Gosling at Sun Microsystems and released in May 1995 as a core component of Sun Microsystems' Java platform. The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun had relicensed most of its Java technologies under the GPL-2.0-only license. Oracle offers its own Hot Spot Java Virtual Machine, however the official reference implementation is the Open JDK JVM which is free open-source software and used by most developers and is the default JVM for almost all Linux distributions.

The Java language is a key pillar in Android, an open source mobile operating system. Although Android, built on the Linux kernel, is written largely in C, the Android SDK uses the Java language as the basis for Android applications but does not use any of its standard GUI, SE, ME or other established Java standards. The bytecode language supported by the Android SDK is incompatible with Java bytecode and runs on its own virtual machine, optimized for low-memory devices such as smartphones and tablet computers. Depending on the Android version, the bytecode is either interpreted by the Dalvik virtual machine or compiled into native code by the Android Runtime. Android does not provide the full Java SE standard library, although the Android SDK does include an independent implementation of a large subset of it.

The syntax of Java is largely influenced by C++ and C. Unlike C++, which combines the syntax for structured, generic, and object-oriented programming, Java was built almost exclusively as an object-oriented language. All code is written inside classes, and every data item is an object, with the exception of the primitive data types, (i.e. integers, floating-point numbers, boolean values, and characters), which are not objects for performance reasons. Java reuses some popular aspects of C++ (such as the method).

Oracle Corporation is the current owner of the official implementation of the Java SE platform, following their acquisition of Sun Microsystems on January 27, 2010. This implementation is based on the original implementation of Java by Sun. The Oracle implementation is available for Microsoft Windows (still works for XP, while only later versions are currently officially supported), macOS, Linux, and Solaris. Because Java lacks any formal standardization recognized by Ecma International, ISO/IEC, ANSI, or other third-party standards organizations, the Oracle implementation is the de facto standard.

The Oracle implementation is packaged into two different distributions: The Java Runtime Environment (JRE) which contains the parts of the Java SE platform required to run Java programs and is intended for end users, and the Java Development Kit (JDK), which is intended for software developers and includes development tools such as the Java compiler, Javadoc, Jar, and a debugger.

OpenJDK is another notable Java SE implementation that is licensed under the GNU GPL. The implementation started when Sun began releasing the Java source code under the GPL. As of Java SE 7, OpenJDK is the official Java reference implementation.

The goal of Java is to make all implementations of Java compatible. Historically, Sun's trademark license for usage of the Java brand insists that all implementations be compatible. This resulted in a legal dispute with Microsoft after Sun claimed that the Microsoft implementation did not support RMI or JNI and had added platform-specific features of their own. Sun sued in 1997, and, in 2001, won a settlement of US$20 million, as well as a court order enforcing the terms of the license from Sun. As a result, Microsoft no longer ships Java with Windows. Platform-independent Java is essential to Java EE, and an even more rigorous validation is required to certify an implementation. This environment enables portable server-side applications.

➢ Types of Java Applications

There are mainly four types of applications that can be created using Java programming:

1) Standalone Application

Standalone applications are also known as desktop applications or window-based applications. These are traditional software that we need to install on every machine. Examples of standalone application are Media player, antivirus, etc.

2) Web Application

An application that runs on the server side and creates a dynamic page is called a web application. Currently, Servlet, JSP, Struts, Spring, Hibernate, JSF, etc. technologies are used for creating web applications in Java.

3) Mobile Application

An application which is created for mobile devices is called a mobile application. Currently, Android and Java ME are used for creating mobile applications.

➢ Java Platforms / Editions

There are 4 platforms or editions of Java:

1) Java SE (Java Standard Edition)

It is a Java programming platform. It includes Java programming APIs such as java.lang, java.io, java.net, java.util, java.sql, java.math etc. It includes core topics like OOPs, String, Regex, Exception, Inner classes, Multithreading, I/O Stream, Networking, AWT, Swing, Reflection, Collection, etc.

2) Java EE (Java Enterprise Edition)

It is an enterprise platform that is mainly used to develop web and enterprise applications. It is built on top of the Java SE platform. It includes topics like Servlet, JSP, Web Services, EJB, JPA, etc.

3) Java ME (Java Micro Edition)

It is a micro platform that is dedicated to mobile applications.

4) JavaFX

It is used to develop rich internet applications and also it uses a lightweight user interface API.

3.3.4 BACK-END DESIGN

- **PHP**

PHP is a general-purpose scripting language geared toward web development. It was originally created by Danish-Canadian programmer Rasmus Lerdorf in 1994. The PHP reference implementation is now produced by The PHP Group. PHP originally stood for Personal Home Page, but it now stands for the recursive initialize PHP: Hypertext Preprocessor.

PHP code is usually processed on a web server by a PHP interpreter implemented as a module, a daemon or as a Common Gateway Interface (CGI) executable. On a web server, the result of the interpreted and executed PHP code – which may be any type of data, such as generated HTML or binary image data.

Additionally, PHP can be used for many programming tasks outside the web context, such as standalone graphical applications and robotic drone control. PHP code can also be directly executed from the command line. The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on a variety of operating systems and platforms.

The PHP language evolved without a written formal specification or standard until 2014, with the original implementation acting as the de facto standard which other implementations aimed to follow. Since 2014, work has gone on to create a formal PHP specification.

W3Techs reports that, as of January 2022, "PHP is used by 78.1% of all the websites whose server-side programming language we know." PHP version 7.4 is the most used version. Support for version 7.3 was dropped on 6 December 2021.

PHP includes various free and open-source libraries in its source distribution, or uses them in resulting PHP binary builds. PHP is fundamentally an Internet-aware system with built-in modules for accessing File Transfer Protocol (FTP) servers and many database servers, including PostgreSQL, MySQL, Microsoft SQL Server and SQLite (which is an embedded database), LDAP servers, and others. Numerous functions familiar to C programmers, such as those in the stdio family, are available in standard PHP builds. The only complete PHP implementation is the original, known simply as PHP.

To disambiguate it from other implementations, it is sometimes unofficially called "Zend PHP". The Zend Engine compiles PHP source code on-the-fly into an internal format that it can execute, thus it works as an interpreter. It is also the "reference implementation" of PHP, as PHP has no formal specification, and so the semantics of Zend PHP define the semantics of PHP. Due to the complex and nuanced semantics of PHP, defined by how Zend works, it is difficult for competing implementations to offer complete compatibility.

PHP's single-request-per-script-execution model, and the fact that the Zend Engine is an interpreter, leads to inefficiency; as a result, various products have been developed to help improve PHP performance. In order to speed up execution time and not have to compile the PHP source code every time the web page is accessed, PHP scripts can also be deployed in the PHP engine's internal format by using an opcode cache, which works by caching the compiled form of a PHP script (opcodes) in shared memory to avoid the overhead of parsing and compiling the code every time the script runs. An opcode cache, Zend Opcache, is built into PHP since version 5.5. Another example of a widely used opcode cache is the Alternative PHP Cache (APC).

Basic object-oriented programming functionality was added in PHP 3 and improved in PHP 4. This allowed for PHP to gain further abstraction, making creative tasks easier for programmers using the language. Object handling was completely rewritten for PHP 5, expanding the feature set and enhancing performance. In previous versions of PHP, objects were handled like value types. The drawback of this method was that code had to make heavy use of PHP's "reference" variables if it wanted to modify an object it was passed rather than creating a copy of it.

PHP 5 introduced private and protected member variables and methods, along with abstract classes, final classes, abstract methods, and final methods. It also introduced a standard way of declaring constructors and destructors, similar to that of other object-oriented languages such as C++, and a standard exception handling model. Furthermore, PHP 5 added interfaces and allowed for multiple interfaces to be implemented. There are special interfaces that allow objects to interact with the runtime system. PHP code is usually processed on a web server by a PHP interpreter implemented as a module, a daemon or as a Common Gateway Interface (CGI). It can be used for many programming tasks outside the web context, such as standalone graphical applications and robotics drone control. PHP is used by 79.2% of all websites whose server-side programming language is known.

3.3.5 DATABASE

- **MYSQL**

MySQL is the most popular Open-Source Relational SQL Database Management System and it is used for developing various web-based software applications. SQL is a standard language for storing, manipulating and retrieving data in databases based upon relational algebra and tuple relational calculus.

MySQL was created by a Swedish company, MySQL AB, founded by Swedes David Ax mark, Allan Larsson and Finland Swede Michael "Monty" Wideners. Original development of MySQL by Wideners and Ax mark began in 1994. The first version of MySQL appeared on 23 May 1995. It was initially created for personal usage from MySQL based on the low-level language ISAM, which the creators considered too slow and inflexible. They created a new SQL interface, while keeping the same API as MySQL. By keeping the API consistent with the MySQL system, many developers were able to use MySQL instead of the (proprietarily licensed) MySQL antecedent.

The scope of SQL includes data insert, query, update and delete, schema creation and modification, and data access control. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL has stand-alone clients that allow users to interact directly with a MySQL database using SQL, but more often, MySQL is used with other programs to implement applications that need relational database capability. MySQL is also used by many popular websites. We have used MYSQL for creating our project database.

MySQL is written in C and C++. Its SQL parser is written in yacc, but it uses a home-brewed lexical analyzer. The MySQL server software itself and the client libraries use dual-licensing distribution. They are offered under GPL version 2, or a proprietary license. Support can be obtained from the official manual. Free support additionally is available in different IRC channels and forums. Oracle offers paid support via its MySQL Enterprise products. They differ in the scope of services and in price. Additionally, a number of third party organizations exist to provide support and services.

MySQL has received positive reviews, and reviewers noticed it "performs extremely well in the average case" and that the "developer interfaces are there, and the documentation (not to mention feedback in the real world via Web sites and the like) is very, very good". It has also been tested to be a "fast, stable and true multi-user, multi-threaded SQL database server".

3.3.6 INTEGRATED DEVELOPMENT ENVIRONMENT

- **ANDROID STUDIO**

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development.

It is available for download on Windows, macOS and Linux based operating systems or as a subscription-based service in 2020. It is a replacement for the Eclipse Android Development Tools (E-ADT) as the primary IDE for native Android application development. Android Studio was announced on May 16, 2013, at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0.

- **Features of Android Studio**

  - Android-specific refactoring and quick fixes
  - Lint tools to catch performance, usability, version compatibility and other problems
  - ProGuard integration and app-signing capabilities
  - Template-based wizards to create common Android designs and components
  - Support for building Android Wear apps
  - Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine.
  - Android Virtual Device (Emulator) to run and debug apps in the Android studio.

Android Studio supports all the same programming languages of IntelliJ (and CLion) e.g. Java, C++, and more with extensions, such as Go; and Android Studio 3.0 or later supports Kotlin and "all Java 7 language features and a subset of Java 8 language features that vary by platform version." External projects backport some Java 9 features. While IntelliJ states that Android Studio supports all released Java versions, and Java 12, it's not clear to what level Android Studio supports Java versions up to Java 12 (the documentation mentions partial Java 8 support). At least some new language features up to Java 12 are usable in Android. Once an app has been compiled with Android Studio, it can be published on the Google Play Store. The application has to be in line with the Google Play Store developer content policy.

➢ VISUAL STUDIO CODE

Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft for Windows, Linux and macOS.

In the Stack Overflow 2021 Developer Survey, Visual Studio Code was ranked the most popular developer environment tool, with 70% of 82,000 respondents reporting that they use it. Visual Studio Code was first announced on April 29, 2015, by Microsoft at the 2015 Build conference. A preview build was released shortly thereafter.

On November 18, 2015, the source of Visual Studio Code was released under the MIT License, and made available on GitHub. Extension support was also announced.[13] On April 14, 2016, Visual Studio Code graduated from the public preview stage and was released to the Web. Microsoft has released most of Visual Studio Code's source code on GitHub under the permissive MIT License, while the releases by Microsoft are proprietary freeware. Visual Studio is based on the Electron framework, which is used to develop Node.js Web applications that run on the Blink layout engine. Visual Studio Code employs the same editor component (codenamed "Monaco") used in Azure DevOps (formerly called Visual Studio Online and Visual Studio Team Services).

Out of the box, Visual Studio Code includes basic support for most common programming languages. This basic support includes syntax highlighting, bracket matching, code folding, and configurable snippets. Visual Studio Code also ships with IntelliSense for JavaScript, TypeScript, JSON, CSS, and HTML, as well as debugging support for Node.js. Support for additional languages can be provided by freely available extensions on the VS Code Marketplace.

Instead of a project system, it allows users to open one or more directories, which canthen be saved in workspaces for future reuse. This allows it to operate as a language- agnostic code editor for any language. It supports many programming languages and a set of features that differs per language. Unwanted files and folders can be excluded from theproject tree via the settings. Many Visual Studio Code features are not exposed through menus or the user interface but can be accessed via the command palette. Visual Studio Code can be extended via extensions, available through a central repository. This includesadditions to the editor] and language support. A notable feature is the ability to create extensions that add support for new languages, themes, debuggers, time travel debuggers, perform static code analysis, and add code linters using the Language Server Protocol.

Source control is a built-in feature of Visual Studio Code. It has a dedicated tab inside of the menu bar where users can access version control settings and view changes made to the current project. To use the feature, Visual Studio Code must be linked to any supported version control system (Git, Apache Subversion, Perforce, etc.). This allows users to create repositories as well as to make push and pull requests directly from the Visual Studio Code program.

Visual Studio Code includes multiple extensions for FTP, allowing the software to be used as a free alternative for web development. Code can be synced between the editor and the server, without downloading any extra software.

Visual Studio Code allows users to set the code page in which the active document is saved, the newline character, and the programming language of the active document. Thisallows it to be used on any platform, in any locale, and for any given programming language.

Visual Studio Code collects usage data and sends it to Microsoft, although this can be disabled. Due to the open-source nature of the application, the telemetry code is accessible to the public, who can see exactly what is collected.

SYSTEM DESIGN

Chapter 4

SYSTEM DESIGN

4.1 INTRODUCTION

The following chapter gives overall flow of the project and algorithm used in the design. Implementation is a stage in software development where the software design is realized as a set of program units. The objects that are identified in the design stage are implemented, and a function which manipulates these objects is realized.

In this phase, the system or system modifications are installed and made operational in a productive environment. Activities in this phase include notification of implementation to end users, execution of the previously defined training plan, data entry or conversion andpost implementation review. This phase continues until the system is operating in production accordance with the defined user requirement. Successful implementation may not guarantee improvement but it will prevent improper installation. They are following four formalities:

- Careful Planning.
- Investigation of the system and constraints.
- Design the methods to achieve the changes.
- Training the admin in the changed phase.

Implementation of any software is always preceded by important decisions regarding selection of the platform, the language used, etc. these decisions are often influenced by several factors such as the real environment in which the system works, the speed that is required, the security concerns, and other implementation specific details. There are three major implementation decisions that have been made before the implementation of the proposed project. They are:

1. Selection of platform (Operating System).
2. Selection of programming language for development of application.
3. Details of the flowchart used in the proposed project.

4.1.1 SELECTION OF PLATFORM

Windows 7 and above version introduced several new features to the Windows line,including:

- Fast user switching, which allows a user to save the current state and open applications of their desktop and allow another user to log on without losing that information.
- Faster start-up and hibernation sequences.
- The ability to discard a newer device driver in favor of the previous one (known as driver rollback), should a driver upgrade not produce desirable results.
- A new, arguably more user-friendly interface, including the framework fordeveloping themes for the desktop environment.

4.1.2 SELECTION OF LANGUAGE

Java is a general-use high-level programming language that bills itself as powerful, fast, friendly, open, and easy to learn. Its high-level built-in data structures, combined with dynamic typing and dynamic binding; make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together.

Rather than having all of its functionality built into its core, Java was designed to be highly extensible. This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications. James gosling vision of a small core language with a large standard library and easily extensible interpreter stemmed from his frustrations with ABC, which espoused the opposite approach. Some of the powerful statistical and numerical packages:

- Apache Commons, is an open-source for creating reusable Java components. It has numerical packages for linear algebra and non-linear optimization.
- Colt provides a set of Open-Source Libraries for High Performance Scientific and Technical Computing.
- Efficient Java Matrix Library (EJML) is an open-source linear algebra library for manipulating dense matrices.

4.2 HIGH LEVEL DESIGN

The purpose of the design phase is to plan a solution of the problem specified by the requirement document. This phase is the first step in moving from the problem domain to the solution domain. The design of the system is perhaps the most critical factor affecting the quality of the software.

Here we build the System Block Diagram that will be helpful to understand the behavior of the system. Here we divide the problem into modules. Data flow Diagrams show flow of data between/among modules. This chapter presents the following:

- Design Considerations: This section describes many issues, which need to be addressed or resolved before attempting to device a complete design solution.
- General Constraints.
- Development Methods.
- System Architecture: This section describes the DFDs, which are the root part for any design.

4.3 DESIGN CONSIDERATION

The purpose of the design is to plan the solution of a problem specified by the requirements document. This phase is the first step in moving from problem to solution domain. In other words, starting with what is needed design takes us to work how to satisfy the needs. The design of the system is perhaps the most critical factor affecting the quality of the software and has a major impact on the later phases, particularly testing and maintenance.

System design aims to identify the modules that should be in the system, the specifications of these modules and to interact with each other to produce the desired results.

At the end of the system design all the major data structures, file formats, output formats as well as major modules in the system and their specifications are decided.

4.4 PROCESS FLOWCHART

A process flow diagram (PFD) is a diagram commonly used in chemical and process engineering to indicate the general flow of plant processes and equipment. The PFD displays the relationship between major equipment of a plant facility and does not show minor details such as piping details and designations.

One of the most important uses of flowcharts is to depict through images how a process is performed from start to finish, typically in sequential order. Another commonly-used term for a PFD is a flow sheet. Fig 4.1 shows the detailed process flowchart of the application.

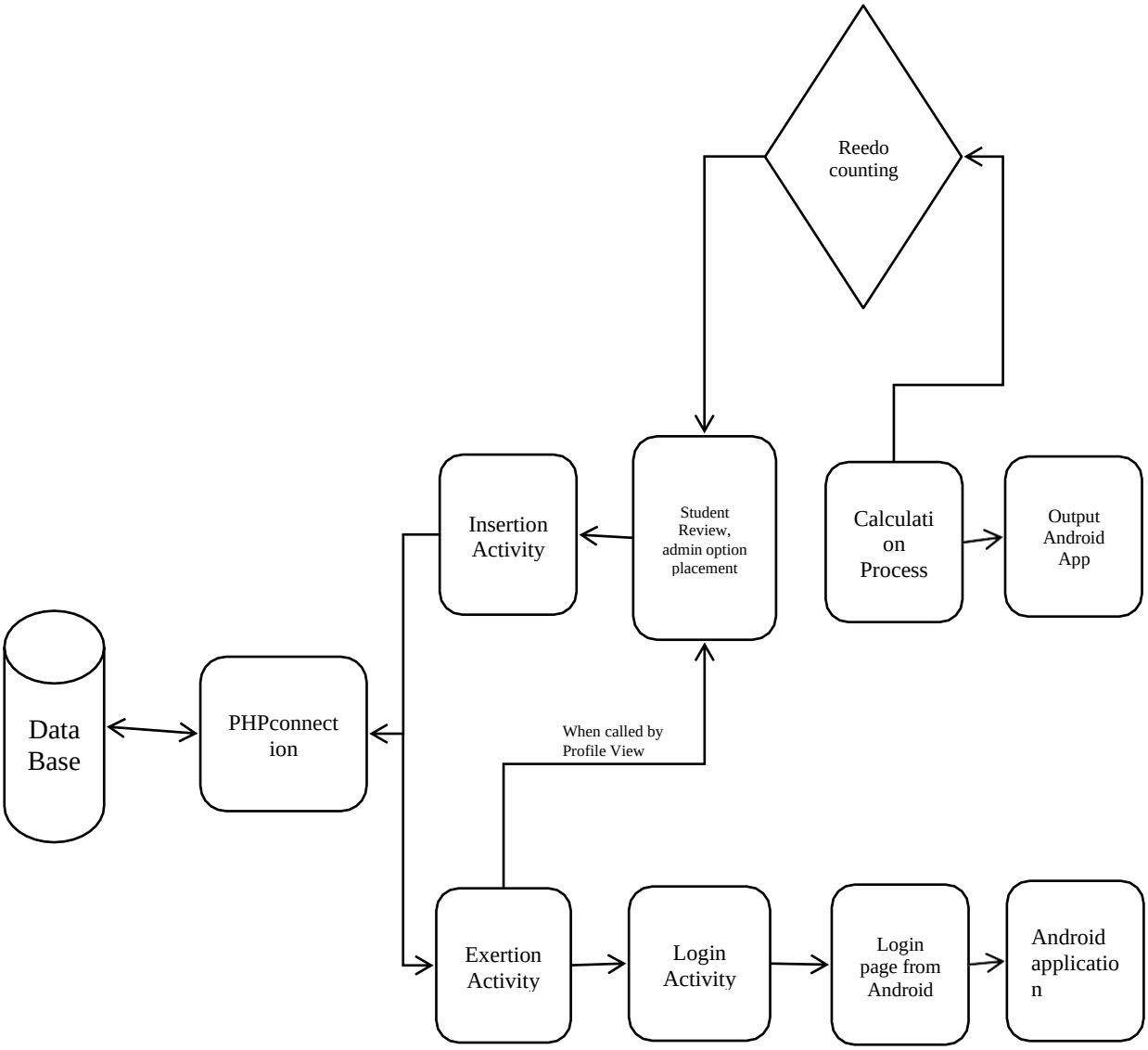


Fig.4.1: Process Flowchart

4.5 DATA FLOW DIAGRAM

Data flow models are an intuitive way of showing how data is processed by a system. At the analysis level, they should be used to model the way in which data is processed in the existing system. The notations used in these models represents functional processing, data stores and data movements between functions. Data flow models are used to show how data flows through a sequence of processing steps. The data is transferred at each step before moving on to the next stage. These processing steps or transformations are program functions when data flow diagrams are used to explain a software design.

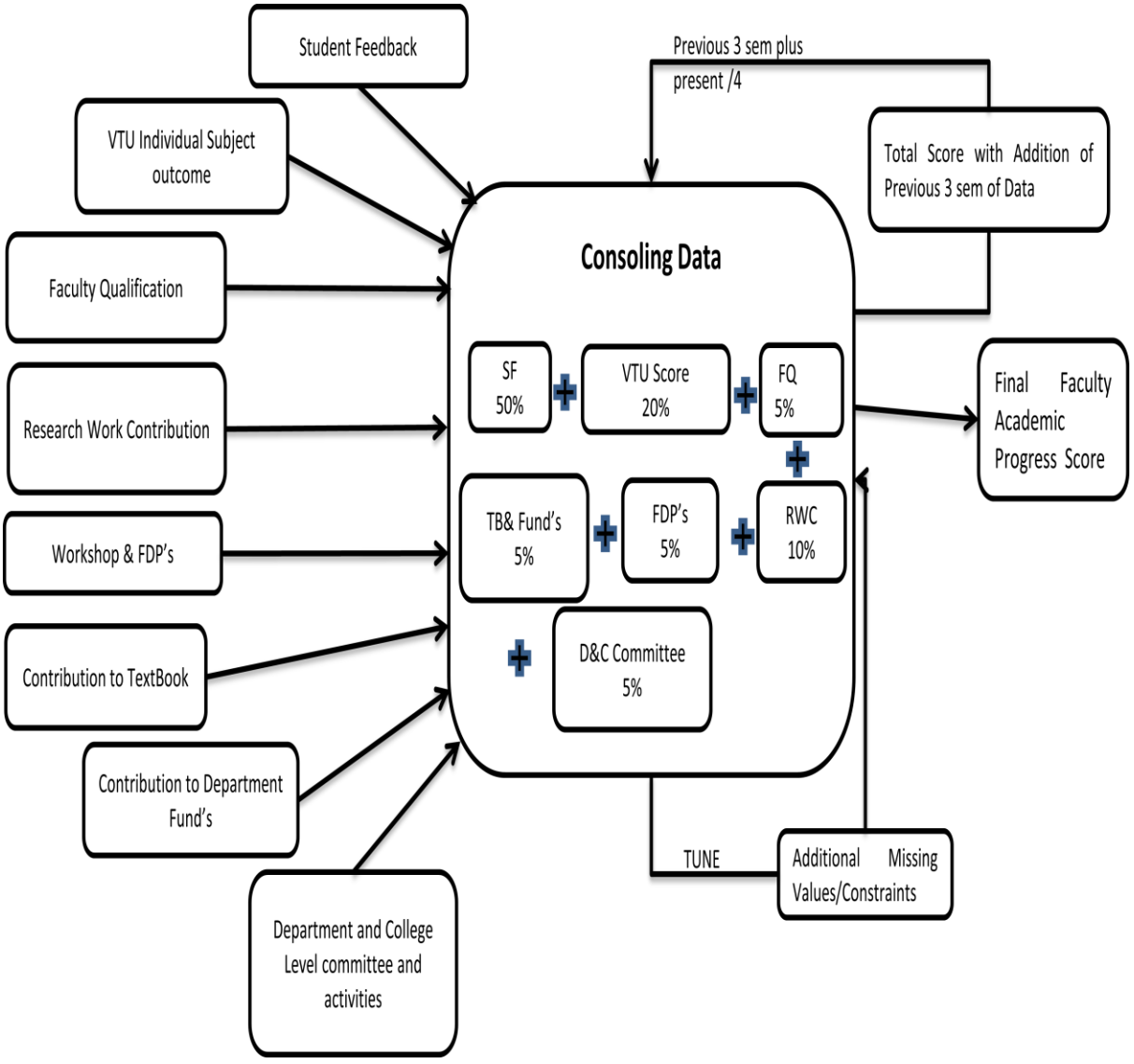


Fig.4.2: Data Flow Diagram

The Data Flow Diagram (DFD) is a structured analysis and design method. It is a visual tool to depict logic models and expresses data transformation in a system. DFD includes a mechanism to model the data flow. It supports decomposition to illustrate details of the data flows and functions. DFD cannot present information on the operation sequence.

Therefore, it is not a process or procedure modeling method. DFD includes following characteristics:

(1) Supporting the analysis and requirement stage of system design.

(2) A diagramming technique with annotation.

(3) Describing a network of activities/processes of the target system.

(4) Allowing for behaviors of parallel and asynchronous.

(5) Stepwise refinement through hierarchical decomposition of processes.

4.5.1 Modeling Process Of DFD

Creating a DFD is a highly iterative process of gradual refinement. The general steps are:

1. Create a preliminary Context Diagram.
2. Identify Use Cases, i.e., the ways in which users most commonly use the system.
3. Create DFD fragments for each use case.
4. Create a level-0 diagram from fragments.
5. Decompose to level 1, 2, . . .
6. Go to step (1) and revise as necessary.
7. Validate the DFDs with users.

4.5.2 Data Flow Diagramming Rules

While developing DFDs, some rules should be obeyed. These rules include general rules and specific rules for DFD symbols, context diagram and decomposition.

1. General rules.
   - Inputs to a process are always different from outputs.
   - Objects always have a unique name.
   - To keep the diagram uncluttered, you can repeat data stores and sources/sinks on a diagram.

2. Rules for activity/process
   - No process can have only outputs (a miracle).
   - No process can have only inputs (black hole).
   - A process has a verb phrase label.

3. Rules for data store
   - Data cannot be moved directly from one store to another.
   - Data cannot move directly from an outside source to a data store.
   - Data store has a noun phrase label.
4. Rules for external entity
   - Data cannot move directly from a source to a sink.
   - A source/sink has a noun phrase label.
5. Rules for data flow
   - A data flow has only one direction of flow between symbols.
   - A fork means that the same data goes from a common location to two or more processes, data stores or sources/sinks.
   - A join means that the same data comes from any two or more different processes, data stores or sources/sinks to a common location.
6. Rules for context diagram
   - Sources and sinks (external entities) as squares.
   - Main data flows depicted.
   - No internal data stores are shown. They are inside the system.
7. Rules for process decomposition
   - Processes can be decomposed/refined. That means one process can be decomposed into a complete DFD.
   - Lower level processes, data flows and data stores can be added on.
   - Sources and sinks remain on level-1.
   - The level-0 can be used as “abstract”.
   - A data flow can be split into separate data flows on a lower level diagram.

IMPLEMENTATION

CHAPTER 5

IMPLEMENTATION

Implementation is the process of converting a new or revised system design into an operational one. Thus, it can be considered to be the stage in achieving a successful new system and it's vital to assure the user confidence that the proposed new system will never cause impairs and it will be effective. The implementation is not carefully planned and controlled; it can cause chaos. Consultation and a demonstration were given to them about the working of the system.

The aim of the system illustration was to identify any logical working of the system when various combinations of test data were fed. Processing accuracy and reliability checking were also made.

The first task in implementation is the implementation planning, that is deciding on methods to be adopted. The approaches of implementation are direct, parallel. In the first approach, the existing system is rejected and the new system is completely implemented. In parallel approach both existing systems and new systems will be working simultaneously.

5.1 MODULES OF PROJECT

This Section describes the different modules of the project and the pseudocode of the important functions.

a) Student Feedback
b) Results Achieved in Subjects Handled
c) Research Work Contribution
d) Qualification
e) Workshop, Conference & Faculty Development Programs (FDPs)
f) Textbook and Funds
g) Department & College Level Committees & Activities

5.1.1 STUDENT FEEDBACK

Students need to provide each faculty with feedback score by viewing their profileand fill in the form consisting following parameters.

1. Time Management
2. Coverage of Syllabus
3. Control over Class
4. Command on Subject
5. Board Management
6. Communication Skills
7. Interaction with students
8. Motivation for students

Using these parameters, the output of this module will contribute 50% for the final academic progress score of each faculty.

5.1.2 RESULTS ACHIEVED IN SUBJECT HANDLED

The class results achieved by the faculty in the subjects they are handling are considered for the academic progress score calculation. This module contributes 20% for the progress score.

5.1.3 RESEARCH WORK CONTRIBUTION

This module includes about the Paper Published by the faculty and it contributes total 10%. This 10% is divided based on different paper like Journal Paper, International Conference Paper and National Conference Paper.

5.1.4 QUALIFICATION

In this module system evaluate on the basis of their degree completion of faculty and have 5% contribution for the academic progress score. Based on the degree obtained the score is given.

5.1.5 WORSHOPS, CONFERENCE AND FACULTY DEVELOPMENT PROGRAMS

In this module the various workshops, conferences and other programs is considered. If the faculty conducts, any above programs will be rewarded with 5% and if attended any programs then will be rewarded with 2% for their score.

5.1.6 TEXTBOOKS AND FUNDS

Faculty have either to publish a technical paper or be an author to technical textbooks. If they publish and bring potential grants or funds to college then they receive full 5% of score

5.1.7 DEPARTMENT AND COLLEGE LEVEL COMMITTEES AND ACTIVITIES

If faculty will be activity organizer or be a part of college/department level committee member, they are awarded with 5% for the score.

Considering respective scores from each modules the final academic progress score is calculated using the formula:

SCORE = (total score obtained/total score)*100

5.2 WORKING EXAMPLE

NAME: Dr. ABC

QUALIFICTAION: B.E., M.TECH, PhD.

- Student Feedback: 45
- Results Achieved in Subject Handled: 20
- Research Work Contribution: 10
- Qualification: 5
- Workshop and Faculty Development Programs: 5
- Textbook and Funds: 5
- Department and College Level Committees and Activities: 5
- As system rewarded this faculty will have the progress of 95%

i.e., score = (total score obtained/total score)*100 = (95/100)*100 = 95%

5.3 PSEUDOCODE

```
public class  User  {
String name, designation;
public User(String name, String designation) {
        this.name = name;
        this.designation = designation;
    }
}
package com.example.custom_listview;

import androidx.appcompat.app.AppCompatActivity;
import android.annotation.SuppressLint;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import  android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.Toast;

public class UserActivity extends AppCompatActivity {

EditText editTextStudentFB, editTextSubjectResult;
Spinner spinnerPaperPublication, spinnerQualification, spinnerFDPs, spinnerTBPublication,
spinnerFunds, spinnerActivity;
    Button btnSubmit;
    Integer studentFB, totalSR;
```

```
    String pp, qual, fdp, tbp, fund, ca;
    String[] paperPublication = {"Journal", "International Conference", "National Conference"};
    String[] qualification = {"PhD Completed", "PhD Registered", "M.Tech"};
    String[] fdps = {"Conduct", "Attend"};
    String[] textbookPublication = {"Yes", "No"};
    String[] funds = {"Yes", "No"};
    String[] activities = {"Yes", "No"};

    @SuppressLint("WrongViewCast")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_user);
        editTextStudentFB = findViewById(R.id.editTextStudentFB);
        editTextSubjectResult = findViewById(R.id.editTextSubjectResult);
        spinnerPaperPublication = findViewById(R.id.spinnerPaperPublication);
        spinnerQualification = findViewById(R.id.spinnerQualification);
        spinnerFDPs = findViewById(R.id.spinnerFDPs);
        spinnerTBPublication = findViewById(R.id.spinnerTBPublication);
        spinnerFunds = findViewById(R.id.spinnerFunds);
        spinnerActivity = findViewById(R.id.spinnerActivity);
        btnSubmit = findViewById(R.id.btnSubmit);
 ArrayAdapter<String> adapter;
adapter = new ArrayAdapter<String>
(UserActivity.this, android.R.layout.simple_spinner_item, paperPublication);
        adapter = new ArrayAdapter<String>
(UserActivity.this, android.R.layout.simple_spinner_item, qualification);
```

```
        adapter = new ArrayAdapter<String>
(UserActivity.this, android.R.layout.simple_spinner_item, fdps);
        adapter = new ArrayAdapter<String>
(UserActivity.this, android.R.layout.simple_spinner_item, textbookPublication);
        adapter = new ArrayAdapter<String>
(UserActivity.this, android.R.layout.simple_spinner_item, funds);
        adapter = new ArrayAdapter<String>
(UserActivity.this, android.R.layout.simple_spinner_item, activities);


        adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        spinnerPaperPublication.setAdapter(adapter);
        spinnerQualification.setAdapter(adapter);
        spinnerFDPs.setAdapter(adapter);
        spinnerTBPublication.setAdapter(adapter);
        spinnerFunds.setAdapter(adapter);
        spinnerActivity.setAdapter(adapter);


        spinnerPaperPublication.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {
            @Override
            public void onItemSelected(AdapterView<?> parent, View view, int position, long id)
{
                paperPublication[position] = parent.getItemAtPosition(position).toString();
            }
            @Override
            public void onNothingSelected(AdapterView<?> parent) {
            }
        });
```

```
        spinnerQualification.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {
            @Override
            public void onItemSelected(AdapterView<?> parent, View view, int position, long id)
{
                qualification[position] = parent.getItemAtPosition(position).toString();
            }
            @Override
            public void onNothingSelected(AdapterView<?> parent) {
            }
        });
        spinnerFDPs.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
            @Override
            public void onItemSelected(AdapterView<?> parent, View view, int position, long id)
{
                fdps[position] = parent.getItemAtPosition(position).toString();
            }
            @Override
            public void onNothingSelected(AdapterView<?> parent) {
            }
        });
        spinnerTBPublication.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {
            @Override
            public void onItemSelected(AdapterView<?> parent, View view, int position, long id)
{
                textbookPublication[position] = parent.getItemAtPosition(position).toString();
            }
```

```
        @Override
        public void onNothingSelected(AdapterView<?> parent) {
        }
    });
    spinnerFunds.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
        @Override
        public void onItemSelected(AdapterView<?> parent, View view, int position, long id)
{
            funds[position] = parent.getItemAtPosition(position).toString();
        }
        @Override
        public void onNothingSelected(AdapterView<?> parent) {
        }
    });
    spinnerActivity.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener()
{
        @Override
        public void onItemSelected(AdapterView<?> parent, View view, int position, long id)
{
            activities[position] = parent.getItemAtPosition(position).toString();
        }
        @Override
        public void onNothingSelected(AdapterView<?> parent) {
        }
    });
    btnSubmit.setOnClickListener(new View.OnClickListener() {
        }
});
```

TESTING

CHAPTER 6

TESTING

Testing can be stated as the process of verifying and validating that a product or application is bug free, meets the technical requirements as guided by its design and development and meets the user requirements effectively and efficiently with handling all the exceptional and boundary cases. Hence testing performs a very critical role for quality assurance and ensuring the reliability of the software. Errors were found and corrected by using the following testing steps and correction was recorded for future references. Thus, a series of testing was performed on the system before it was ready for implementation.

Test data are inputs which have been detect the behavior within it during the failure modes within the software is kindly generally not feasible because of the process of software testing , multiple inputs are taken and each test data are considered and they are verified, and rather test cases are written for both successful ones as well as failure ones and generally most feasible data is taken, the overall software testing is taken into those considering within the both of the process in which they need to satisfy both of the process verification and validation.

It can be objects, variables, functions or any other multiple modules. during the process of system testing the overall composed components are to be integrated to form one complete system, hence testing must be done in such a way that they meet all the functional specification as the system requirements must meet up the user requirements in the overall approach, and must ensure that the system must not behave in the unexpected way, test data are those where the input is considered and tested with it, and the test cases are those which probably of these operates on system specifications, the system made available they will ensure.

Test data are inputs which have been detect the behavior within it during the failure modes within the software is kindly generally not feasible because of the process of software testing , multiple inputs are taken and each test data are considered and they are verified, and rather test cases are written for both successful ones as well as failure ones and generally most feasible data is taken, the overall software testing is taken into those considering within the both of the process in which they need to satisfy both of the process verification and validation.

- **Verification**: Verification is done with the help of specified document, It verifies that the software those are being developed and simultaneously implemented specific functions in order to design the document.
- **Validation**: Validation has to be done in order to verify that they must satisfy the process of the software requirement specification document, this software which is developed which can be implemented specified simultaneously.

6.1 TESTING PROCESS

The testing process is a vital process within a single monolithic unit, these must not be tested which is again a testing process which is processed throughout the simple components and then they are integrated to one full system into one system. The testing process is done step bystep and carried out in increment fashion by incrementing the sequence, these are the steps with implementation part, where the error programs may be within the lighter stage if in case of errors is checked and corrected as they must meet all the needs of the functional requirements. This helps the overall process of testing category in this way they are helpful.

➢ AIM OF TESTING

The overall purpose of testing is to discover the errors, it is the process of identifying the errors in prepared components this is the overall process which is trying to serve within the conceiving fault or weakening the working product, it is to check the functions of the components available within the sub-assemblies in a finished products, it has to check the available functions with conceivable fault as well as weakening and checks the available assemblies within the finished products it is to exercise the software with the intentionthat the software of ensuring the requirements with that of the user required the values with those meeting the requirements, which falls or fail within the user requirements and do not fail in any unacceptable manner, there are many types of test, each and every test have their own specific testing requirement.

6.1.1 UNIT TESTING

Unit testing is all that which usually involves in maintaining multiple designs within those of the available test cases within those of the available programming language which has many functional property, which has many input which is valid throughout the available inputs, all

the branches within the internal codes there are many flow which is validated these testing must be done individually then all the test cases are then integrated and added to one system, then each valid and invalid inputs are taken these are multiple branches, these testing are done using the individual software units within the applications, which can rely on the available knowledge in the construction and is important, unit testing performing basic test at components level and within the specific business processes, applications within the system configurations, within that of unit testing where all the documents are verified with the available process that performs the documents, that contains that clearly about the used inputs as well as the expected results.

6.1.2 INTEGRATION TESTING

The integration of the tests are designed to those of the software components to determine carefully the individual components of the code and to check the main software after validating all the validated components are integrated to one complete system and then they are finalized, testing is the events that is used to determine the programs it is more concerned with the basic screens that has multiple outcomes of the fields and screens then integrating it with the software associated with those of running of those programs, and the function is event driven and the components are tested and aimed to correct the exposed problems and to maintain the consistency.

6.1.3 FUNCTIONAL TESTING

The functional testing is the vital process where every single functions is tested which has too many systematic demonstrations that are functioned and tested which is made available that looks into all technical requirements within the overall documentation and these are user manuals which is considered for many aspects such as user manuals and technical requirements.

- **Testing is allocated on the following steps:**

Validity : classification of valid input must be accepted.

Not accepted : classification of invalid input must be rejected.

Allocations : classification functions must be exercised.

Outsourcing : classifying the outputs must be exercised.

Procedure structures: classifying the procedures must be invoked.

The overall functions of those testing is then focused testing within key functions or special test cases these in addition are the coverage pertaining to identify the business process that must be considered in the testing, these functions are then complete which can have additional tests this is in turn identified within the data fields that can have successive process which is considered and then they are complete with that of the additional testing which can be identified and that of the effectiveness in obtaining the values, fields and some complete additional tests which can have effective values of those input test validated and determined. Which gives the confidence on the new systems which ensures that the system works effectively and efficiently, just according to the user windows. There is that existing long time process which has an overall proposed system which is developed within applet window which is caused by a long time process where the transmissions but the system developed that has a user friendly tool that has a menu based interface for the graphical user interface as well. Coding as well as testing is analyzed to being within the installation on the necessary system.

6.1.4 SYSTEM TESTING

The system testing usually ensures in the entire integration which meets the software requirements, that usually are known and then they are predictable results, an example of the system testing which is oriented within this system that is based on the description as well as flows that emphasize the process links and the integration points.

6.1.5 WHITE BOX TESTING

The white box testing in which the software testing has knowledgeable within the inner workings, software tester which is structure and within the language structure and also used to test the areas which cannot be reached from a black box level.

6.1.6 BLACK BOX TESTING

The black box is tested without any prior knowledge within the inner workings of the structure or the language of those modules being tested within the black box tests that are if many kinds, within the written source of documents which is tested, that the software documents, responds to the output without considering the software.

6.2 TESTCASES

Test cases are used to test the pair of data to the program sets and to verify if at all the users are getting the desired output, it is usually used to set a pair of data assets for each of the available variables, it has multiple sets of available data with two or more notions within any one of the executions rather they are much more elaborated in this chapter, with various test cases and also helps in generating test data and easily validation could be completed.

Those programs required as well as the tested data help them in constructing multiple test data, execution of the test cases is little time consuming, but it's an essential phase were the overall phases have to change the functions within the scenario. Usually, the testers generate few test cases and then they can try executing the program if in case if any of the codegenerate the errors these can generally make use of the available program with these characters.

Then the results are thus obtained, then the software testers usually discuss if there is any kind of error generated and discussion is done are not, then the error correction will be done and then the testers enter the debugging phase. Then the test cases are formally manipulated and developed for the required system.

1. ADMIN LOGIN

Test Case No.	1
Name of Test	User login
Test Case Description	Login is done by user, by giving valid password
Sample Input	Text data
Expected Output	Login successful
Actual Output	As expected
Remarks	Pass
Comments	Working properly

Table.6.1: Test Case 1 – Admin Login

2. HOME PAGE

Test Case No.	2
Name of Test	Home page
Test Case Description	Home page of Academic Progress Score
Sample Input	Text data and digits
Expected Output	It should accept correctly
Actual Output	As expected
Remarks	Pass
Comments	Working properly

Table.6.2: Test Case 2 – Home Page

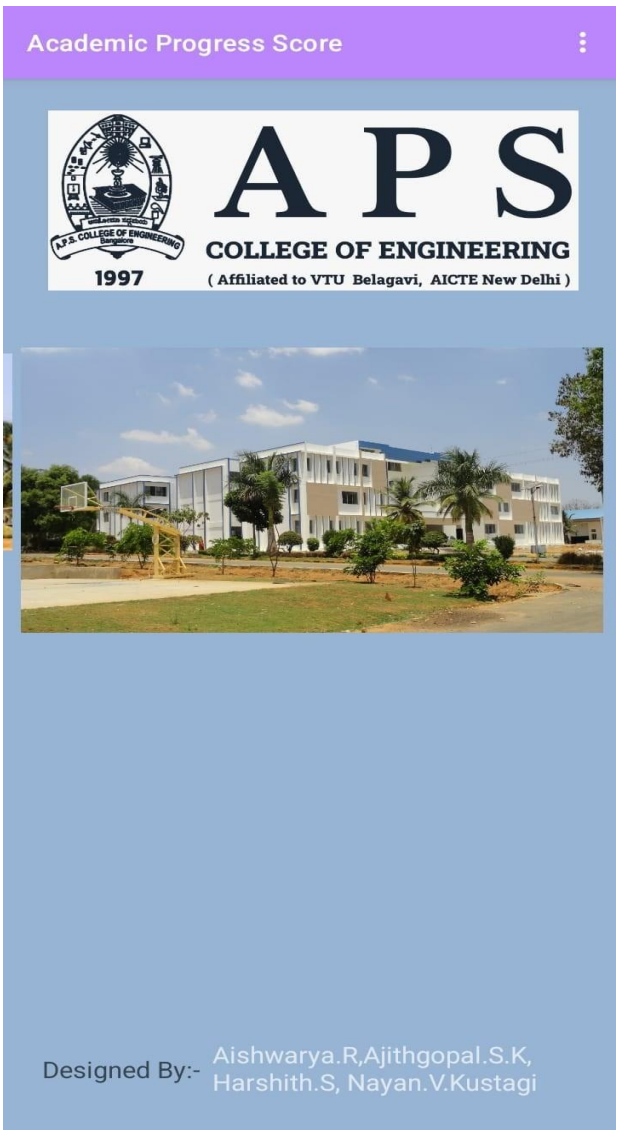
RESULTS AND SNAPSHOTS

CHAPTER 7

RESULTS AND SNAPSHOTS

7.1 HOME PAGE

As the Academic Progress Score application opens in the android phone it will show the home page where we can view the pictures of the college. Here we can select to login into the application or view the faculty profile.



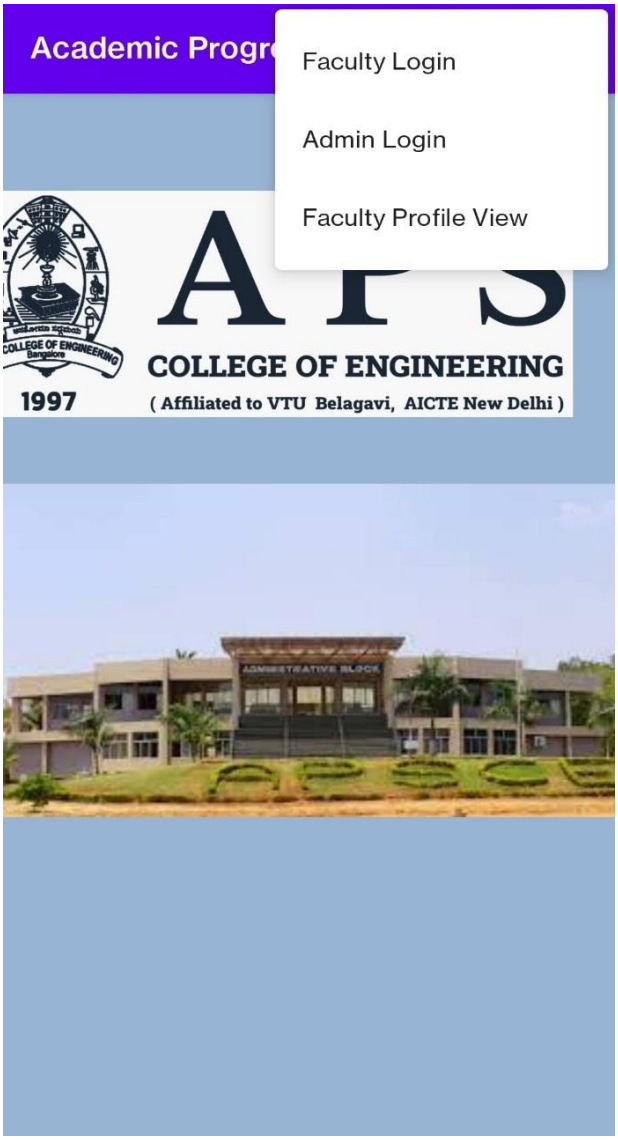


Fig.7.1: Home Page

7.2 ADMIN LOGIN PAGE

Using this login page an authorized admin can login into the system by entering valid credentials. Admin can login to the system and add, delete or update the faculty profile.

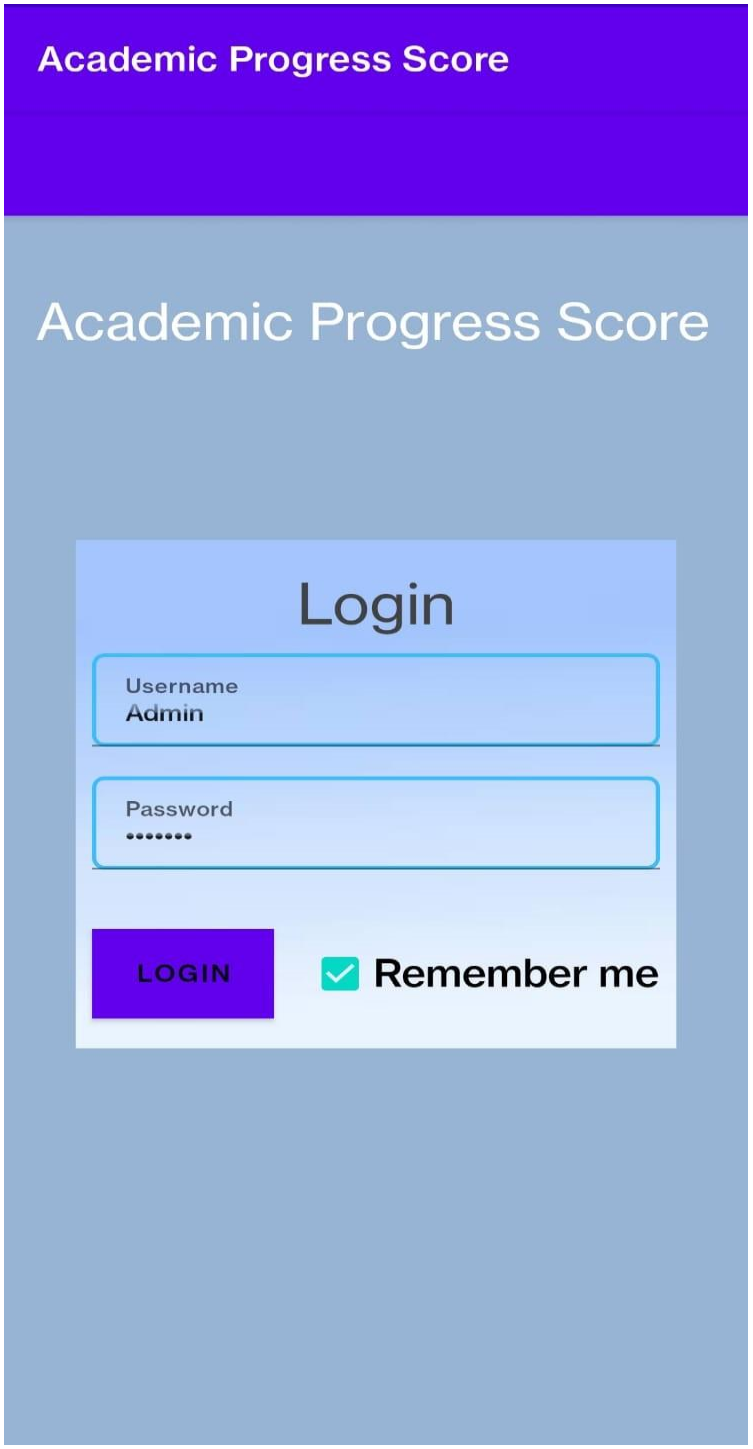


Fig.7.2: Admin Login Page

7.3 FACULTY LOGIN PAGE

Each faculty is given with credentials to login into the system. After the credentials is verified, the faculty can view their respective profiles.

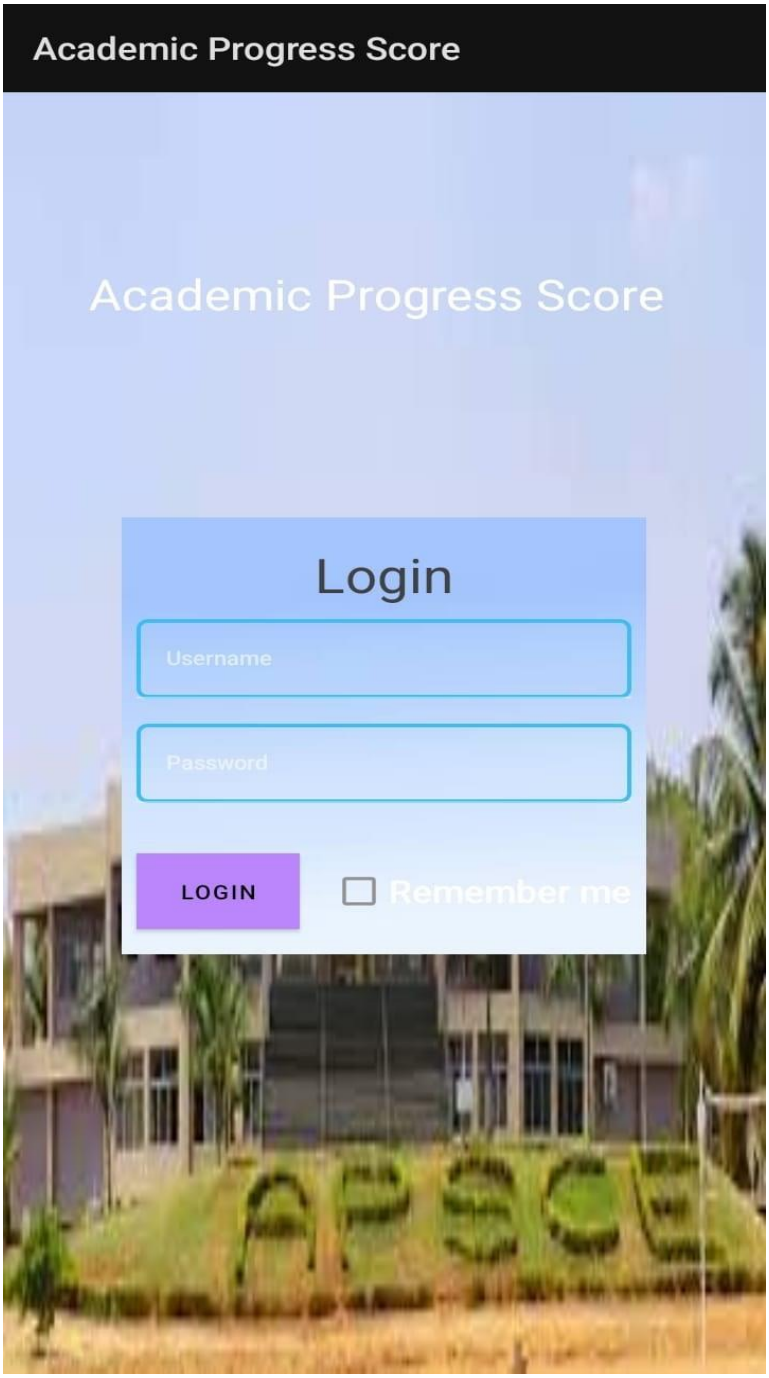


Fig.7.3: Faculty Login Page

7.4 FACULTY PROFILE VIEW

This page displays all the faculty's profile. Students can view each faculty profile before they provide feedback score to them. Students need not to login to the system to provide the feedback score.

Fig.7.4: Faculty Profile View

7.5 STUDENT'S FEEDBACK FORM

As the student select the profile of the faculty to which they want to give the feedback score, application will direct them to the feedback form. This form consists of various parameters based on which student should give the feedback score and submit.

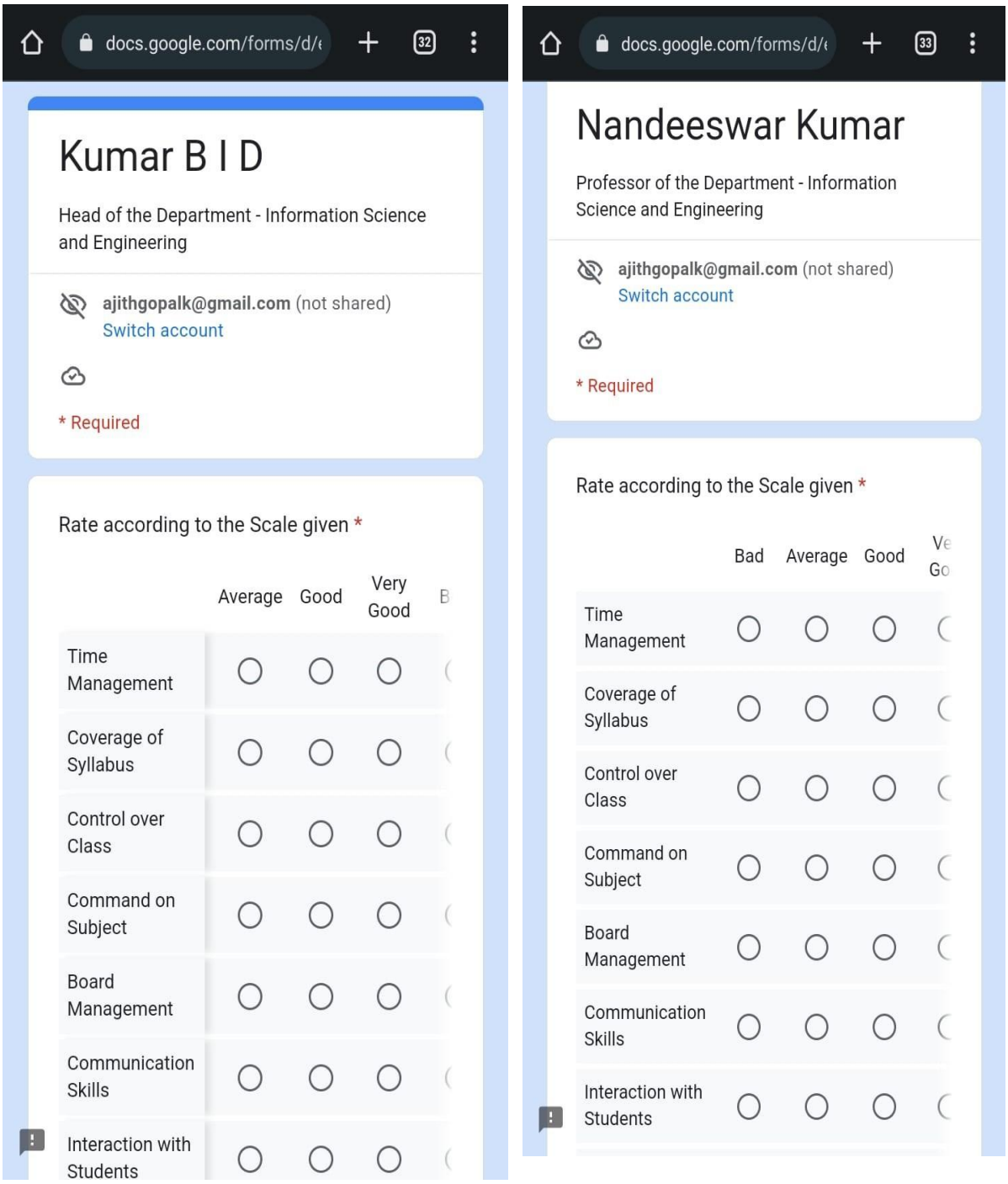


Fig.7.5: Feedback Form

7.6 ACADEMIC PROGRESS SCORE

Faculty should login to the system to view their profile along with the Academic Progress Score. This is the final result produced by the application after calculating the academic progress score for each faculty.

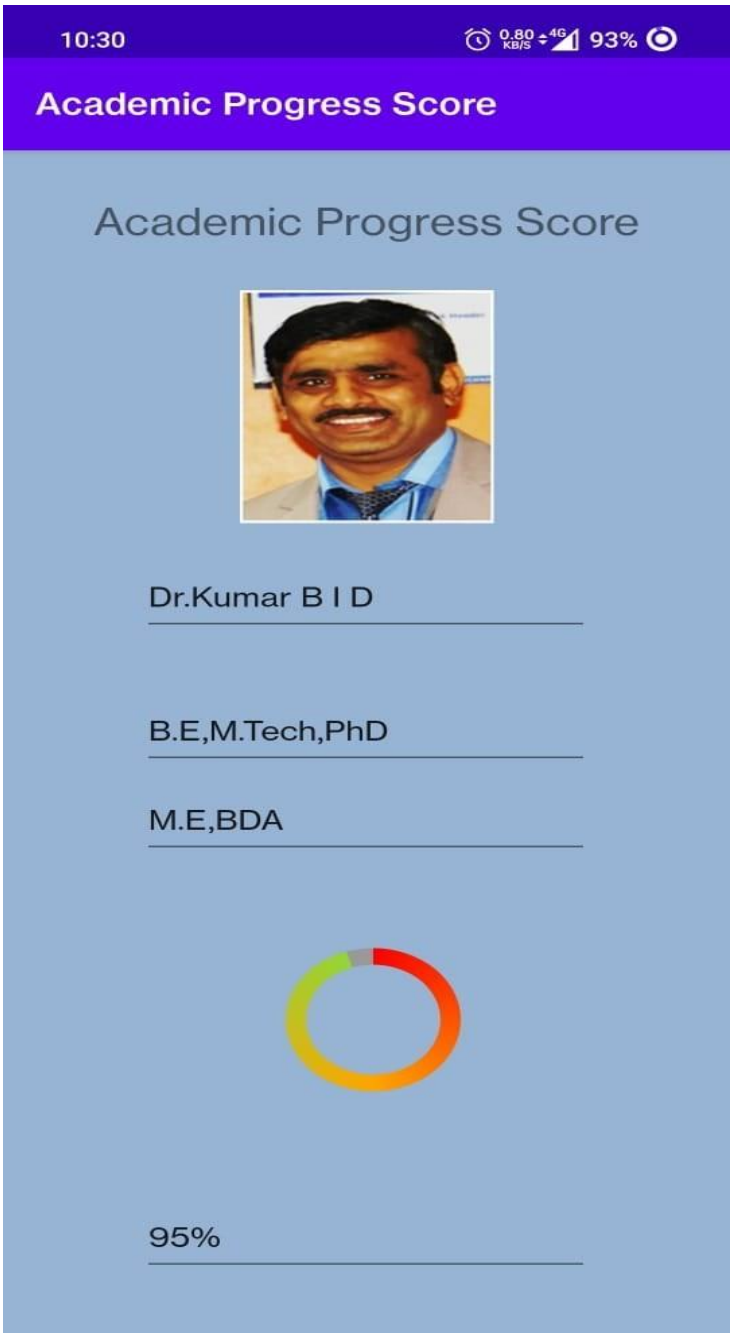


Fig.7.6: Faculty's Profile with their Academic Progress Score

CONCLUSION

CHAPTER 8

CONCLUSION

Academic Progress Score is an application developed to facilitate to automate and easy processing of collecting student feedback. This student feedback score is utilized to calculate the respective academic progress score of each faculty. Each faculty can login to the system and view their score along with their profile.

The academic progress score is calculated by the application considering both the feedback given by the students about each faculty and achievements, qualification and results achieved by the faculty.

This academic progress score can be used by the faculty to improve their skills and overall development of individual. This application can be even used by the educational institutions to take view on the faculty and improve the standard of education. Collecting the feedback provides a way for students to keep up their opinion in front of the institution and using this review of students the institution can adopt the various methods to improve the quality of the education.

8.1 FUTURE ECHANCEMENT

Future enhancements that can be included is timetable. This is to enable students and teachers to view their respective class schedule within the application.

Academic progress score application is restricted to the educational domain. Further this system can be by different business organization, industries and other working sectors to collect the feedback on their employees and report the performance of each employee. This analysis of each employee performance would help the organizations and industries to review their employee and provide necessary training required for the employee. By reviewing about the employees would help the organization to improve the services they provide and help the organization to develop in future.

REFERENCES

REFERENCES

[1] M. Tarare1, M.Manwani2, A. Paidlewar3, S. Maturkar4, P. Chaudhari5, J. V. Shiral6," Feedback Management System for Evaluating And Generating Monthly Report", International Journal of Emerging Technology and Advanced Engineering, Volume 4, Issue 3, March 2014.

[2] J. Hatie, H. Timperley, "The power of feedback", J. Review of Educational Research, 87(1), pp. 81-112, 2007.

[3] Software Engineering by Roger S. Pressman (Tata MC-GRAW hill, 5th edition).

[4] Data Base Management System by Raghu Rama Krishnan (Tata MC-GRAW hill, 3rd edition).

[5] Hwang,G. J."A Conceptual Map Model for Developing Intelligent Tutoring Systems". International Journal of Computers and Education, 40, no. 3 (2003). 217-235.

[6] Beatty, D.I.,Gerace, J.W.,Leonard, J.W.,Dufresene, J.R. (2006). Designing effective questions for classroom response system teaching. American Association of physics teachers.74(1),31-39.

[7] M. Zibran, F. Eishita and C. Roy, "Useful but usable? factors affecting the usability of APIs", Proc. 18th Working Conf. Reverse Eng., pp. 151-155, 2011.

[8] J. Businge, A. Serebrenik and M. van de Brand, "Eclipse API usage: The good and the bad", Softw. Quality J..

[9] M. Linares-Vásquez, G. Bavota, C. Bernal-Cárdenas, M. Di Penta, R. Oliveto and D. Poshyvanyk, "API change and fault proneness: A threat to the success of Android apps", Proc. 9th Joint Meeting Found. Softw. Eng., pp. 477-487, 2013.

[10] M. Fisher, M. Pinzger and H. Gall, "Populating a release history database from version control and bug tracking systems", Proc. 19h Int. Conf. Softw. Maintenance, pp. 23-32, 2003.

[11] Kim, H., Choi, B., W.E. Wong: Performance testing of mobile applications at the unit test level. In: 3rd IEEE International Conference on Secure Software Integration and Reliability Imrovement (2009).

[12] D. Kurokawa, O. Lev, J. Morgenstern, and A. D. Procaccia, “Impartial Peer Review,” Technical Report, 2014. Available: http://www.cs.cmu.edu/ arielpro/papers/peer review.pdf

[13] M. Kandori, “Social norms and community enforcement,” Review of Economic Studies, vol. 59, no. 1, pp. 63 – 80, 1992.

[14] Y. Zhang, J. Park, and M. van der Schaar, “Rating protocols for online communities,” ACM Transactions on Economics and Computation, vol. 2, no. 1, 2013.

[15] Y. Xiao and M. van der Schaar, “Socially-optimal design of service exchange platforms with imperfect monitoring,” Technical Report, 2013.

Fig.8.1: A picture of our team with our project guide - Prof. Pallavi H B, Asst. Professor, Dept of ISE, A P S College Of Engineering.
Teammates name from Left to right:
Ajithgopal S Kulkarni, Harshith S, Aishwarya R, Nayan V Kustagi.