FACEGUARD:
AN APPLICATION TO PROVIDE SECURE PHOTO SHARING IN SOCIAL NETWORK

A PROJECT REPORT

Submitted by

AISWARYA T	**VVT15CS005**
ANJU K	**VVT15CS010**
MREDULA PK	**VVT15CS020**

to

the APJ Abdul Kalam Technological University

in partial fulfillment of the requirements for the award of the Degree

of

Bachelor of Technology in*Computer science*

Department of Computer Science

Vedavyasa Institute of Technology

Karadparamba

JUNE 2019

DECLARATION

We undersigned hereby declare that the project report **"AN APPLICATION TO PROVIDE SECURE PHOTO SHARING IN SOCIAL NETWORK"**, submitted for partial fulfillment of the requirements for the award of degree of Master of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by **Ms.AISWARYA T (VVT15CS005),Ms.ANJU K (VVT15CS010) ,Ms.MREDULA PK (VVT15CS020)** under supervision of **Ms.JANCY MANOHARAN**. This submission represents our ideas in our own words and where ideas or words of others have been included, we have adequately and accurately cited and referenced the original sources. We also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place Name and Signature of Students

Date

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Vedavyasa institute of technology

Karadparamba

CERTIFICATE

This is to certify that the report entitled “**AN APPLICATION TO PROVIDE SECURE PHOTO SHARING IN SOCIAL NETWORK”** submitted by **Ms.AISWARYA T (VVT15CS005), Ms.ANJU K (VVT15CS010), Ms.MREDULA PK (VVT15CS020)** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in **Bachelor of Technology in Computer Science and Engineering** is a bonafide record of the project work carried out by him/her under my/our guidance and supervision.. This report in any form has not been submitted to any other University or Institute for any purpose.

Name & Signature of the Project Guide Name & Signature of the HOD

ACKNOWLEDGEMENT

We are greatly indebted to God almighty for being the guiding light throughout with his abundant grace and blessing that strengthened me to do this endeavor with confidence.

We express our heartfelt gratitude towards ***Dr.SANGEETHA.S Principal*** *Vedavyasa institute of technology for extending all the facilities required for doing our project work***.**

We would also like to thank ***Prof.KAVITHA MURUGESAN,*** *Head of Department of computer science and Engineering, for providing constant support and encouragement.*

Now we extend my sincere thanks to our guide ***Ms.JANCY MANOHARAN,*** *Assistant professor Department of Computer Science and Engineering, for guiding us in our work and providing timely advices.*

We also like to thank all other teaching and non-teaching staff in our department for their kind co-operation.

Last but not the least, we extend our heartfelt gratitude to our parents and friends for their support and assistance.

AISWARYA T

ANJU K

MREDULA P K

CONTENTS

Contents. Page No.

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

FR	: face recognition
OSN	: online social network
DFD	: data flow diagram
IDE	: integrated development environment
SQL	: structured query language
ADT	: android developer tools

ABSTRACT

Social network play on an important role in our lives, security and privacy is a must. Billions of pictures are uploaded daily to social networks and with them, parts of our private life are disclosed. Unfortunately ,it may leak users privacy if they are allowed to post, comment, and tag a photo freely. In this work we propose a practical solution for secure photo sharing on social network. To prevent possible privacy leakage of a photo, we design a mechanism to enable each individual in a photo be aware of the posting activity and participate in the design making on the posting. For this purpose we need a efficient FR system that can recognize everyone in this photo

CHAPTER 1
INTRODUCTION

1.1 GENERAL BACKGROUND

OSNS have become integral part of our daily life and has profoundly changed the way we interact with each other, fulfilling our social needs–the needs for social interactions, information sharing, appreciation and respect. It is also this very nature of social media that makes people put more content, including photos,

over OSNs without too much thought on the content. However, once something, such as a photo, is posted online, it becomes a permanent record, which may be used for purposes we never expect. For example, a posted photo in a party may reveal a connection of a celebrityto a mafia world. Because OSN users may be careless in posting content while the effect is so far-reaching ,privacy protection over OSNs becomes an important issue. When more functions such as photo sharing and tagging are added, the situation becomes more complicated. For instance, nowadays we can share any photo as we like on OSNs, regardless of whether this photo contains other people (is a co-photo) or not. Currently there is no restriction with sharing of co-photos, on thecontrary, social network service providers like Facebook are encouraging users to post co-photos and tag their

friends in order to get more people involved. However, what if the co-owners of a photo are not willing to share this photo? Is it a privacy violation to share this cophoto without permission of the co-owners? Should the co-owners have some control over the co-photos?

To answer these questions, we need to elaborate on the privacy issues over OSNs. Traditionally, privacy is regarded as a state of social withdrawal. According to Altman’s privacy regulation theory , privacy is a dialectic and dynamic boundary regulation process where privacy is not static but “a selective control of access to the self or to ones group”. In this theory, “dialectic”refers to the openness and closeness of self to others and “dynamic” means the desired privacy level changes with time according to environment. During the process of privacy regulation, we strive to match the achieved privacy level to the desired one. At the optimum privacy level, we can experience the desired confidence when we want to hide or enjoy the desired attention when we want to show. However, if the actual level of privacy is greater than the desired one, we will feel lonely or isolated; on the other hand, if the actual level of privacy is smaller than the desired one, we will feel over-exposed and vulnerable.

Unfortunately, on most current OSNs, users have no control over the information appearing outside their profile page. In, Thomas, Grier and Nicol examine how the lack of joint privacy control can inadvertentlyreveal sensitive information about a user. To mitigatethis threat, they suggest Facebook’s privacy model to be adapted to achieve multi-party privacy. Specifically, there should be a mutually acceptable privacy policy determining which information should be posted and

shared. To achieve this, OSN users are asked to specify a privacy policy and a exposure policy. Privacy policy is used to define group of users that are able to access a photo when being the owner, while exposure policy is used to define group of users that are able to access when being a co-owner. These two policies will together mutually specify how a co-photo could be accessed. However,

before examining these policies, finding identities in cophotos is the first and probably the most import step. In the rest of this paper we will focus on a RF engine to find identities on a co-photo. FR problems over OSNs are easier than a regular FR problem because the contextual information of OSN could be utilized for FR. For example, people showing up together on a co-photo are very likely to be friends on OSNs, and thus, the FR engine could be trained to recognize social friends (people in social circle) specifically. Training techniques could be adapted from the off-the-shelf FR training algorithms, but how to get enough training samples is tricky. FR engine with higher recognition ratio demands more training samples (photos of each specific person), but online photo resources are often insufficient. Users care about privacy are unlikely to put photos online. Perhaps it is exactly those people who really want to have a photo privacy protection scheme. To break this dilemma, we propose a privacy-preserving distributed collaborative training system as our FR engine. In our system, we ask each of our users to establish a private photo set of theirown. We use these private photos to build personal FR engines based on the specific social context and promise that during FR training, only the discriminating rules are revealed but nothing else. With the training data (private photo sets) distributed among users, this problem could be formulated as a typical secure multi-party computation problem. Intuitively, we may apply cryptographic technique to protect the private photos, but the computational and communication cost may pose a serious problem for a large

OSN. In this paper, we propose a novel consensusbased approach to achieve efficiency and privacy at the same time. The idea is to let each user only deal with his/her private photo set as the local train dataand use it to learn out the local training result. Afterthis, local training results are exchanged among users to form a global knowledge. In the next round, each user learns over his/hers local data again by taking the global knowledge as a reference. Finally the information will be spread over users and consensus could be reached. We show later that by performing local learning in parallel, efficiency and privacy could be achieved at the same time

1.2 OBJECTIVE

The project entitled ”Faceguard” deals with face detection using machine learning.Once the face is identitified the application sends a notification to the owner of the face that have been identified. Then the owner thereby provide the permission whether to upload their face or not. The main objective of this application is to provide security while sharing photos in social network.

CHAPTER 2
LITERATURE SURVEY

2.1 Photo privacy in a tagged world

A survey was conducted in to study the effectiveness of the existing countermeasure of untagging and shows that this countermeasure is far from satisfactory: users are worrying about offending their friends when untagging.

As a result, they provide a tool to enable users to restrict others from seeing their photos when posted as a complementary strategy to protect privacy. However,

this method will introduce a large number of manual tasks for end users. Photo tagging is a popular feature of many social network sites that allows users to annotate uploaded images with those who are in them, explicitly linking the photo to each person's profile. In this paper, we examine privacy concerns and mechanisms surrounding these tagged images. Using a focus group, we explored the needs and concerns of users, resulting in a set of design considerations for tagged photo privacy. We then designed a privacy enhancing mechanism based on our findings, and validated it using a mixed methods approach. Our results identify the social tensions that tagging generates, and the needs of privacy tools to address

the social implications of photo privacy management.

Author Keywords

Privacy, photo sharing, social network sites, Facebook.

2.2 Adaptive Conflict Resolution for Multi-Party Privacy Policies for Social Media

Things shared through Social Media may impact more than one customer's security — e.g., photos that depict diverse customers, comments that determine diverse customers, events in which different customers are invited, et cetera. The nonappearance of multi-social occasion insurance organization support in current standard Social Media bases makes customers not ready to fittingly control to whom these things are truly shared or not. Computational frameworks that can mix the security slants of different customers into a single methodology for a thing can deal with this issue. In any case, mixing diverse customers' insurance slants is not a straightforward undertaking, since security slants may strife, so techniques to decide conflicts are required. Likewise, these systems need to consider how customers' would truly go to an understanding around a response for the conflict remembering the deciding objective to propose courses of action that can be agreeable by most of the customers impacted by the thing to be shared. Current systems are either unreasonably asking for or simply consider settled strategies for gathering security slants. In this paper, we propose the essential computational framework to decide conflicts for multi-party security organization in Social Media that can acclimate to different circumstances by exhibiting the concessions that customers make to accomplish a response for the disputes. We also present results of a customer study in which our proposed instrument beat other existing philosophies to the extent how regularly every strategy composed customers' behavior.

Keywords: Multi-Party Privacy Policies, Security, Social Media.

2.3 Maintaining Privacy on Photo Sharing

Photo sharing refers to the transfer or publishing of a user’s digital photos online and the website which provides such acquaintances offer services such as hosting, uploading, sharing and managing of photos through online system. This function provides the upload and display of images through the websites and applications. The usage of online photo galleries including photo blogs is increased. The photo sharing term can be set up and managed by individual users. It means that other users can view but not essentially download the photos, users being able to select

different copy-right options for their photos. Unfortunately, it may reveal users privacy if they are permitted to post, comment, and tag a photo liberally. Communication feature on social network is not secured. Users can not send confidential message through social chat. To address these problem, this project proposes an efficient facial recognition system that can recognize everyone in the photo. Online photo sharing applications have become popular. To share photos with a range of people, it provides users various new and innovative alternatives. The photo sharing feature is incorporated in many social networking sites which allow users to post photo for their loving ones, families and friends. Encryption technique is applied for communication between users. AES algorithm is implemented for encrypting chat. For users of social networking sites such as Facebook, this system focuses on the privacy concerns and needs of the users, at the same time explores ideas for privacy protection mechanisms. By considering users current concerns and behaviors, the tool can be designed as per the user’s desire which they can adoptand then can be motivated to use. **Key words:** Social network, Photo Privacy, Secure Multi-Party Computation, Collaborative Learning

2.4 Unpacking "Privacy" for a Networked World

Although privacy is broadly recognized as a dominant concern for the development of novel interactive technologies, our ability to reason analytically about privacy in real settings is limited. A lack of conceptual interpretive frameworks makes it difficult to unpack interrelated privacy issues in settings where information technology is also present. Building on theory developed by social psychologist Irwin Altman, we outline a model of privacy as a dynamic, dialectic process. We discuss three tensions that govern interpersonal privacy management in everyday life, and use these to explore select technology case studies drawn from the research literature. These suggest new ways for thinking about privacy in sociotechnical environments as a practical matter.

Keywords

Privacy, surveillance, monitoring, access regulation, boundary management, disclosure, social psychology

2.5 Autotagging Facebook: Social Network Context Improves Photo Annotation

Most personal photos that are shared online are embedded in some form of social network, and these social networks are a potent source of contextual information that can be leveraged for automatic image understanding. In this paper, we investigate the utility of social network context for the task of automatic face recognition in personal photographs. We combine face recognition scores with social context in a conditional random field (CRF) model and apply this model to label faces in photos from the popular online social network Facebook, which is now the top photo-sharing site on the Web with billions of photos in total. We demonstrate that our simple method of enhancing face recognition with social network context substantially increases recognition performance beyond that of a baseline face recognition system. An increasing number of personal photographs are uploaded to online social networks, and these photos do not exist in isolation. Each shared image likely arrives in a batch of related photos from a trip or event; these are then associated with their photographer and broadcast out to that photographer's hundreds of online friends, and they join a collection of billions of other photographs, some of which have been manually labeled with the people they contain and other information. Social networks are an important source of image annotations, and they also provide contextual information about the social interactions among individuals that can facilitate automatic image understanding.

KEYWORD: conditional random field(CRF), social network

2.6 Collaborative Face Recognition for Improved Face Annotation in Personal Photo Collections Shared on Online Social Networks

Using face annotation for effective management of personal photos in online social networks (OSNs) is currently of considerable practical interest. In this paper, we propose a collaborative face recognition (FR) framework, improving the accuracy of face annotation by effectively making use of multiple FR engines available in an OSN. Our collaborative FR framework consists of two major parts: selection of FR engines and merging (or fusion) of multiple FR results. The selection of FR engines aims at determining a set of personalized FR enginesthat are suitable for recognizing query face images belonging to a particular member of the OSN. For this purpose, we exploit both social network context in an OSN and social context in personal photo collections. In addition, to take advantage of the availability of multiple FR results retrieved from the selected FR engines, we devise two effective solutions for merging FR results, adopting traditional techniques for combining multiple classifier results. Experiments were conducted using 547 991 personal photos collected from an existing OSN. Our results demonstrate that the proposed collaborative FR method is able to significantly improve the accuracy of face annotation, compared to conventional FR approaches that only make use of a single FR engine. Further, computational cost and comes with a design that is suited for deployment in a decentralized OSN.

Key words—Collaboration, face annotation, face recognition, online social network, personal photos, social context.

2.7 Propagation of Facial Identities in a Social Network

We address the problem of automated face recognition on a social network using a loopy belief propagation framework. The proposed approach propagates the identities of faces in photos across social graphs. We characterize performance in terms of structural properties of a social network. We propose a distance metric defined using face recognition results for detecting hidden connections. The result demonstrates that the constraints imposed by the social network have the potential to improve the performance of face recognition methods. The result also shows it is possible to discover hidden connections in a social network based on face recognition. A social network reflects the relationship structure among entities. It typically consists of different kinds of information, such as text, images and videos. The structure of a social network has been shown to play an important role in many fields such as marketing and epidemiology. It is an open question in face recognition, and computer vision in general, how algorithms can be adapted to solve vision problems on a social network. In this paper we address this question for face recognition algorithms. Millions of facial images are uploaded to social network websites. The faces in these images are usually taken with point and shoot cameras or cell phones in unconstrained environments. This class of images is among the most challenging for face recognition. In this paper, we study how the structure of a social network can be used to improve the performance of automated face recognition algorithms.

CHAPTER 3

Automated photo tagging in Facebook

The Facebook platform allows for users to upload pictures and tag people on these images. Here tagging means the process of placing a box around the person's faces in the image and identifying them. Figure shows an example of a placed tag. In Facebook'scurrent tagging implementation, a constant-sized rectangle is placed over a persons face, and only the center of the box is stored. To date, these tags must be manually placed by users who recognize the faces of their friends in an image. We want to improve on this and have the tagging operation performed automatically when a picture is added to the platform. However, the tags already placed by users are very useful for the learning process, since they supply the algorithm with numerous training examples of a person's face (if we were limited to profile pictures, we would have only one training image per person).

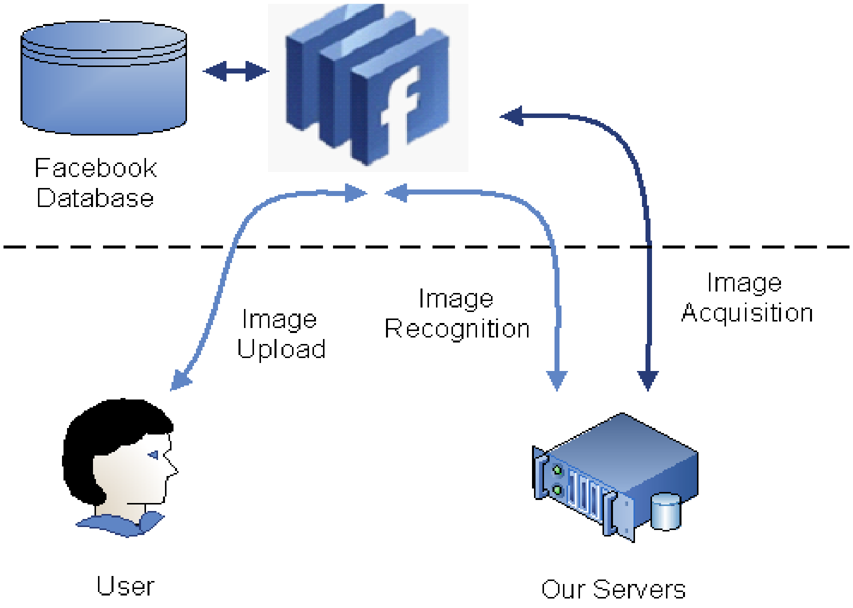


Fig 1 : schematic of the facebook crawler

CHAPTER 4

FACEGUARD

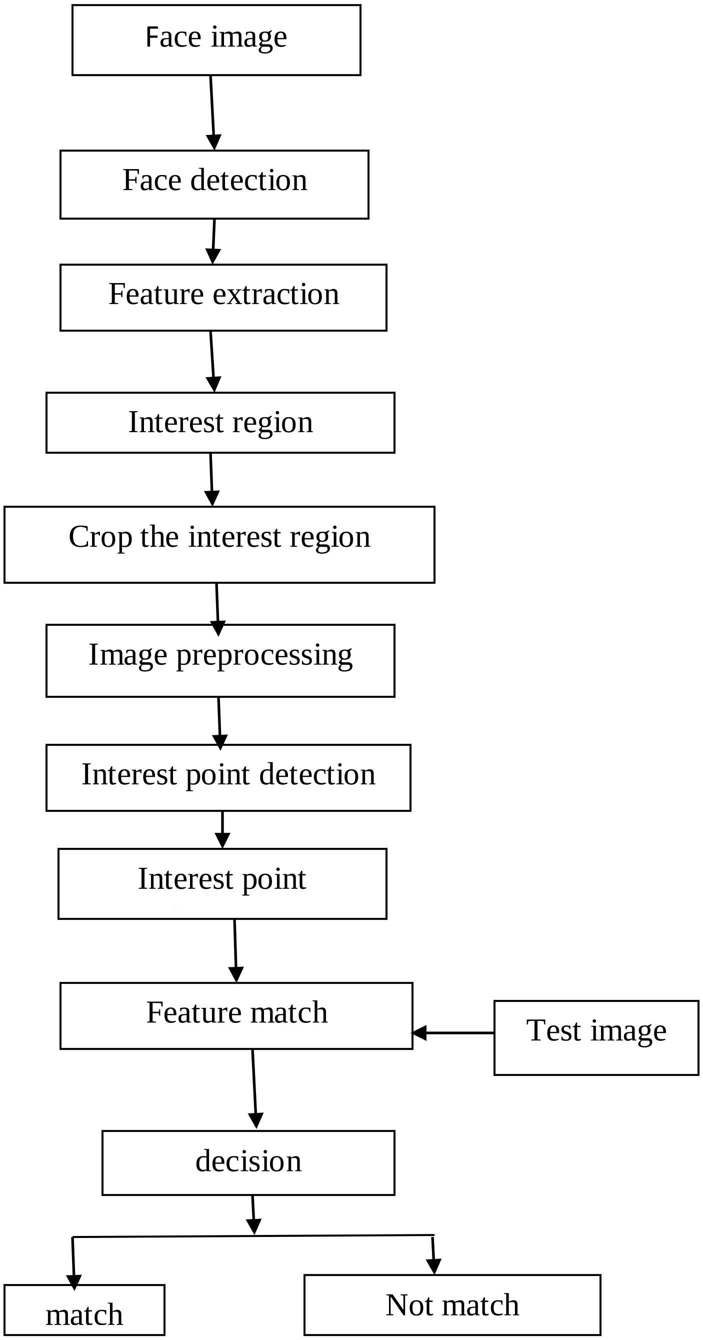


Fig 2: flow diagram of our application

4.1 Modules Description:

4.1.1 photo privacy:

Users care about privacy are unlikely to put photos online. Perhaps it is exactly those people who really want to have a photo privacy protection scheme. To break this dilemma, we propose a privacy-preserving distributed collaborative training system as our FR engine. In our system, we ask each of our users to establish a private photo set of their own. We use these private photos to build personal FR engines based on the specific social context and promise that during FR training, only the discriminating rules are revealed but nothing else With the training data (private photo sets) distributed among users, this problem could be formulated as a typical secure multi-party computation problem. Intuitively, we may apply cryptographic technique to protect the private photos, but the computational and communication cost may pose a serious problem for a large OSN.

4.1.2 Social network:

study the statistics of photo sharing on social networks and propose a three realms model: “a social realm, in which identities are entities, and friendship a relation; second, a visual sensory realm, of which faces are entities, and co-occurrence in images a relation; and third, a physical realm, in which bodies belong, with physical proximity being a relation.” They show that any two realms are highly correlated. Given information in one realm, we can give a good estimation of the relationship of the other realm. Stone et al., for the first time, propose to use the contextual information in the social realm and co photo relationship to do automatic FR. They define a pair wise conditional random field (CRF) model to find the optimal joint labeling by maximizing the conditional density. Specifically, they use the existing labeled photos as the training samples and combine the photo co occurrence statistics and baseline FR score to improve the accuracy of face annotation.   discuss the difference between the traditional FR system and the FR system that is designed specifically for OSNs. They point out that a customized FR system for each user is expected to be much more accurate in his/her own photo collections. social networks such as Face book. Unfortunately, careless photo posting may reveal privacy of individuals in a posted photo. To curb the privacy leakage, we proposed to enable individuals potentially in a photo to give the permissions before posting a co-photo. We designed a privacy-preserving FR system to identify individuals in a co-photo.

4.1.3 Friend list:

Basically, in our proposed one-against-one strategy a user needs to establish classifiers between self, friend and friend, friend also known as the two loops in Algorithm. 2. During the first loop, there is no privacy concerns of Alice's friend list because friendship graph is undirected. However, in the second loop, Alice need to coordinate all her friends to build classifiers between them. According to our protocol, her friends only communicate with her and they have no idea of what they are computing for. Friend list could also be revealed during the classifier reuse stage. For example, suppose Alice want to find ubt between Bob and Tom, which has already been computed by Bob. Alice will first query user k to see if ukj has already been computed. If this query is made in plaintext, Bob immediately knows Alice and Bob are friends. To address this problem, Alice will first make a list for desired classifiers use private set operations in [10] to query against her neighbors' classifiers lists one by one. Classifiers in the intersection part will be reused. Notice that even with this protection, mutual friends between Alice and Bob are still revealed to Bob, this is the trade-off we made for classifiers reuse. Actually, OSNs like Face book shows mutual friends anyway and there is no such privacy setting as “hide mutual friends”

4.1.4 Deep Learning:

Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. These methods have dramatically improved the state-of-the-art in speech recognition, visual object recognition, object detection and many other domains such as drug discovery and genomics. Deep learning discovers intricate structure in large data sets by using the backpropagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation in the previous layer. Deep convolutional nets have brought about breakthroughs in processing images, video, speech and audio, whereas recurrent nets have shone light on sequential data such as text and speech.

4.2 REQUIREMENT ANALYSIS

Requirements analysis understands the users requirements within the framework of the organization objective and the environment in which the system being installed. Requirements analysis is a software engineering task that bridges the gap between system level software allocation and software design. The first step in designing a system was to identify the underlying cause behind the problems in the existing system. Requirement study has been done to the gather the required information. Requirements analysis enables the system engineer to specify software function and performance, indicate software’s interface with other system elements, and establish constraints that software must meet. Requirements analysis allows the software engineer to refine the software allocation and build models of the data, Requirements analysis provides the software designer with models that can be translated into data, architectural, interface, and procedural design. Finally, the requirement specification provides the developer and the customer with the means to assess quality once software is built and behavioral domains that will be treated by software.

4.3 SOFTWARE LIFECYCLE

Requirement analysis enables the system engineer to specify the function and performance, indicate software's interface with other system elements, and establish constraints that software must meet. Requirement analysis allows the software engineer to refine the software allocation and build models of the data, requirements analysis provides the software designer with models that can be translated into data, architectural, interface, and procedural design. Finally, the requirement specification provide developer and the customer with the means to assess quality once software is built.

Software requirements analysis may be divided into 4 area of effort:

1. Problem recognition
2. Evaluation and synthesis
3. Models of the software
4. Review

Initially, the analyst studies the system specification and the software project plan. It is important to understand software in a system context and to review the software scope that was used to generate planning estimates.

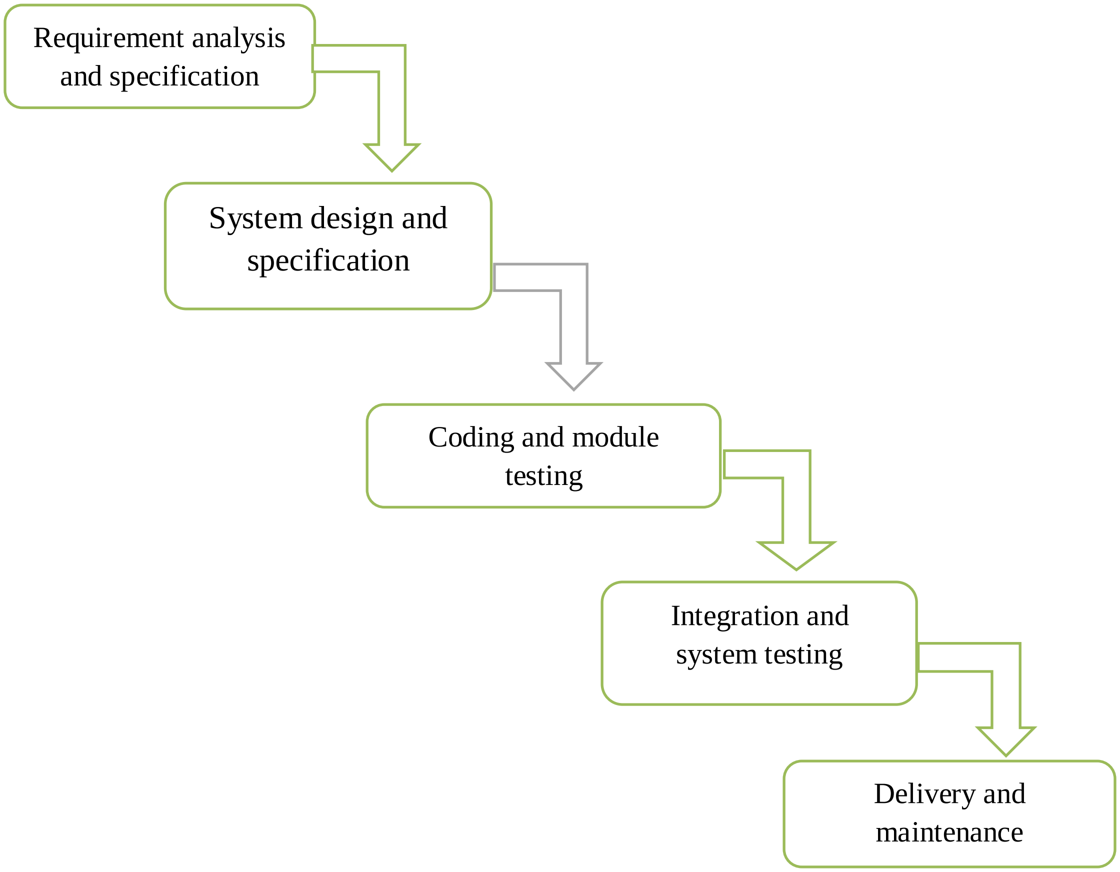

Requirement analysis and specification
System design and specification
Coding and module testing
Integration and system testing
Delivery and maintenance

Fig 3: Software lifecycle

Next communication for analysis must be established so that the problem recognition of the basic problem elements as perceived by the user/customer. Problem evaluation and solution synthesis is the major area of effort for analysis. The analyst defines all externally observable data objects; evaluate the flow and content information; define and elaborate all software functions; understand software behavior in the context of the events that affect the system; establish system interface characteristics, and uncover additional design constraints. Each of these serves to describe the problem so that an overall approach or solution may be synthesized. During the evolution and solution synthesis activity, the analyst creates models of the system in an effort to better understand data and control flow, functional processing and behavioral operations, and information content. The model serves as a foundation for software design and as the basis for the creation of a specification for the software.

4.4 Feasibility study

A feasibility study is a test of system proposal according to its workability, impact on the organization, ability to meet user needs and effective use of resources. The objective of feasibility study is not to solve problem, but to acquire a sense of its scope. During the study, the problem definition is crystallized and aspects of problem to be included in the system are determined, consequently costs and benefits are estimated with greater detail at this stage. The result of feasibility study is a system formal proposal. This is simply a form of documenting or detailing the nature and scope of proposed solution. The proposed summarizes what is known and what is going to be done. Following are the key considerations involved in the feasibility analysis

- **Pre-Feasibility Study**

A pre-feasibility study may be conducted first to help sort out relevant scenarios. Before proceeding with a full-blown feasibility study, you may want to do some pre-feasibility analysis of your own. If you find out early-on that the proposed business idea is not feasible, it will save your time and money. If the findings lead you to proceed with the feasibility study, your work may have resolved some basic issues. A consultant may help you with the pre-feasibility study, but you should be involved. This is an opportunity for you to understand issues of business development.

- **Market Assessment**

A market assessment may be conducted that will help to determine the viability of a proposed product in the marketplace. The market assessment will help to identify opportunities in market or market segment. If no opportunities are found, there may be no reason to proceed with a feasibility study. If opportunities are found, the market assessment can give focus and direction to the construction of business scenario to investigate in the feasibility study. A market assessment will provide much of the information for the marketing feasibility section of the feasibility study.

- **Result and Conclusion**

The conclusions of the feasibility study should outline in depth the various scenarios examined and the implications, strengths and weaknesses of each. The project need to study the feasibility study and challenges its underlying assumptions. This is the time to be skeptical. Don't expect one alternative to "jump off the page" as being the best scenario. Feasibility studies do not suddenly become positive or negative. As you accumulate information and investigate alternatives, neither a positive nor negative outcome may emerge. The decision of whether to proceed is often not clear-cut. Major stumbling blocks may emerge that negate the project. Sometimes these weaknesses can be overcome. Rarely does the analysis come out overwhelmingly positive. The study will help you assess the tradeoff between the risks and rewards of moving forward with the business project. Remember, it is not the purpose of the feasibility study or the role of the consultant to decide whether or not to proceed with a business idea. It is the role of the project leader to make this decision, using information from the feasibility study and input from the consultant.

➢ Go/No-Go Decision

The go/no-go decision is one of the most critical thing in business development. It is the point of no return. Once you have definitely decide to pursue a business scenario, there is usually no turning back. The feasibility study will be a major information source in making this decision. This indicates the importance of a properly developed feasibility study.

➢ Reason Given Not to Do a Feasibility Study

- Manpower–programmers, testers we know it’s feasible. An existing business is already doing it.
- No need of another feasibility study when one was just a few years ago.
- Feasibility studies are just a way for consultants to make money.
- The market analysis has already been done by the business that is going to sell us the equipment.
- Hire a general manager who can do the study.
- Feasibility studies are a waste of time. We need to buy the building, tie the site and bid on the equipment.

The reason given above should not dissuade you from conducting a meaningful and accurate feasibility study. Once decision have been made about proceeding with a proposed business, they often very difficult to change. You may need to live with these decisions for a long time.

- **Reason to do a Feasibility Study**

Conducting a feasibility study is a good business practice. If you examine successful businesses, you will find that they did not go into a new business venture without first thoroughly examining all of the issues and assessing the probability of business success. Below are other reasons to conduct a feasibility study.

- Give focus to the project and outline alternatives.
- Narrow business alternatives.
- Identifies reasons not to proceed.
- Enhance the probability of success by addressing and mitigating factors early on that could affect the project.
- Provide quality information for decision making
- Provides documentations that the business venture was thoroughly investigated.
- Helps in secure funding from leading institutions and other monetary sources.
- Help to attract equity investment.

The feasibility study is critical step in the business assessment process. If properly conducted, may be the best investment you ever made.

4.4.1 ECONOMIC FEASIBILITY

The economic analysis is to determine the benefits and savings that are expected from candidates system and compare them with the cost. The system is economically feasible, as the organization possesses the hardware and software resources required for the functioning of the system. Any additional resources, if required, can also be easily acquired. Feasibility studies are crucial during the early development of any project and from a vital component in the business development process. Accounting and advisory feasibility studies enable organizations to assess the viability, cost and benefits before of project financial resources are allocated. They also provide independent project assessment and enhance project credibility.

Built on the information provided in the feasibility study, a business case is used to convince the audience that a particular project should be implemented. It is often a prerequisite for any funding approval. The business case will detail the reason why a particular project should be prioritized higher than others. It will also sum up the strengths, weaknesses and validity of assumption as well as assessing the financial and non-financial costs and benefits underlying preferred options. A Deloitte feasibility study can help your organization to:

- Define the business requirements that must be met by the selected project and include the critical success factor for the project.
- Detail alternative approaches that will meet business requirement, including comparative cost/benefit and risk analyses.
- Recommend best approach for preparing a business case or moving through the implementation process.

Feasibility studies and business cases can help to know following:

- Whether the alternatives been carefully, thoroughly and objectively examined.
- The consequence of each choice on all relevant areas.
- The results of any cost/benefit studies.
- The costs and consequences of no action.
- What are the timelines for decision?
- Consequences displayed to make comparisons easier.

4.4.2 TECHNICAL FEASIBILITY

It centers on the existing computer system and to what extent it can support the proposed addition. Since the minimum requirements of the system like IIS o the server and a browser on the client, are met by any average user. A large part of the determining resources has to do with assessing technical feasibility. It considers the technical requirements of the proposed project. The technical requirements are then compared to technical capability of the organizations. The system project is considered technically feasible if the internal technical capability is sufficient to support the project requirements. The analyst must find out whether current technical resources can be upgraded or added to in a manner that fulfils the request under consideration. This is where the expertise of system analysts is beneficial, since using their own experience and their contact with vendors they will be able to answer the question of technical feasibility. The essential ideas that help in testing the operational feasibility of a system include the following:

- Whether the project is feasible within the limits of current technology.
- Whether the technology exist at all.
- Availability within given resource constraints.
- Whether the practical proposition.
- Debuggers.
- Software and hardware.
- Whether the current technical resources sufficient for the new system
- Whether they be upgraded to provide the level of technology necessary for the new system.
- Possessing the necessary technical expertise, and is the schedule reasonable.
- Technology be easily applied to current problems

- Technology have the capacity to handle the solution
- Currently possess the necessary technology

4.4.3 OPERATIONAL FEASIBILITY

The system operation is the longest phase in development life cycle of a system.so, operational feasibility should be given much importance. The users of the system don’t need thorough training on the system. All they are expected to know to operate the system is the basic net surfing knowledge. It has a user-friendly interface.

Operational feasibility is dependent on human resource available for the project and involves projecting whether the system will be used if it is developed and implemented. Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantages of the opportunities identified during scope definition and how it satisfies the requirements identified the requirements analysis phase of development. Operational feasibility reviews the willingness of the organization to support the proposed system. This is probably the most difficult of the feasibility to gauge. In order to determine the feasibility, it is important to understand the management, it is likely that there is management support and the system will be accepted and used. However, it is also important that employee base will be accepting of the change. The essential idea that help in testing the operational feasibility of a system include the following:

- Current mode of operation provides adequate throughput and response time.
- Current mode provides end users and managers with timely, pertinent, accurate and useful formatted information.
- Current mode of operation provides cost-effective information services to the business.

- Reduction in cost and or increase in benefits
- Current mode of operation offer effective controls to protect against fraud and to guarantee accuracy and security of data and information.
- Current mode of information makes maximum use of available resources, including people, time and flow of form.
- Current mode of operation provides reliable services.
- Services flexible and expandable.
- Current work practices and procedures adequate to support the new system.
- Will the system developed be used?
- Manpower problems.
- Labour objects.
- Manager resistance.
- Organizational conflicts and policies.
- Social acceptability.
- Government regulations.
- Management support for the project.
- Happiness of users with current business practices.
- Considerable reduction in operational time.
- Involvement of users in the planning and development of the project.
- Benefits of the proposed system to the organization.
- Overall response increase.
- Accessibility of information being lost.
- System effect on the customers in considerable way.

- How do the end-users feel about their role in the new system?
- End user or managers resist or not use the system.
- Change in working environment of the end-user.
- Adaption of end-users and management to the change.

4.4.4 BEHAVIOURAL FEASIBILITY

In today’s world, where computer is an inevitable entry, system like auction site, which requires no special effort than surfing the net are enjoying wide acceptance. Thus organization is convinced that the system is feasible. People are inherently resistant to change, and computers have been known to facilitate change. An estimate should be made of how strong a reaction the user staff is likely to have toward the development of a computerized system. It is common knowledge that the computer installation have something to do with turnover, transfer, retaining and change in employee job status. Therefore, it is understandable that the introduction of a candidate system requires special effort to educate, sell, and train the staff on new ways of conducting business.

- An estimate should be made of how strong a reaction the user staff is likely to have towards the development of a computerized system.
- It is common knowledge that computer installation have something to do with turnover, transfer and change in employee job status.

Normal human psychology of human beings indicate that people are resistant to change and computer are known facilitate change. Any project formulations should consider the factor also. Before the development of the project titled “Delhi metro”, the need to study the feasibility of successful execution of the project was felt and thus the following factors are considered for feasibility study.

1. Need Analysis
2. Provide the users information pertaining to the preceding requirement

4.5 SYSTEM DESIGN

The most creative and challenging phase of the system development is thesystem design. It provides the understanding and procedural details necessaryfor implementing the system recommended in the feasibility study. Design goesthrough the logical and physical stages of development.In designing a new system, the analyst must have a clear understanding ofthe objectives, which the design is aiming to fulfill. The first step is to determinehow the output is to be produced and in what format. Second input dataandmaster files have to be designed to meet the requirements of the proposed output.The operational phases are handled through program construction and testing.Finally details related to justification of the system and an estimate of the impactof the candidate system on the user and the organization are documented andevaluated by the management.

Design of a system can be defined as a process of applying varioustechniques and principles for the purpose of defining a device, a process or asystem in sufficient detail to permit its physical realization. Thus system design is a solution, “how to” approach to the creation of a new system. The design stepprovides a data design, architectural design and a procedural design.The data design transforms the information domain created during analysisinto the data structure that will be required to implement the software. Thearchitectural design defines the relationship among major structural componentsand procedural description of the software. Source code is generated and testingconducted to integrate and validate the software. System design goes through two phases of development:

- Logical design
- Physical design

Logical design

The part of the design process that is independent of any specifichardware or software platform is referred to as logical design. During logicaldesign, all functional features of the system chosen for development in analysisphase are described independently of any computer platform. Logical designconcentrates on the business aspects of the system and tends to be oriented to ahigh level of specificity.

Physical design

Physical design is the part of the design phase in which the logicalspecifications of the system from logical design are transferred into technology-specific details from which all programming and system construction can beaccomplished. As a part of the physical design, analysts design the various partsof the system to perform the physical operation necessary to facilitate datacapture, processing, and information output.

4.5.1. INPUT DESIGN

The first step in system design is to design input and output withinpredefined guidelines. In input design, user originated inputs are convertedinto computer based format. In output design, the emphasis is on producingthe hard copy of the information requested or displaying the output on a CRTscreen in a predefined format. The following features have been incorporatedinto the input design of the proposed system.

Easy Data Input

Data entry has beendesigned in a manner much similar to the sourcedocuments. Appropriate messages are provided in the message area, whichprompts the user in entering the right data. Erroneous data inputs arechecked at the end of each screen entry.

Data Validation

The input data is validated to minimize errors in data entry. For certaindata specific codes have been given and validation is done which enables theuser to enter the required data or correct them if they entered wrong codes.

User Friendliness

User is never left in a state of confusion as to what is happening, insteadappropriate error and acknowledge messages are sent. Error maps are used toindicate the error codes and specific error messages.

Consistent Format

A fixed format is adopted for displaying the title messages. Every screen has line, which displays the operation that can be performed after the data entry. They are normally done at the touch of a key.

Interactive Dialogue

The system engages the user in an interactive dialogue. The system is able to extract missing or omitted information from the user by directing the user through appropriate messages, which are displayed.

4.5.2 OUTPUT DESIGN

The output is the most important and direct source of information into the user. The output should be provided in a most efficient formatted way. Based on the options given by the users and the administrator various types of output screens have been generated. The computer output is the most important and direct source of information to the user. Efficient and intelligible output design improves the system's relationship with the user and helps in decision-making. Output desighnwas studied going actively during the study phase. The objective of the outputdesign is defined the contents and format of all documents and reports in anattractive and useful format

4.6 MODULE DESCRIPTION

The application contain only one part

- Android app

Android part feature

1. Users can signup or login to the application
2. post photos and co-photos
3. Provide notification when someone post users photo

4. view post of others
5. Send request to other users
6. View request
7. Like and comment post
8. Chat with friends

4.7 DATAFLOW DIAGRAM

A data flow diagram (DFD) is a graphical representation of the flow of data through an information system, modeling its process aspects. A DFD shows what kinds of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. These are expanded by level, each explaining its process in detail. Processes are numbered for easy identification and are normally labeled in block letters. Each Data flow is labeled for easy understanding.

Data flow diagrams are made up of a number of symbols, which representssystem components. Data flow modeling method uses four kinds of symbols.

Process

Process shows the work of the system. Each process has one or more datainputs and produce one or more data outputs. Processes are represented bycircles in data flow diagrams.

Data Stores

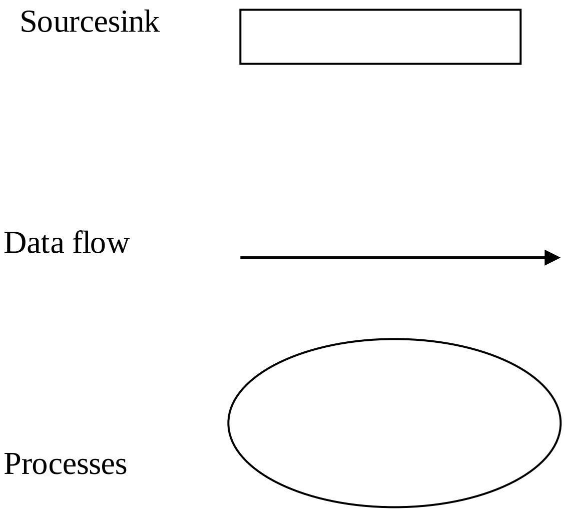
A data store is a repository of data. Processes can enter data into a store orretrieve data from the data store. Data stores are represented by two parallellines, which may be depicted horizontally or vertically.

Data Flows

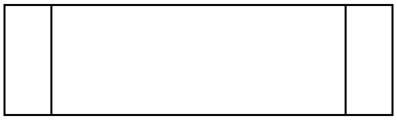
The arrows represent data flow. A data flow is data in motion. A data flowrepresents an input of data to a process or the output of the data from a process.A data flow is also used to represent the creation, reading, deletion, or updatingof data in a file or database.

External Entities

External entities are outside the system but they either supply input thesystem or use other systems output. They are entities on which the designer hascontrol. External entities that supply data into the system are sometimes called*source*. External entities that use the system data are called*sinks*.Thesearerepresented by rectangles in the data flow diagram.



Data store



LEVEL 0:

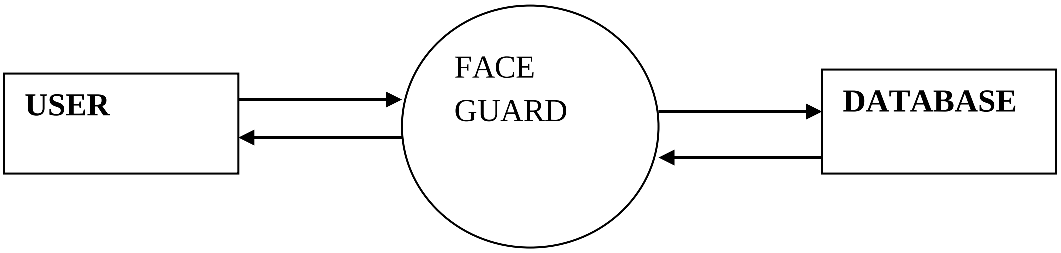


Fig 4: level 0 for user

LEVEL 1

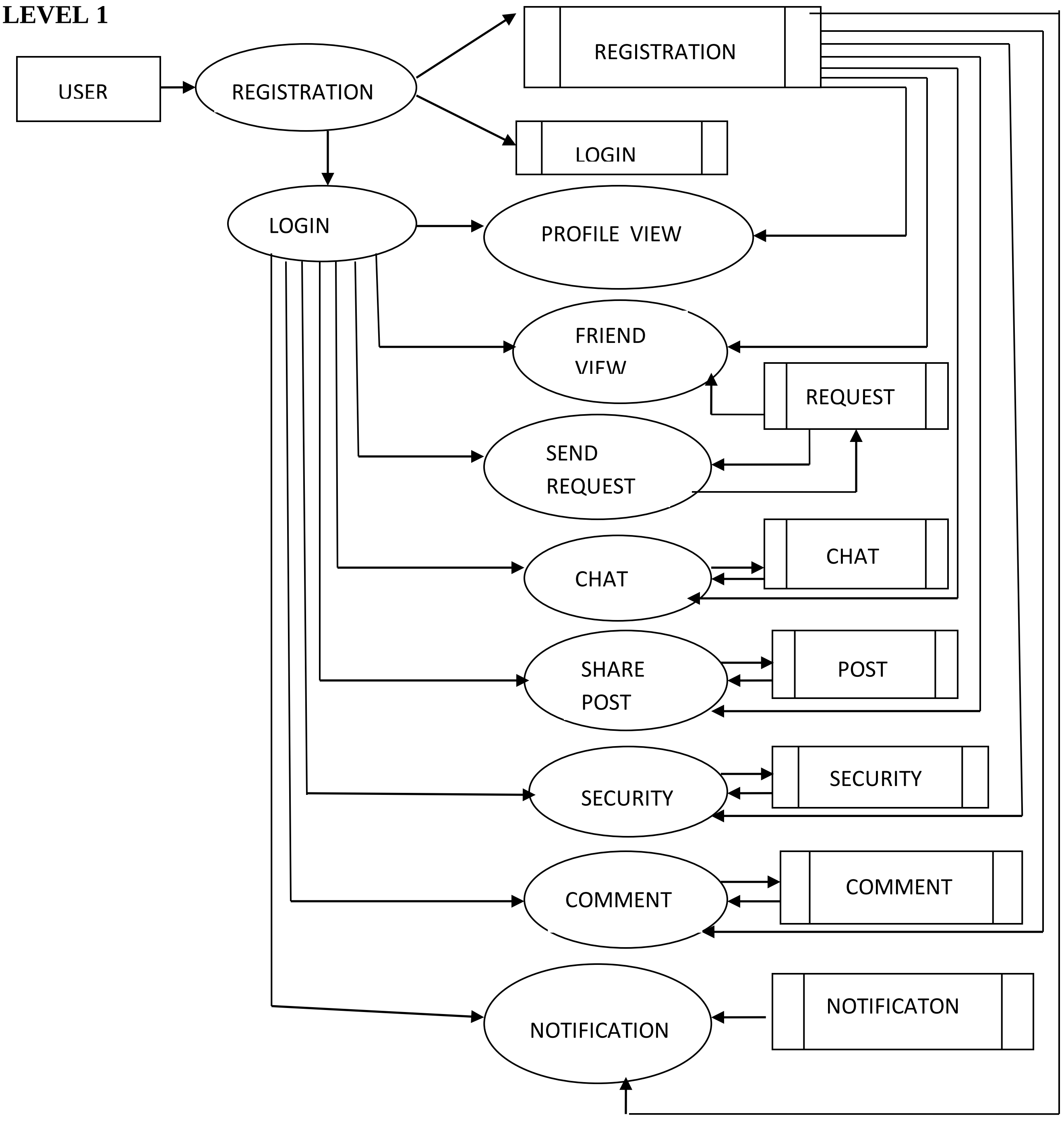


Fig 5: level 1 for user

4.8 DATA BASE DESIGN

A data base is a collection of inter related data stored with minimum redundancy to serve many quickly and efficiently. The general way is to make information accessing easy, quick, inexpensive and flexible for the user. In data base design several objectives are considered controlling redundancy, ease of learning and use, data dependence, more information at low cost, accuracy and integrity are some of them.

CHAPTER 5
SYSTEM REQUIREMENTS

The system specification refers to a detailed functional and non-functional description of a system. This term can also be defined as an explicit set of requirement that need to be satisfied be specific system. System specification includes software and hardware specification of project.

5.1 HARDWARE SPECIFICATION

The selection of hardware is very important in the existence and proper working of any software. Then selection hardware, the size and capacity requirements are also important.

• Processor	:	Intel Pentium Core i3 and above
• Primary Memory	:	1 GB RAM and above
• Storage	:	40 GB hard disk and above
• Display	:	VGA Color Monitor
• Key Board	:	Windows compatible
• Mouse	:	Windows compatible

5.2. SOFTWARE SPECIFICATION

One of the most difficult task is selecting software for the system, once the system requirements is found out then we have to determine whether a particular software package fits for those system requirements. The application requirement:

Front end	:	C#
Back end	:	SQL server 2008
Operating system	:	windows 7 and above
IDE	:	Visual studio 2010

TECHNOLOGIES

Coding	:	C#
Design	:	CSS
Connection	:	ADO .Net

ANDROID

HARDWARE REQUIREMENTS

A mobile phone with **Android** operating system

Version: Android 2.1 or above

SOFTWARE REQUIREMENTS:

Platform	- ANDROID
Front End	- Java (JDK 6), XML (Android Development Tool)
IDE	- Eclipse Indigo

Software used for development – Android Development Kit (Plug-in to the Eclipse IDE)

MINIMUM HARDWARE REQUIREMENTS

❖ System	:	Pentium IV 2.4 GHz.
❖ Hard Disk	:	40 GB.
❖ Floppy Drive	:	1.44 Mb.
❖ Monitor	:	14’ Colour Monitor.
❖ Mouse	:	Optical Mouse.
❖ Ram	:	512 Mb.

FRONT END

4.1.1 CODING STANDARDS

To build python web applications and android application, which are reliable, scalable and maintainable, it is important for development teams to adopt proven design techniques and good coding standards. The adoption of coding standards results in code consistency, which

Pycharm (Python Interpreter)

Python is an interpreter, object-oriented, high-level programming language with dynamic semantics. 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. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse.

The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by metaprogramming and metaobjects (magic methods)). Many other paradigms are supported via extensions, including design by contract and logic programming. Python uses dynamic typing, and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution.**Android**

Android is an operating system based on the Linux kernel. The project responsible for developing the Android system is called the *Android Open Source Project* (AOSP) and is primarily lead by Google. The Android system supports background processing, provides a rich user interface library, supports 2-D and 3-D graphics using the OpenGL-ES (short OpenGL) standard and grants access to the file system as well as an embedded SQLite database. An Android application typically consists of different visual and non visual components and can reuse components of other applications. The Android system is a full software stack, which is typically divided into the four areas

The levels can be described as:

- Applications - The Android Open Source Project contains several default applications, like the Browser, Camera, Gallery, Music, Phone and more.
- Application framework - An API which allows high-level interactions with the Android system from Android applications.
- Libraries and runtime - The libraries for many common functions (e.g.: graphic rendering, data storage, web browsing, etc.) of the Application Framework and the Dalvik runtime, as well as the core Java libraries for running Android applications.
- Linux kernel - Communication layer for the underlying hardware.

The Linux kernel, the libraries and the runtime are encapsulated by the application framework. The Android application developer typically works with the two layers on top to create new Android applications.The *Android Software Development Kit* (Android SDK) contains the necessary tools to create, compile and package Android applications. Most of these tools are command line based. The primary way to develop Android applications is based on the Java programming language. The Android SDK contains the *Android debug bridge* (adb), which is a tool that allows you to connect to a virtual or real android device, for the purpose of managing the device or debugging your application. Google provides two integrated development environments (IDEs) to develop new applications. The *Android DeveloperTools* (ADT) are based on the Eclipse IDE. ADT is a set of components (plug-ins), which extend the Eclipse IDE with Android development capabilities. Google also supports an IDE called *Android Studio* for creating Android applications. This IDE is based on the IntelliJ IDE. Both IDEs contain all required functionality to create, compile, debug and deploy Android applications. They also allow the developer to create and start virtual Android devices for testing. Both tools provide specialized editors for Android specific files. Most of Android's configuration files are based on XML. In this case these editors allow you to switch between the XML representation of the file and a structured user interface for entering the data. Eclipse uses plug-ins to provide all the functionality within and on top of the runtime system. Its runtime system is based on Equinox, an implementation of the OSGi core framework specification.

In addition to allowing the android Platform to be extended using other programming languages such as C and Python, the plug-in framework allows the Eclipse Platform to work with typesetting languages like LaTeX networking applications such as telnet and database management systems. The plug-in architecture supports writing any desired extension to the environment, such as for configuration management. Java and CVS support is provided in the Eclipse SDK, with support for other version control systems provided by third-party plug-ins. The Eclipse SDK includes the Eclipse Java development tools (JDT), offering an IDE with a built-in incremental Java compiler and a full model of the Java source files. This allows for advanced refactoring techniques and code analysis. The IDE also makes use of a workspace, in this case a set of metadata over a flat files pace allowing external file modifications as long as the corresponding workspace "resource" is refreshed afterwards. Eclipse implements use the graphical control elements of the Java toolkit called SWT, whereas most Java applications use the Java standard Abstract Window Toolkit (AWT) or Swing. Eclipse's user interface also uses an intermediate graphical user interface layer called JFace, which simplifies the construction of applications based on SWT.

BACK END

MySQL is an open source relational database management system. It is based on the Structured Query Language (SQL), which is used for adding, removing and modifying information in the database. Standard SQL commands, such as ADD, DROP, INSERT and UPDATE can be used with MySQLMicrosoft SQL Server is a relational database management system developed by Microsoft. As a database, it is a software product whose primary function is to store and retrieve data as requested by other software applications, be it those on the same computer or those running on another computer across a network (including the Internet). There are at least a dozen different editions of Microsoft SQL Server aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users. Its primary query languages are T-SQL and ANSI SQL.

Microsoft SQL Server is a widely accepted standard database sublanguage used in querying updating and managing relational databases. SQL Server 2000 has features that will import, transform and export data from multiple sources, from Oracle to text files. It can also act as the backbone to your Business Intelligence infrastructure, as an XML processing tool, or even to process natural language queries. It has the ability to be "clustered" so that it can automatically fail over to another system in case of a catastrophe.

It can also serve as a reporting server front-end for your users. The Code Editor component of SQL Server Management Studio contains integrated script editors for authoring Transact-SQL, MDX, DMX, and XML/A. The Object Explorer component of SQL Server Management Studio is an integrated tool for viewing and managing objects in all server typesSql Server manager includes features such as Supports most administrative tasks for SQL Server,A single, integrated environment for SQL Server Database Engine management and authoring,Dialogs for managing objects in the SQL Server Database Engine, Analysis Services, and Reporting Services, that allows you to execute your actions immediately, send them to a Code Editor, or script them for later execution,Non-modal and resizable dialogs allow access to multiple tools while a dialog is open,A common scheduling dialog that allows you to perform action of the management dialogs at a later time.

It includes the following features:

- Internet Integration
- Scalability and availability
- Enterprise-level database features
- Ease of installation , deployment and use of warehousing
- Query Analyzer and Sql profiler
- Data definition and data manipulation languages and database keys

4.1.3 Hardware Requirements:

- Processor : Intel core i3 or above
- Hard Disk : 80 GB or above
- RAM : 4 GB or above
- Monitor : 15 CRT, or LCD monitor
- Keyboard : Any Compatible
- Mouse : Any Compatible
- Compatible : Any Intranet or Internet

4.1.4 Software Requirements:

- Operating system : Windows 10 Or above
- Front End : Python 3.4(Web Application)

- Java (Android Application)
- Back End Software : MySQL Database (sql log 8.03)
- IDE :LiClipse

- Android Studio 2.3.3
- Framework : Flask
- Web Server : Internet Explorer, Mozilla Fire fox, Google chrome

5.3 SYSTEM TESTING

Testing is an activity to verify that a correct system is being built and isperformed with the intent of finding faults in the system. However not restricted tobeing performed after the development phase is complete, but this is to carry out inparallel with all stages of system development, starting with requirementsspecification. Testing results, once gathered and evaluated, provide aqualitativeindication of software quality and reliability and serve as a basis for designmodification if required. A project is said to be incomplete without proper testing.

System testing is a process of checking whether the developed system is workingaccording to the original objectives and requirements. The system should be testedexperimentally with test data so as to ensure that system works according to therequired specification. When the system is found working, test it with actual dataand check performance.The testingprocedure that has been used as follows:

- Unit Testing
- Integration Testing
- Validation Testing
- Output Testing
- User Acceptance Testing

Unit Testing

The first level of testing is called as unit testing. Here the differentmodules are tested and the specification produced during design for the modules.Unit testing is essential for verification of the goal and to test the internal logic ofthe modules. Unit testing is conducted to different modules of the project. Errorswerenoted down and corrected down immediately and the program clarity wasincreased. The testing was carried out during the programming stage itself. Inthis step each module is found to be working satisfactory as regard to be expectedout from the module.

Integration Testing

The second level of testing includes integration testing. It is a systematictesting of constructing structure. At the same time tests are conducted touncover errors with the interface. It need not to be the case, that softwarewhose modules when run individually showing results will also show perfectresults when run as a whole.The individual modules are tested again and the results are verified. Thegoal is to see if the modules integrated between the modules. This testing activitycan be considered as testing the design and emphasizes on testing modulesinteraction.

Validation Testing

The next level of testing is validation testing. Here the entire software istested. The reference document for this process is the requirement and the goal isto see if the software meets its requirements.The requirement document reflects and determines whether the softwarefunctions as the user expected. At culmination of integration testing, software iscompletely assembled as a package and corrected and a final series of softwaretest validation test begins. The proposed system under construction has beentested by using validation testing and found to be working satisfactory.Data validation checking is done to see whether the corresponding entriesmade in different tables are done correctly. Proper validation checks are done incase of insertion and updating of tables, in order to see that no duplication ofdata has occurred. If any such case arises proper warning message will bedisplayed. Double configuration is done before the administrator deletes a data inorder to get positive results and to see that o data have been deleted by accident.

Output Testing

The output of the software should be acceptable to the system user. Theoutput of requirement is defined during the system analysis. Testing of thesoftware system is done against the output and the output testing was completedwith success.

User Acceptance Testing

An acceptance test has the objective of selling the user on the validity andreliability of the system. It verifies that the system procedures operate to systemspecification and the integrity of the vital data is maintained.

5.4 IMPLEMENTATION

System implementation is the final phase i.e., putting the utility into action. Implementation is the state in the project where theoretical design turnedintoworkingsystem. Implementation involves the conversion of a basic application to complete replacement with a computer system. It is the process ofconverting to a new or revised system design into an operational one. During the design phase, the products structure, its undergoing data structures, the generalalgorithms and the interfaces and control/data linkages needed to supportcommunication among the various sub structures were established.

Implementation process is simply a translation of the design abstraction into the physical realization, using the language of the target architecture.Implementation includes all those activities that take place to convert from the old system to the new. The new system may be totally new replacing anexisting manual or automated system, or it may be major modification to anexisting system. In either case, proper implementation isessential to provide areliable system to meet organizational requirements.There are three types of implementation:

- Implementation of a computer system to replace a manual system.
- Implementation of a new computer system to replace an existing one.
- Implementation of a modified application to replace an existing one, usingthe same computer.

The common approaches for implementation are:

Parallel Conversion

In parallel conversion the existing system and new system operatesimultaneously until the project team is confident that the new system is workingproperly. The outputs from the old system continue to be distributed until the newsystem has proved satisfactorily parallel conversion is a costly method because of theamount of duplication involved.

Direct Conversion

Under direct conversion method the old system is discontinued altogether and thenew system becomes operational immediately. A greater risk is associated withdirect conversion is no backup in the in the case of system fails.

Pilot Conversion

A pilot conversion would involve the changing over of the part of thesystem either in parallel or directly. Use of the variation of the two main methodsis possible when part of the system can be treated as a separate entity.

User Training

After the system is implemented successfully, training of the user is one ofthe most important subtasks of the developer. For this purpose user manuals areprepared and handled over to the user to operate the developed system. Thus theusers are trained to operate the developed system. In order to put newapplication system into use, the following activities were taken care of:

- Preparation of user and system documentation.
- Conducting user training with demo and hands on.
- Test run for some period to ensure smooth switching over the system.

The major implementation procedures are:

- Test plans
- Training
- Conversion

Test Plans

The implementation of a computer based system requires that the test data be prepared and the system and its elements be tested in a structured manner.

Training

The purpose of training is to ensure that all the personel who are to beassociated with the computer based business system possesses the necessaryknowledge skills. As the system provides user friendliness only basic training isneeded.

Conversion

It is the process of performing all of the operations that results directly intheturnover of the new system to the user. Conversion has two parts:

The creation of a conversion plan at the start of the development phase andthe implementation of the plan throughout the development phase.

The creation of a system change over plan at the end of the developmentphase and the implementation of the plan at the beginning of the operationphase.

5.5 SYSTEM MAINTENANCE

The maintenance is an important activity in the life cycle of a softwareproduct. Maintenance includes all the activities after the installation of softwarethat is performed to keep the system operational. The maintenance phase of asoftware life cycle is the time period in which a product performs useful work.Maintenance is classified into four types:

- Corrective Maintenance
- Adaptive Maintenance
- Perfective Maintenance
- Preventive Maintenance

Corrective Maintenance

Corrective maintenance refers to changes made to repair defects in thedesign, coding, or implementation of the system. Corrective maintenance is oftenneeded for repairing processing or performance failures or making changesbecause of previously uncorrected problems or false assumptions. Most correctivemaintenance problems surface soon after the installation. When correctivemaintenance problems surface, they are typically urgent and need to be resolvedto curtail possible interruptions in normal business activities.

Adaptive Maintenance

Adaptive maintenance involves making changes to an information systemto evolve its functionality or to migrate it to different operating environment.Adaptive maintenance is usually less urgent than corrective maintenancebecause of business and technical changes typically occur some period of time.

Perfective Maintenance

Perfective maintenance involves making enhancements to improveprocessing performance, interface usability, or to add desired, but not necessarilyrequired, system features. Many system professionals feel that perfectivemaintenance is not really the maintenance but new development.

Preventive Maintenance

Preventive maintenance is the only maintenance activity which is carriedout without formal maintenance request from the user. When a softwarecompany or maintenance agency realizes that the methodologies used in aprogram have become obsolete, it may decide to develop or modify parts of theprogram, which do not confirm to the current trends.Of these types, more time and money is spending on perfective than on correctiveand adaptive maintenance together.

5.6 SYSTEM SECURITY

Security features are considered while developing a system, so as to avoidtheerrors and omissions that may lead to serious problems. Computer system issecure against a particular threat if counter measures have been taken to reduceacceptability low level amounts of loss that the threat may be expected to causeover a given period of time. A computer should be protected from the followingthree problems:

- Loss of availability
- Loss of integrity
- Loss of confidentiality

A threat to a computer system is any events that adversely affected theone or more assets or resources, which make up a system. An event can be any ofthe following:

- Interruption of communication
- Destruction of hardware
- Modification of programs
- Removal of programs
- Disclosure of information

There are many methods for handling a threat

- Altering the design
- Threat retention
- Threat reduction

There are many possible threats to the security and integrity of anysystem where more than one user is associated with the system. Softwareintegrity has become increasingly important. The attribute measures a system'sability to withstand attacks, both accidental and intentional on its security.Attacks can be made on all three components of software programs, data anddocuments.

CHAPTER 6

RESULT AND DISCUSSION

To prevent possible privacy leakage of a photo, we design a mechanism to enable each individual in a photo be aware of the posting activity and participate in the decision making on the photo posting. For this purpose, we need an efficient facial recognition (FR) system that can recognize everyone in the photo. Applications providing face detection usually takes a large dataset for comparison but FACEGUARD uses only a single registered photo for comparison.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

Photo sharing is one of the most popular features in online social networks such as Facebook. Unfortunately, careless photo posting may reveal privacy of individuals in a posted photo. To curb the privacy leakage, we proposed to enable individuals potentially in a photo to give the permissions before posting a co-photo. We designed a privacy-preserving FR system to identify individuals in a co-photo. The proposed system is featured with low computation cost and confidentiality of the training set. Theoretical analysis and experiments were conducted to show effectiveness and efficiency of the proposed scheme. We expect that our proposed scheme be very useful in protecting users' privacy in photo/image sharing over online social networks. However, there always exist trade-off between privacy and utility. For example, in our current Android application, the co-photo could only be post with permission of all the co-owners. Latency introduced in this process will greatly impact user experience of OSNs. More over, local FR training will drain battery quickly.

Our future work could be how to move the proposed training schemes to personal clouds like Dropbox and/or icloud.

APPENDIX A

SCREENSHOTS

REGISTRATION LOGIN

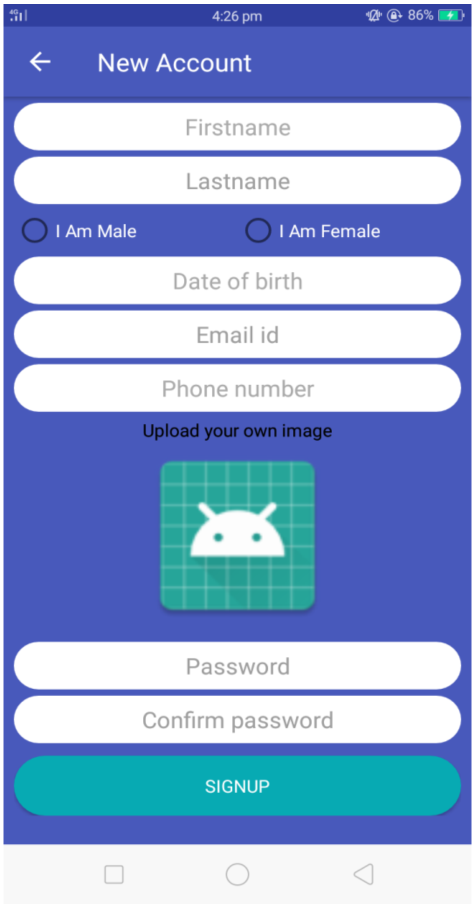
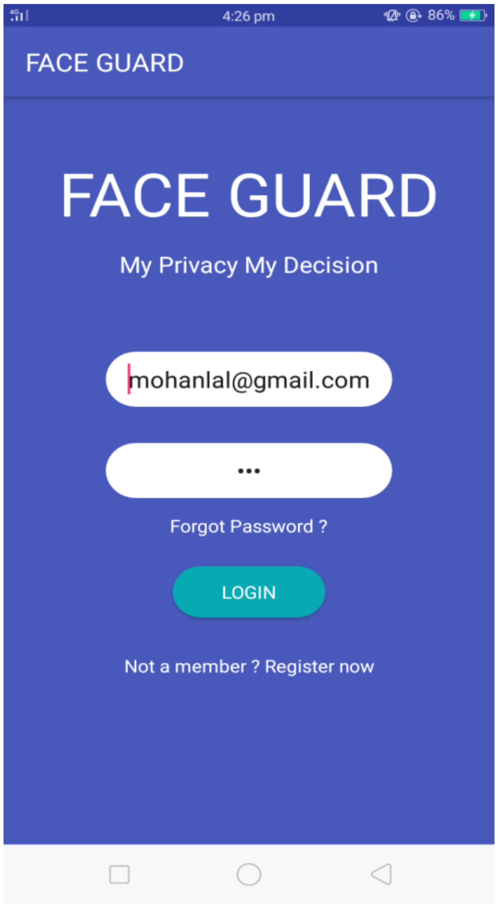


fig 6: registration page



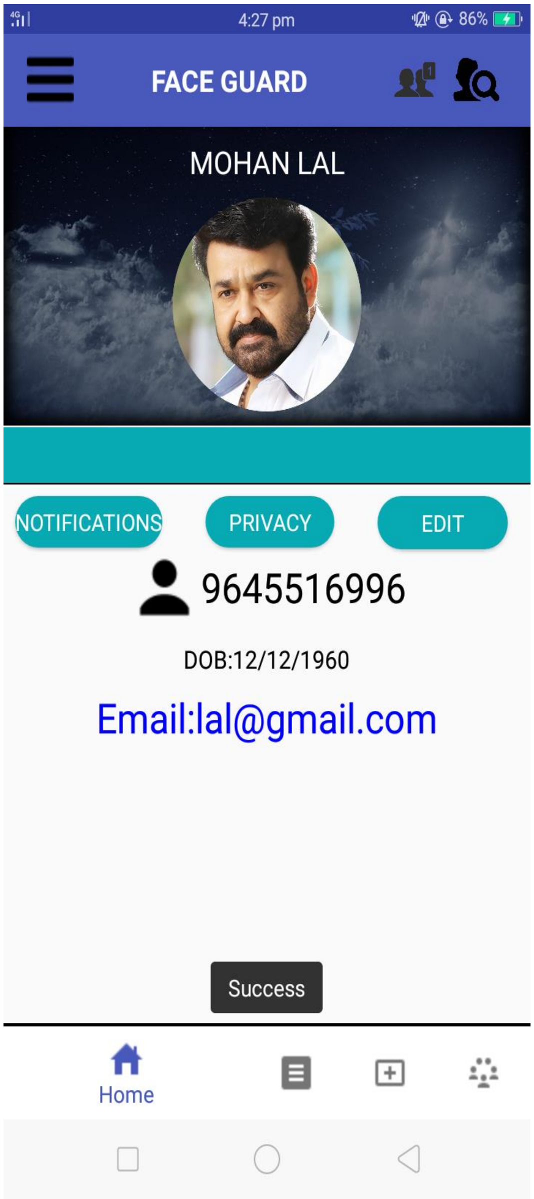
mohanlal@gmail.com

•••

fig 7: login page

HOMEPAGE

FRIEND REQUEST



MOHAN LAL

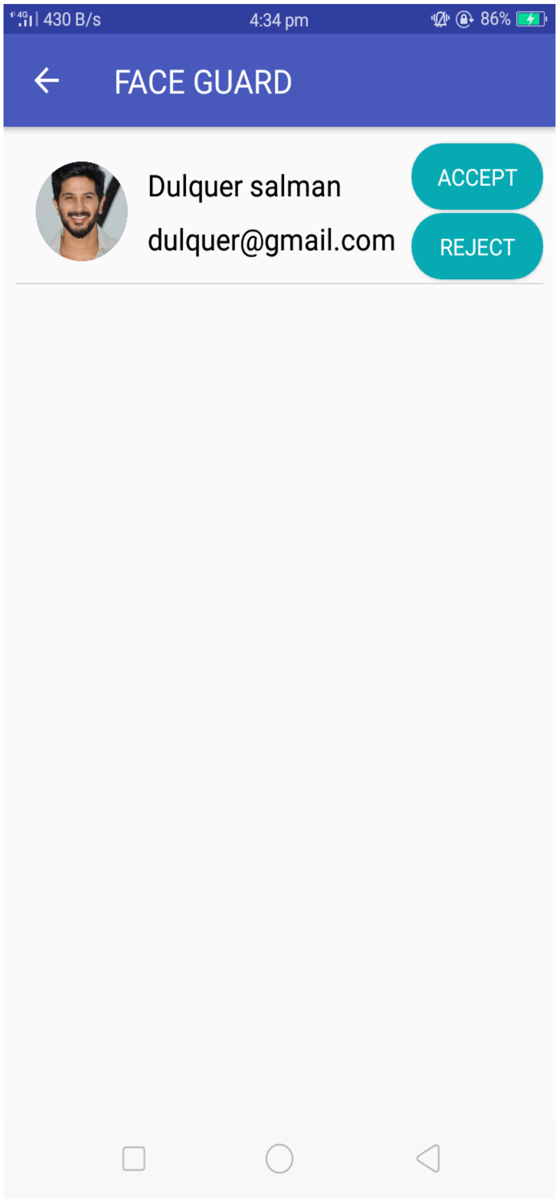
9645516996

DOB:12/12/1960

Email:lal@gmail.com

Success

Fig 8: home page



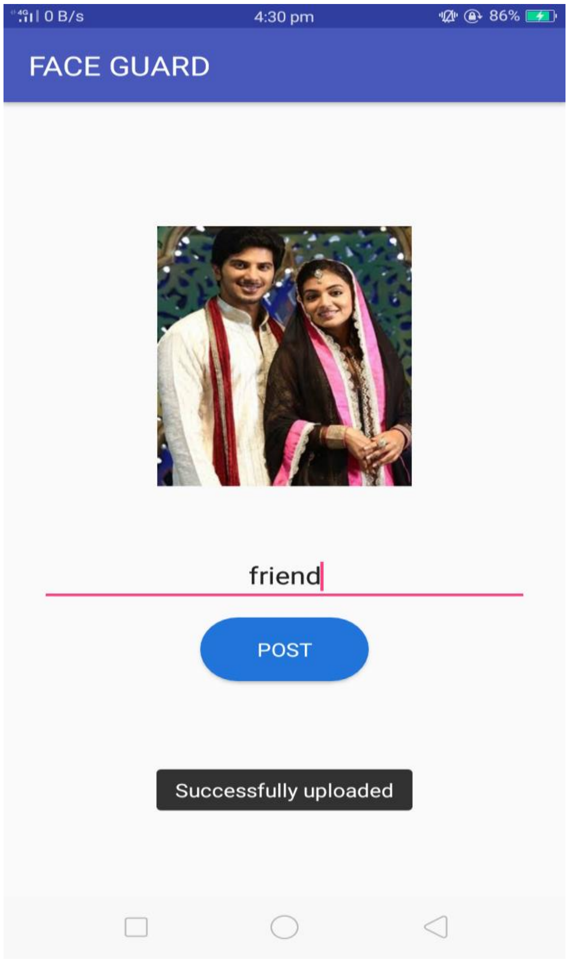
Dulquer salman

dulquer@gmail.com

fig 9: friend request page

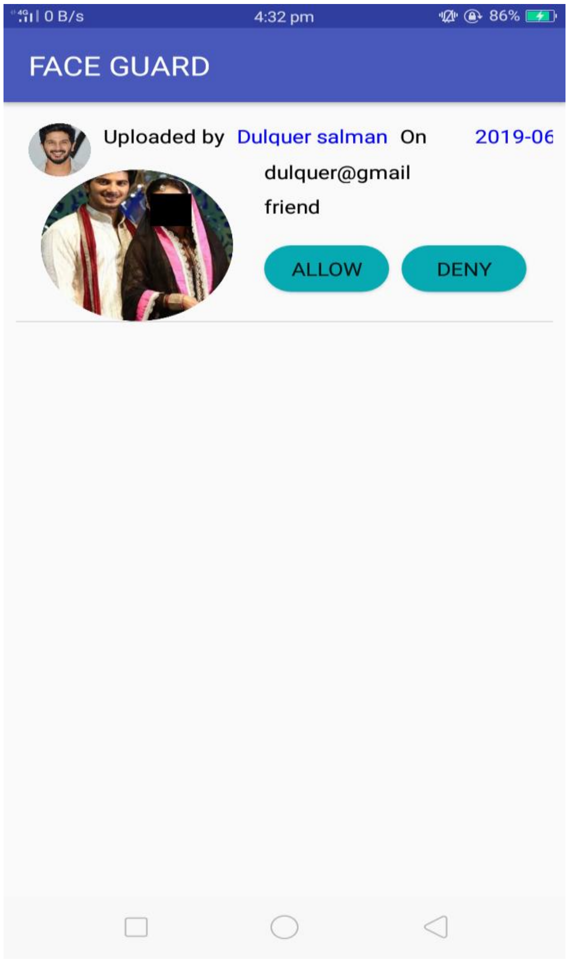
POST IMAGE

NOTIFICATION



friend

Successfully uploaded



Uploaded by Dulquer salman On 2019-06

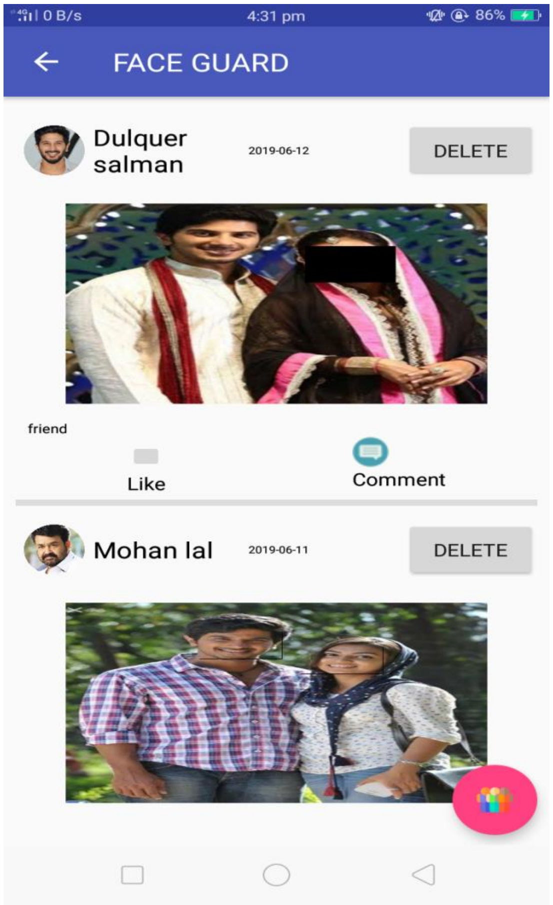
dulquer@gmail

friend

Fig 10:page for posting image

fig 11: notification page

BEFORE APPROVAL



0 B/s 4:31 pm 86%

FACE GUARD

Dulquer salman 2019-06-12 DELETE

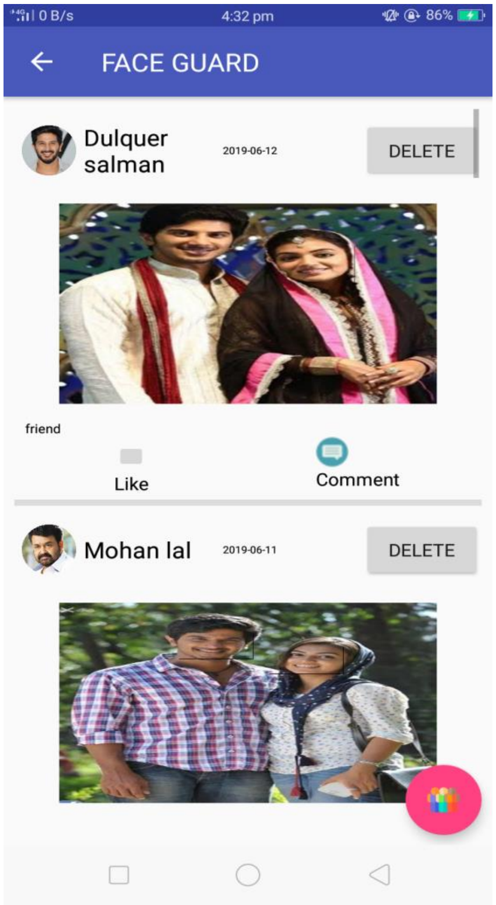
friend

Like Comment

Mohan lal 2019-06-11 DELETE

Fig 12: before approval

AFTER APPROVAL



0 B/s 4:32 pm 86%

FACE GUARD

Dulquer salman 2019-06-12 DELETE

friend

Like Comment

Mohan lal 2019-06-11 DELETE

fig 13 : after approval

APPENDIX B

To post an image

```
def postimg():
    uid=request.form["uid"];
    imgpost=request.form["imgpost"];
    say=request.form["say"];
    a = base64.b64decode(imgpost)
    q="select max(postid) from tbl_post"
    c=conn()
    mxid=c.mid(q)
    print(a)
    path = "/home/god/Desktop/mpmd/MPMD/static/postimage/" + str(mxid)
+ ".jpg"
    # fh=open("static/kkk/" +nam+ ".jpg", "wb")
    fh = open(path, "wb")
    fh.write(a)
    fh.close()
    abc = "static/postimage/" + str(mxid) + ".jpg "
    print(say)
    qrry="insert into tbl_post(uid,photopost,description,date)values
('"+uid+"','"+abc+"','"+say+"',CURDATE())"
    c=conn()
    c.nonreturn(qrry)
    query = "select photo,uid from user_reg where uid !='" + uid + "'"
    c=conn()
    print(query);
    results = c.selectall(query)
    known_faces = []
    userids = []
    if results is not None:
        for result in results:
            img = "/home/god/Desktop/mpmd/MPMD/"+result[0].rstrip()
            print(img)
            b_img = face_recognition.load_image_file(img)
            b_imgs = face_recognition.face_encodings(b_img)[0]
            known_faces.append(b_imgs)
            userids.append(result[1])
            print(img + "done")
    print(path)
    unknown_image = face_recognition.load_image_file(path)
    try:
        m = len(face_recognition.face_encodings(unknown_image))
        facelocations = face_recognition.face_locations(unknown_image)
        for a in range(m):
            unknown_face_encoding =
face_recognition.face_encodings(unknown_image)[a]
            results = face_recognition.compare_faces(known_faces,
unknown_face_encoding,tolerance=0.5)
            for i in range(len(results)):
                if results[i] == True:
                    ukimg = Image.open(path)
                    spic = Image.open(path)
                    top, right, bot, lef = facelocations[a]

flox=str(lef)+","+str(top)+","+str(right)+","+str(bot)
                    s="select max(notid) from tbl_notification"
                    c = conn()
                    mid=c.mid(s)
                    qry = "insert into tbl_notification
(uid,postid,status,date,floc)values('" + str(userids[i]) + "','" +
str(mxid) + "','pending',curdate(),'"+flox+"')"
                    c.nonreturn(qry)
                    box = (lef, top, right, bot)
                    print("boxx")
                    cropped_image = ukimg.crop(box)
                    pth = "/home/god/Desktop/mpmd/MPMD/static/notif/"
+ str(mid) + ".jpg"
                    cropped_image.save(pth)
                    dr = ImageDraw.Draw(spic)
                    dr.rectangle(box, fill=(0, 0, 0))
                    spic.save(path)
        # img.show()
    except Exception as ex:
        print(ex)
    return jsonify(status='ok',psid=mxid)
```

Accepting and rejecting request to post image

```
@app.route("/acceptrejectpost",methods=['POST'])
def acceptrejectpost():
    postid=request.form["postid"]
    uid=request.form["uid"]
    status=request.form["status"]
    k = "select notid,floc from tbl_notification where postid='" +
postid + "' and uid='" + uid + "'"
```

```
    c = conn()
    kk=c.selectone(k)
    if kk is not None:
        if status=="accepted":
            floc=kk[1].split(',')
            ogpathnot="/home/god/Desktop/mpmd/MPMD/static/notif/" +
str(kk[0]) + ".jpg"
            pth = "/home/god/Desktop/mpmd/MPMD/static/postimage/" +
postid + ".jpg"
            image=Image.open(pth)
            logo=Image.open(ogpathnot)
            position=(int(floc[0]),int(floc[1]))
            image.paste(logo,position)
            image.save(pth)
            ss = "update tbl_notification set status='DONE' where
notid='" + str(kk[0]) + "'"
            print(ss)
            c.nonreturn(ss)
        else:
            ss = "update tbl_notification set status='"+status+"'
where notid='" + str(kk[0]) + "'"
            print(ss)
            c.nonreturn(ss)
    return jsonify(status="ok")
```

Android activity code

Post

```
public class custom_post extends BaseAdapter {

   String[] name, usrimg, post,date,discription;

    private Context context;

    public custom_post(Context appcontext, String[] name1, String[]
usrimg1, String[] date1, String[] post, String[] discription) {

        this.context = appcontext;

        this.name = name1;

        this.usrimg = usrimg1;
        this.post = post;

        this.date = date1;

        this.discription=discription;

    }

    @Override

    public int getCount() {

        return name.length;

    }



    @Override

    public Object getItem(int i) {

        return null;

    }



    @Override

    public long getItemId(int i) {

        return 0;

    }



    @Override

    public View getView(int i, View view, ViewGroup viewGroup) {

        LayoutInflater inflator = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
```

```
        View gridView;

        if (view == null) {

            gridView = new View(context);

            //gridView=inflator.inflate(R.layout.customview, null);

            gridView = inflator.inflate(R.layout.custom_post, null);

        } else {

            gridView = (View) view;

        }

        TextView tv_name = (TextView)
gridView.findViewById(R.id.textView25);

        TextView tv_date = (TextView)
gridView.findViewById(R.id.textView26);

        TextView tv_dis = (TextView)
gridView.findViewById(R.id.textView27);

        ImageView im_usr = (ImageView)
gridView.findViewById(R.id.imageView9);

        ImageView im_post = (ImageView)
gridView.findViewById(R.id.imageView10);



        tv_name.setTextColor(Color.BLACK);

        tv_name.setText(name[i]);

        tv_dis.setTextColor(Color.BLACK);

        tv_dis.setText(discription[i]);
        tv_date.setTextColor(Color.BLACK);

        tv_date.setText(date[i]);




        SharedPreferences sh =
PreferenceManager.getDefaultSharedPreferences(context);

        String ip = sh.getString("ip", "");

        String url = "http://" + ip + ":5000" + usrimg[i];

        Picasso.with(context).load(url).into(im_usr);

        String url1 = "http://" + ip + ":5000" + post[i];

        Picasso.with(context).load(url).into(im_post);


        return gridView;

    }

}
```

Home page

```
public class home_page extends AppCompatActivity
        implements NavigationView.OnNavigationItemSelectedListener {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home_page);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        FloatingActionButton fab = (FloatingActionButton)
findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Snackbar.make(view, "Replace with your own action",
Snackbar.LENGTH_LONG)
                        .setAction("Action", null).show();
            }
        });

        DrawerLayout drawer = (DrawerLayout)
findViewById(R.id.drawer_layout);
        ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
                this, drawer, toolbar,
R.string.navigation_drawer_open, R.string.navigation_drawer_close);
        drawer.addDrawerListener(toggle);
        toggle.syncState();

        NavigationView navigationView = (NavigationView)
findViewById(R.id.nav_view);
        navigationView.setNavigationItemSelectedListener(this);
    }

    @Override
    public void onBackPressed() {
        DrawerLayout drawer = (DrawerLayout)
findViewById(R.id.drawer_layout);

        if (drawer.isDrawerOpen(GravityCompat.START)) {

            drawer.closeDrawer(GravityCompat.START);

        } else {

            super.onBackPressed();

        }
```

Notification

```
public class Custom_notification extends BaseAdapter {

    String[] name, frndimg, notfn,date;


    private Context context;


    public Custom_notification(Context appcontext, String[] name1,
String[] frndimg1, String[] notfn1, String[] date1) {

        this.context = appcontext;

        this.name = name1;

        this.frndimg = frndimg1;

        this.notfn = notfn1;

        this.date = date1;

    }


    @Override

    public int getCount() {

        return notfn.length;

    }
    @Override
    public Object getItem(int i) {
        return null;
    }

    @Override
    public long getItemId(int i) {
        return 0;
    }

    @Override
    public View getView(int i, View view, ViewGroup viewGroup) {
        LayoutInflater inflator = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);

        View gridView;
        if (view == null) {
            gridView = new View(context);
            //gridView=inflator.inflate(R.layout.customview, null);
            gridView = inflator.inflate(R.layout.custom_notification,
null);

        } else {
            gridView = (View) view;

        }
        TextView tv1 = (TextView)
gridView.findViewById(R.id.textView15);
```

```
        TextView tv2 = (TextView)
gridView.findViewById(R.id.textView19);

        TextView tv3 = (TextView)
gridView.findViewById(R.id.textView20);

        ImageView im = (ImageView)
gridView.findViewById(R.id.imageView4);


        tv1.setTextColor(Color.BLACK);

        tv1.setText(name[i]);

        tv2.setTextColor(Color.BLACK);

        tv2.setText(notfn[i]);

        tv3.setTextColor(Color.BLACK);

        tv3.setText(date[i]);




        SharedPreferences sh =
PreferenceManager.getDefaultSharedPreferences(context);

        String ip = sh.getString("ip", "");

        String url = "http://" + ip + ":5000/" + frndimg[i];

        Picasso.with(context).load(url).into(im);


        return gridView;

    }

}

public class Custom_chat extends BaseAdapter {

    String[] name, frndimg, text, time;


    private Context context;
    public Custom_chat(Context appcontext, String[] name1, String[]
frndimg1, String[] text1, String[] time1) {

        this.context = appcontext;

        this.name = name1;

        this.frndimg = frndimg1;

        this.text = text1;

        this.time = time1;

    }


    @Override

    public int getCount() {

        return name.length;

    }


    @Override

    public Object getItem(int i) {

        return null;

    }


    @Override

    public long getItemId(int i) {

        return 0;

    }


    @Override

    public View getView(int i, View view, ViewGroup viewGroup) {

        LayoutInflater inflator = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);


        View gridView;
        if (view == null) {

            gridView = new View(context);

            //gridView=inflator.inflate(R.layout.customview, null);

            gridView = inflator.inflate(R.layout.custom_chat, null);

        } else {

            gridView = (View) view;

        }

        TextView tv1 = (TextView)
gridView.findViewById(R.id.textView16);

        TextView tv2 = (TextView)
gridView.findViewById(R.id.textView18);

        EditText tv3 = (EditText)
gridView.findViewById(R.id.editText15);

        ImageView im = (ImageView)
gridView.findViewById(R.id.imageView6);


        tv1.setTextColor(Color.BLACK);

        tv1.setText(name[i]);

        tv2.setTextColor(Color.BLACK);

        tv2.setText(time[i]);

        tv3.setTextColor(Color.BLACK);

        tv3.setText(text[i]);

        SharedPreferences sh =
PreferenceManager.getDefaultSharedPreferences(context);

        String ip = sh.getString("ip", "");

        String url = "http://" + ip + ":5000/" + frndimg[i];

        Picasso.with(context).load(url).into(im);


        return gridView;

    }

}
```

APPENDIX C

SUMMARY TERMS

Deep learning

Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. These methods have dramatically improved the state-of-the-art in speech recognition, visual object recognition, object detection and many other domains such as drug discovery and genomics.

Base 64

Base64 is a way in which 8 bit binary data is encoded into a format that can be represented in 7 bits. This is done using only the character A-Z ,a-z, 0-9, + and /. Inorder to represent data , with = used to pad data. For instance , using this encoding ,three 8-bit bytes are converted into four 7-bit bytes.