# DEVELOPMENT OF AN ANDROID BASED MOBILE APPLICATION TO COLLECT GAS METER READING WITH APPROPRIATE CUSTOMER'S APPROVAL

**by**
**MD. MUHIT KABIR SARNEABAT**

**POST GRADUATE DIPLOMA IN INFORMATION AND COMMUNICATION TECHNOLOGY**

**Institute of Information and Communication Technology (IICT)**
**BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY (BUET)**
**September 2018**

The project report titled "**DEVELOPMENT OF AN ANDROID BASED MOBILE APPLICATION TO COLLECT GAS METER READING WITH APPROPRIATE CUSTOMER'S APPROVAL**" submitted by Md. Muhit Kabir Sarneabat, Roll No: 1009311002P, Session: October-2009 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of Post Graduate Diploma (ICT) held on 08th September, 2018.

## BOARD OF EXAMINERS

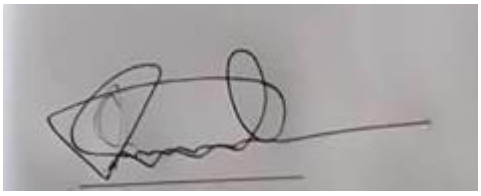
1. **Dr. Md. Rubaiyat Hossain Mondal** — Chairman (Supervisor)
   Associate Professor
   IICT, BUET, Dhaka

2. **Dr. Md. Saiful Islam** — Member
   Professor
   IICT, BUET, Dhaka

3. **Dr. Md. Liakot Ali** — Member
   Professor
   IICT, BUET, Dhaka

# CANDIDATE'S DECLARATION

It is hereby declared that this project report or any part of it has not been submitted elsewhere for the award of any degree or diploma.



_____

Md. Muhit Kabir Sarneabat

**Dedicated to**

**The Sacrifices of**

**My Parent**

**Table of Contents**

**List of Figures**

**List of Abbreviations**

| Abbreviations | Elaboration |
|---|---|
| AI | Artificial Intelligence |
| DBMS | Database Management System |
| DFD | Dataflow Diagram |
| E-R | Entity-Relationship |
| ERD | Entity relationship Diagram |
| ERP | Enterprise Resource Planning System |
| OAS | Office Automation System |
| RAD | Rapid Application Development |
| OOD | Object Oriented Design |
| SDLC | Systems Development Life Cycle |
| SQL | Structured Query Language |
| WWW | World Wide Web |

# Acknowledgement

At first, I would like to thank and all praise to the Almighty Allah, the most merciful, the most gracious, the source of knowledge and wisdom endowed to mankind, who provided me with the power of mind, strength, patience and capability to carry me through the work and enable me to complete this project.

I would like to thank my Project Supervisor, Dr. Md. Rubaiyat Hossain Mondal, Associate Professor, Institute of Information and Communication Technology, Bangladesh University of Engineering and Technology, Dhaka, Bangladesh for his kind, constant, and inspiring guidance, close encouragement, advice, and valuable suggestions at all stages for preparing this dissertation. I also remember the profound contribution, guideline, inspiration and kind support of my earlier supervisors Dr. Mohammad Shah Alam, Associate Professor and Mohammad Imam Hasan Bin Asad, Lecturer throughout this persuation.

I would like to thank Prof. Dr. Md. Saiful Islam and Prof. Dr. Md. Liakot Ali, of Institute of Information and Communication Technology, Bangladesh University of Engineering and Technology, Dhaka, Bangladesh for their very valuable and cordial guidelines, inspirations, support and instructions in this regard.

In completing this project, I have been fortunate to have help, support and encouragement from many people. I would like to acknowledge them for their cooperation.
I would like to thank my family, specially my parents for their continuous support and inspiration throughout the whole period of this undertaking.

# Abstract

There are different types of reading collection processes using different types of meters or additional hardware across the world by different gas distribution organizations. In this project, a generalized Android application has been designed for the gas consumption metering purpose for gas distribution organization. This Android application system considers a metering system which includes a gas meter, meter reading collection, customers' verification from customer point of view, data processing, consumption calculation, bill processing, etc. Variation of billing system from one utility company to another are pointed out and generalization is devised in this design of this Android application. Existing billing system of local gas Distribution Company has been considered as standard in proposed system during design phase. User friendly nature has been given a priority. This method can be very cost-effective and can be used for household premises where proper billing system is the main concern. In the proposed system, subscriber can have easy access to billing system and meter reader resulting prompt meter reading, transparency, greater accountability and subsequently company may have greater revenue collection. The design has been completed by drawing the E-R model for the application. The logical design is then extended to physical model which in turn is used to design the table structures for the proposed database. This Android application has been developed using Cordova platform through command window, phonegap, Apache ANT, bootstrap, HTML 5, Java Script and CSS. Using this Android application meter reader captures the image and sends to the server. The server does pre-processing on image and OCR may extract Meter reading and Customer number. Using these extracted details Bill may be generated, updating the database and PDF of the bill may be send via mail to the customer. Based on the proposed application system, a generalized database has been developed utilizing PHP and MySQL. Finally, the implementation guideline of the proposed application and database has been formulated. The software issues during implementation of the designed database have been addressed in this project. The developed application is likely to play an important role in the automation of the meter reading collection system of gas utility companies.

# CHAPTER ONE

# Introduction

## 1.1 Introduction:

In our country, industrial uses of gas resources began back in 1959. A major portion of gas distribution system is used for household purposes. At present, major portion of gas resources are used for industrial use as well electricity production.

The gas distribution network is dominated by the Titas Gas Company, as well as regional companies in north Bengal and Sylhet. Titas Gas is the largest distributor in Bangladesh.

Billing and collection for the gas usages is one of the most important activities of gas distribution organization. Various techniques have been practiced in order to find best method of billing and collection. Most importantly, because of rapid development of technology, new sophisticated methods have been proposed and implemented.

Gas distribution companies depends heavily on manual process billing system in industrial uses. Manual procedure may have some faults in bill generation process as well time consuming. This procedure makes a lot of burden on meter reader and customer, as well create inconvenience for the customer to lodge any complaints. With advancement of technology things are becoming simpler and easier for us. Using the Android application meter reader captures the image of meter reading, reading data and sends to the server, upon verification of Consumer. The server does processing on image, stores meter reading data, prepare and process customer wise gas consumption details in a single platform. Using these extracted details Bills may be generated, updating the database and PDF of the bill may be send via mail to the customer at implementation phase through a report generation software. Thus this automated process reduces workload on the employees and incorrectness in bill generation and saves revenues. The time consuming process may be turned to fast and completely automated.

### 1.1.1 Motivation

Android based billing system is a modern choice for any utility providing company. It is difficult to maintain billing procedures and preserving consumption related data of a company in a very conventional way. Huge space and man power is required for bookkeeping as well record maintenance. So, it comes to the consideration to make an automated system. Today, internet is the important technology on the world stage. Consumer wants proper and hassle free billing system, and distribution company focuses on greater revenue collection and reduction of system loss. Our "Android Based Mobile Application To Collect Gas Meter Reading With Appropriate Customer's Approval" may serve the purpose and deliver a majestic way of ultimate profit at a glance.

### 1.1.2 Overview

After analyzing different types of reading collection processes using special types of meters or additional hardware across the world thoroughly, we reached a conclusion about what will be the functionality of meter reading related mobile application. In the proposed design practice, every month a meter reader shall go to customer premises, takes the meter reading and subsequently take a snapshot which is uploaded to the utility company software for consumption calculation. Apart from the bill calculation the company also analyzes the data to find out if there is any discrepancy or misuse. Sometimes discrepancy happens due to human error and another person has to go to the customer premise to check the actual reading and correct the data. If the discrepancy found is not due to the reading error but for the misuse, the company can then take necessary actions. But finding out these types of cases require some days due to present metering system. In the case of admin, an admin can add, edit and delete and retrieve the information according to the accessibility of the system and in the case of user, a subscriber can only view the information according to the accessibility and exactly what we have explained, our system acts accordingly. In essence, our system plays role of the entire billing process.

### 1.1.3 Goal

The project work will focus on the mechanism of proper billing system which will ensure proper usages of natural gas resources and bring proper billing system in distribution channel. Automation is the use of control systems and information technologies to reduce the need for human work in the production of goods and services. The solution to the above problem is

proposed which makes advanced version of current procedure. In this project we suggest Android application which is carried by meter reader and a web application for customer to interact with electricity supplier companies.

### 1.1.4 Objective with specific aims and possible outcome

The aim of this project is to develop an Android based meter reading system. The following objectives will be achieved:

(i) To build an Android application to take meter reading and corresponding data.
(ii) To store and update detailed billing information to ensure transparency and accountability.
(iii) To ensure customers participation and authenticity of reading.
(iv) To make meter reading process easier and transparent.

The outcome of this project is a prototype of effective gas meter reading system.

## 1.2 Different Billing and Collection System and adoptability of the project work:

The billing and collection systems used in Utility Distribution Companies can be classified in different way. On the basis of location of metering, all Systems can be divided into 2 categories. They are:

1. On-site metering.
2. Remote metering.

The project may be used for both type of metering system.

On the basis of payment (collection) methods, the metering system can again be subdivided into two categories. They are:

1. Post-paid metering.
2. Prepaid metering.

As the name suggest, the post-paid metering system is the system in which utility is used first and the bill is paid later. On the other hand , in prepaid system consumers pay first and use the electricity later. Because of several difficulties of postpaid system, the prepaid metering system is gaining popularity in Bangladesh. The project may be used for prepaid metering system.

## 1.3 Organization of the Project Report:

Chapter 1: Introduction: The project documentation starts with the introduction of an Android Based Mobile Application To Collect Gas Meter Reading with a discussion on the existing systems available. It is followed by Motivation, Overview, Goal, Objective of the specific aims and possible outcome, different billing and collection system and adoptability of the project work and organization of the documentation.

Chapter 2: System Design and Development Methodology: In this chapter, the detailed design and development of an Android application is described along with System Development Life cycle (SDLC) and its implementation, different SDLC methodologies, Application of RAD, E-R Diagram and Different Risk and its mitigation are discussed.

Chapter 3: Existing System Analysis & Proposed System: Existing system analysis and proposed system and user group is analyzed.

Chapter 4: System Design & Tools: In this chapter, the detailed description of the components used is described along with Data Flow Diagram, E-R Diagram, UML Diagram, Design Tools of the project work and procedure of the design process has been discussed.

Chapter 5: Implementation & Testing: In this chapter, the detailed simulation results, different interface with different utility, interrelation and correlation of Android application and database of the project work is elaborated.

Chapter 6: Conclusion: Finally achievement, limitation as well future scope for further work and project's significance in implementation phase are discussed.

# CHAPTER TWO

# System Design and Development Methodology

Once upon a time, software development consisted of a programmer writing code to solve a problem or automate a procedure. Nowadays, systems are so big and complex that teams of architects, analysts, programmers, testers and users must work together to create the millions of lines of custom-written code that drive our enterprises.

To manage this, a number of system development life cycle (SDLC) models have been created: waterfall, fountain, spiral, build and fix, rapid prototyping, incremental, and synchronize and stabilize.

## 2.1 System Development Life cycle (SDLC):

In a 1991 Information Center Quarterly article, Larry Runge said that SDLC "works very well when we are automating the activities of clerks and accountants. It doesn't work nearly as well, if at all, when building systems for knowledge workers -- people at help desks".

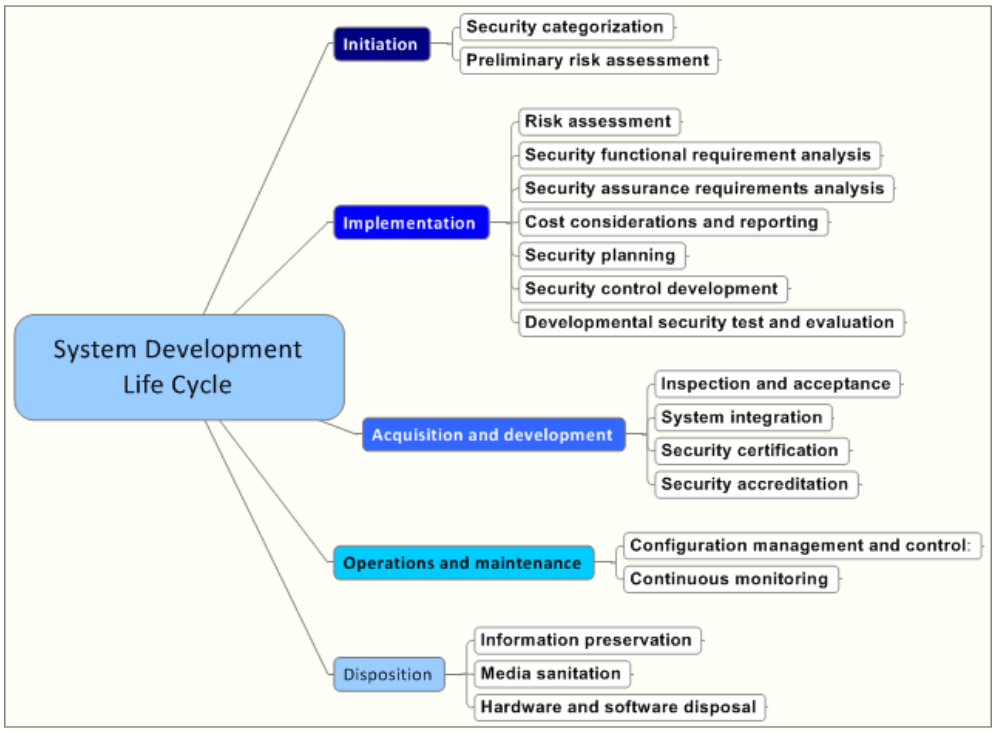



**Fig. No. 2.1 System Development Life Cycle**

**a. Planning**

This is the first phase in the systems development process. It identifies whether there is the need for a new system to achieve a business's strategic objectives. This is a preliminary plan (or a feasibility study) for a company's business initiative to acquire the resources to build on an infrastructure to modify or improve a service. The purpose of this step is to find out the scope of the problem and determine solutions. Resources, costs, time, benefits and other items should be considered at this stage.

**b. Systems Analysis and Requirements**

The second phase is where businesses will work on the source of their problem or the need for a change. In the event of a problem, possible solutions are submitted and analyzed to identify the best fit for the ultimate goal(s) of the project. It is also where system analysis takes place—or analyzing the needs of the end users to ensure the new system can meet their expectations. There are several tools businesses can use that are specific to the second phase. They include:

- CASE (Computer Aided Systems/Software Engineering)
- Requirements gathering
- Structured analysis

**c. Systems Design**

The third phase describes, in detail, the necessary specifications, features and operations that will satisfy the functional requirements of the proposed system which will be in place. It"s during this phase that they will consider the essential components (hardware and/or software) structure (networking capabilities), processing and procedures for the system to accomplish its objectives.

**d. Development**

The fourth phase is when the real work begins—in particular, when a programmer, network engineer and/or database developer are brought on to do the major work on the project. This work includes using a flow chart to ensure that the process of the system is properly organized. The development phase marks the end of the initial section of the process. Additionally, this phase signifies the start of production. The development stage is also

characterized by instillation and change. Focusing on training can be a huge benefit during this phase.

**e. Integration and Testing**

The fifth phase involves systems integration and system testing (of programs and procedures)—normally carried out by a Quality Assurance (QA) professional—to determine if the proposed design meets the initial set of business goals. Testing may be repeated, specifically to check for errors, bugs and interoperability. This testing will be performed until the end user finds it acceptable. Another part of this phase is verification and validation, both of which will help ensure the program's successful completion.

**f. Implementation**

The sixth phase is when the majority of the code for the program is written. Additionally, this phase involves the actual installation of the newly-developed system. This step puts the project into production by moving the data and components from the old system and placing them in the new system via a direct cutover. While this can be a risky (and complicated) move, the cutover typically happens during off-peak hours, thus minimizing the risk. Both system analysts and end-users should now see the realization of the project that has implemented changes.

**g. Operations and Maintenance**

The seventh and final phase involves maintenance and regular required updates. This step is when end users can fine-tune the system, if one wishes, to boost performance, new capabilities or meet additional user requirements have to be meet.

**Importance of the SDLC**

If a business determines a change is needed during any phase of the SDLC, the company might have to proceed through all the above life cycle phases again. The life cycle approach of any project is a time-consuming process. Even though some steps are more difficult than

others, none are to be overlooked. An oversight could prevent the entire system from functioning as planned.1

## 2.2 Different software development lifecycle methodologies

## Software Process Models

A structured set of activities are required to develop a software system. A software process model is an abstract representation of a software process. Each process model represents a process from a particular perspective that only provides partial information about the process. There are several popular process models such as

1. Agile Model
2. Lean Model
3. Waterfall Model
4. Iterative Model
5. Spiral Model
6. DevOps Model

Each of these approaches varies in some ways from the others, but all have a common purpose: to help teams deliver high-quality software as quickly and cost-effectively as possible.

Reviewing a brief description of the six most common SDLC methodologies may help to decide which is best for a team:

### 2.2.1. Agile Model

The Agile model has been around for about a decade. But lately, it has become a major driving force behind software development in many organizations. Some businesses value the Agile methodology so much that they are now applying it to other types of projects, including non-tech initiatives.

[1]https://www.computerworld.com/article/2576450/app-development/app-development-system-development-life-cycle.html

In the Agile model, “fast failure” is a good thing. The approach produces ongoing release cycles, each featuring small, incremental changes from the previous release. At each iteration, the product is tested. The Agile model helps teams identify and address small issues on projects before they evolve into more significant problems, and engage business stakeholders and get their feedback throughout the development process.

As part of their embrace of this methodology, many teams are also applying an Agile framework known as Scrum to help structure more complex development projects. Scrum teams work in “sprints,” which usually last two to four weeks, to complete assigned tasks. Daily Scrum meetings help the whole team monitor progress throughout the project. And the ScrumMaster is tasked with keeping the team focused on its goal.

### 2.2.2. Lean Model

The Lean model for software development is inspired by lean manufacturing practices and principles. The seven Lean principles (in this order) are: eliminate waste, amplify learning, decide as late possible, deliver as fast as possible, empower the team, build integrity in, and see the whole.

The Lean process is about working only on what must be worked on at the time, so there’s no room for multitasking. Project teams are also focused on finding opportunities to cut waste at every turn throughout the SDLC process, from dropping unnecessary meetings to reducing documentation.
The Agile model is actually a Lean method for the SDLC, but with some notable differences. One is how each prioritizes customer satisfaction: Agile makes it the top priority from the outset, creating a flexible process where project teams can respond quickly to stakeholder feedback throughout the SDLC. Lean, meanwhile, emphasizes the elimination of waste as a way to create more overall value for customers — which, in turn, helps to enhance satisfaction.

### 2.2.3. Waterfall Model

The Waterfall Model is a sequentialsoftware development model (a process for the creation of software) in which development is seen as flowing steadily downwards (like a waterfall)

through the phases of Requirements Analysis and Definition, System and Software Design, Implementation and Unit Testing, Integration and System Testing and Operation and Maintenance.

The downside of Waterfall is its rigidity. Sure, it's easy to understand and simple to manage. But early delays can throw off the entire project timeline. With little room for revisions once a stage is completed, problems can't be fixed until one get to the maintenance stage. This model doesn't work well if flexibility is needed or if the project is long term and ongoing.

Even more rigid is the related Verification and Validation model — or V-shaped model. This linear development methodology sprang from the Waterfall approach. It's characterized by a corresponding testing phase for each development stage. Like Waterfall, each stage begins only after the previous one has ended. This SDLC model can be useful, provided your project has no unknown requirements.

Some experts argue that the Waterfall model was never meant to be a process model for real projects, the Waterfall model is widely considered the oldest of the structured SDLC methodologies. It's also a very straightforward approach: finish one phase, then move on to the next. No going back. Each stage relies on information from the previous stage and has its own project plan.

The phases are described below:

a) **Requirements Analysis and Definition:** the system's services, constraints and goals are established by consultation with system users. They are then defined in detail and serve as a system specification.

b) **System and Software Design:** the system's design process partitions the requirements to either hardware or software systems. It establishes overall system architecture. Software design involves identifying and describing the fundamental software system abstractions and their relationships.

c) **Implementation and Unit Testing:** during this stage, the software design is realized as a set of programs or program units. Unit testing involves verifying that each unit meets its specification.

d) **Integration and System Testing:** the individual program units or programs are integrated and tested as a complete system to ensure that the software requirements have been met. After testing, the software system is delivered to the customer.

e) **Operation and Maintenance**: Normally (although not necessarily) this is the longest life-cycle phase. The system is installed and put into practical use. Maintenance involves correcting errors which were not discovered in earlier stages of the life cycle and improving the implementation of system units and enhancing the system's services as new requirements are discovered.

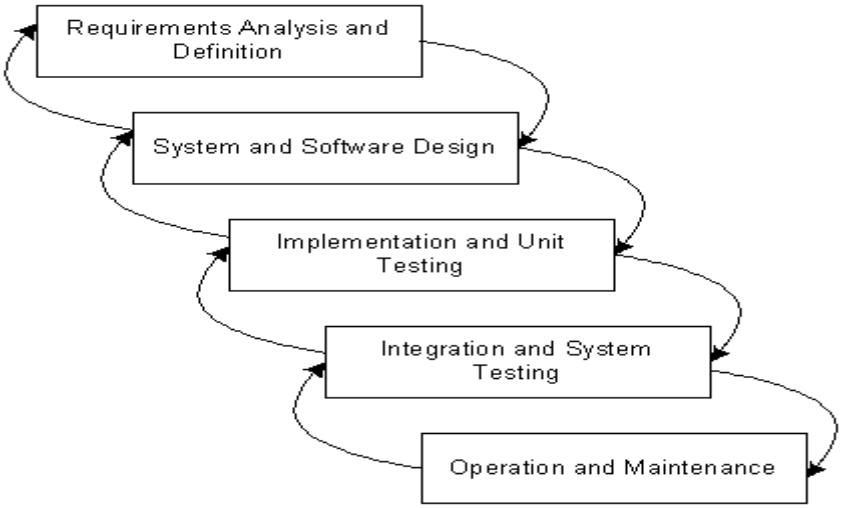



**Fig: 2.2 Flowchart of Waterfall Model**

Some major advantages and disadvantages of Waterfall Model are given below.

**Advantages:**

1. Having fewer errors.
2. System requirements are identified before programming starts.
3. Changes are minimized to the requirements as the project proceeds.

**Disadvantages:**

1. Most time consuming process model.
2. Huge amount of paperwork is necessary.
3. Does not allow going next stage before finishing the previous stage.

## 2.2.4. Iterative Model

The Iterative model is repetition incarnate. Instead of starting with fully known requirements, project teams implement a set of software requirements, then test, evaluate and pinpoint further requirements. A new version of the software is produced with each phase, or iteration. Rinse and repeat until the complete system is ready.

Advantages of the Iterative model over other common SDLC methodologies is that it produces a working version of the project early in the process, and makes it less expensive to implement changes. One disadvantage: Repetitive processes can consume resources quickly.

One example of an Iterative model is the Rational Unified Process (RUP), developed by IBM's Rational Software division. As explained in this document from IBM, RUP is a "process product" designed to enhance team productivity that also "captures many of the best practices in modern software development in a form that is suitable for a wide range of projects and organizations."

RUP divides the development process into four phases: inception, when the idea for a project is set; elaboration, when the project is further defined, and resources are evaluated; construction, when the project is developed and completed; and transition, when the product is released. Each phase of the project involves business modeling, analysis and design, implementation, testing, and deployment.

### 2.2.5. Spiral Model:

One of the most flexible SDLC methodologies, the Spiral model takes a cue from the Iterative model and its repetition; the project passes through four phases (planning, risk analysis, engineering and evaluation) over and over in a "spiral" until completed, allowing for multiple rounds of refinement.

The Spiral model is typically used for large projects. It enables development teams to build a highly customized product, and incorporate user feedback early on in the project. Another benefit of this SDLC model is risk management. Each iteration starts by looking ahead to potential risks, and figuring out how best to avoid or mitigate them.

### 2.2.6. DevOps Model

The DevOps methodology is the newcomer to the SDLC scene. As this article explains, it emerged from two trends: the application of Agile and Lean practices to operations work, and the general shift in business toward seeing the value of collaboration between development and operations staff at all stages of the SDLC process.

In a DevOps model, Developers and Operations teams work together closely — and sometimes as one team — to accelerate innovation and the deployment of higher-quality and more reliable software products and functionalities. Updates to products are small but frequent. Discipline, continuous feedback and process improvement, and automation of manual development processes are all hallmarks of the DevOps model.

Amazon Web Services describes DevOps like this: "DevOps is the combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes." So, like many SDLC models, DevOps is not only an approach to planning and executing work, but also a philosophy that demands significant mindset and culture changes in an organization.2

Choosing the right SDLC model for your software development project will require careful thought. But keep in mind that a methodology for planning and guiding your project is only one ingredient for success. Even more important is assembling a solid team of skilled talent committed to moving the project forward through every unexpected challenge or setback.3

## 2.3 Rapid Application Development:

RAD (rapid application development) is a concept that was born out of frustration with the waterfall software design approach which too often resulted in products that were out of date or inefficient by the time they were actually released. The term was inspired by James Martin, who worked with colleagues to develop a new method called Rapid Iterative Production Prototyping (RIPP). In 1991, this approach became the premise of the book Rapid Application Development.

Gathering requirements using workshops or focus groups

Prototyping and early, reiterative user testing of designs

The re-use of software components

A rigidly paced schedule that defers design improvements to the next product version

Less formality in reviews and other team communication

[2]https://existek.com/blog/sdlc-models/

[3]https://www.roberthalf.com/blog/salaries-and-skills/6-basic-sdlc-methodologies-which-one-is-best

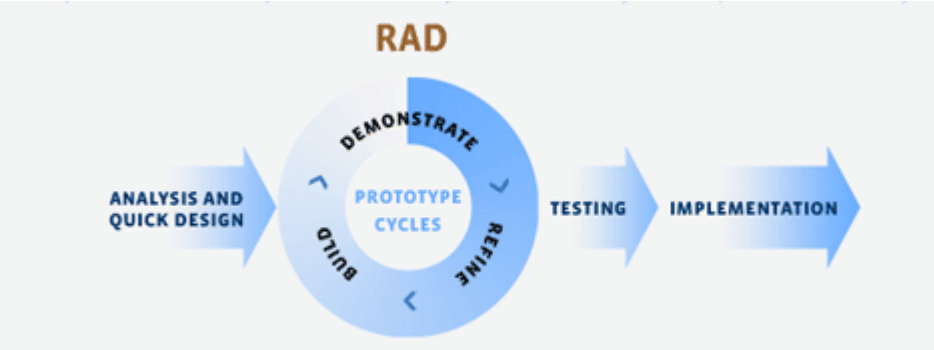



Fig:2.3: RAD Development Diagram

Rapid application development is still in use today and some companies offer products that provide some or all of the tools for RAD software development. (The concept can be applied to hardware development as well.) These products include requirements gathering tools, prototyping tools, computer-aided software engineering tools, language development environments such as those for the Java platform, groupware for communication among development members, and testing tools.

RAD usually embraces object-oriented programming methodology, which inherently fosters software re-use. The most popular object-oriented programming languages, C++ and Java, are offered in visual programming packages often described as providing rapid application development.

## A Few RADicalSteps :

Getting started with rapid application development generally follows a cyclical process that includes four basic steps:

A) Planning Requirements: During this initial stage designers, developers, and users come to a rough agreement on project scope and application requirements, so that future stages with prototyping can begin.

b) User Design: User feedback is gathered with heavy emphasis on determining the system architecture. This allows initial modeling and prototypes to be created. This step is repeated as often as necessary as the project evolves.

c) Rapid Construction: Once basic user and system design has begun, the construction phase is where most of the actual application coding, testing, and integration takes place. Along with User Design, the Rapid Construction phase is repeated as often as necessary, as new components are required or alterations are made to meet the needs of the project.

d) Cutover: The final Cutover (or Transition) stage allows the development team time to move components to a live production environment, where any necessary full-scale testing or team training can take place.

## User Interfacing, Early and Often

In the slow, methodical software development methods of olde, receiving useful and concrete user feedback has been inherently difficult, costly, and time consuming. Long meetings and phone calls, and even longer design docs, were a necessary evil to lay out even the most basic concrete plans of proper software design. With typical waterfall methods, rudimentary user feedback was often many months if not years in the future, after all planning and most development had taken place.

In stark contrast, one of the biggest benefits to rapid application development is the ability to both easily and frequently receive feedback from users who are directly interfacing with the application during development and prototyping. While this advantage is most readily visible within the UI/UX components of the system, iterative design intrinsically means user feedback can be at the forefront of the process.

## A Prototype of Prototyping

While various forms of RAD emphasize slightly different concepts and design methodologies, a common inclusion in most RAD systems is the heavy use of prototyping. As an alternative to heavy-handed design specifications, the use of prototypes throughout the development cycle provides for a number of unique benefits:

User Involvement: Unlike a traditional waterfall model, which requires the design team to discuss with users what features or implementations might be required and plan specifications around those ideas, a rapid prototype allows users to actually use the software and provide feedback on a live system, rather than attempting to provide abstract evaluations of a design document.

Feasibility: Prototyping allows the development team to quickly evaluate the feasibility of a particularly complex or risky component right out of the gate. By recognizing and working on complicated systems early in the development lifecycle, the software will be more robust, less error-prone, and better structured for future design additions.

Error Reduction & Debugging: With rapid prototype releases during a project, it is far more likely that errors will be both discovered (and subsequently squashed) far earlier in the development cycle than with a typical waterfall approach.

## Pros of Rapid Application Development

While there are a number of benefits to using a rapid application development method, we've highlighted a handful of the most crucial when considering whether RAD is the right choice for your next project.

Measurable Progress: With frequent iterations, components, and prototypes coming down the pipe, progress on the overall project, as well as lesser segments, can be easily measured and evaluated to maintain schedules and budgets.

Quickly Generate Productive Code: As a larger percentage of active software developers move into multi-discipline roles (i.e. full-stack developers), a RAD methodology allows skilled team members to quickly produce prototypes and working code to illustrate examples that might otherwise take weeks or months to see the light of day using a slower development technique.

Compartmentalization of System Components: Much in the same way that object-oriented programming practices keep objects and components quarantined from one another, RAD inherently has the same beneficial impact on the components generated during development. By forcing designers and developers to create components that are functional and independent on their own, to be used in an iterative release or prototype, each element within the overall software system is compartmentalized and therefore easily modified as the needs of the software evolve.

Rapid, Constant User Feedback: As discussed above, obtaining relevant user feedback during development is invaluable. RAD methodologies allow for near-constant user interfacing and feedback through frequent iterations and prototype releases, giving the entire team priceless evaluation and criticism when it's needed most.

Early Systems Integration: While most waterfall method software projects must, by their very nature, wait until the tail end of the lifecycle to begin integrations with other systems or services, a rapidly developed application becomes integrated almost immediately. By requiring early integrations within a prototype, a RAD system quickly identifies any errors or complications within integrations and forces immediate resolutions.

Simple Adaptability: During development, software is a fairly malleable form. Since code can be changed that dramatically alters the entire system or generates new components, it is to the advantage of the development team to make use of this flexibility early and often, by iterating and prototyping potential concepts or ideas throughout development.

## Cons of Rapid Application Development

No software development method is without fault and RAD is no different. While the benefits typically outweigh the disadvantages, we'll examine a few of the most likely roadblocks when implementing RAD into a new project.

- Requires Modular Systems: Since each component within the system should be iterable and testable on its own accord, the overall system design when using RAD requires that each component be modular, allowing elements to be swapped in and out or altered by a variety of team members.
- Difficulty Within Large-Scale Projects: While rapid application development methods lead to far greater flexibility during the design and development process, it will also tend to reduce control and restrictions. While this isn't inherently negative, properly managing this added flexibility and volatility within the scope of the whole project can be difficult for larger applications.
- Demands Frequent User Interfacing: Gaining user insight and feedback early and often is certainly a benefit from a design perspective, but this double-edged sword requires that the team be both willing and able to communicate with users on a much more frequent basis, in comparison to a typical waterfall development method.
- Depends Upon Skilled Developers: While many developers these days are multi-disciplined, it is worth noting that use of RAD techniques does require a greater overall skill across the development team, in order to quickly adapt as the system and components evolve.

## 2.4 Entity Relationship Diagram (ERD):

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is a component of data. In other words, ER diagrams illustrate the logical structure of databases.

At first glance an entity relationship diagram looks very much like a flowchart. It is the specialized symbols, and the meanings of those symbols, that make it unique.

## Common Entity Relationship Diagram Symbols

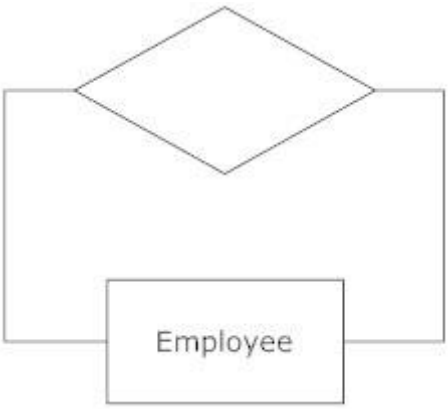
An ER diagram is a means of visualizing how the information a system produces is related. There are five main components of an ERD:

Entities, which are represented by rectangles. An entity is an object or concept about


which you want to store information. A weak entity is an entity that must defined by a foreign key relationship with another entity as it cannot be uniquely identified by its own attributes alone.

Actions, which are represented by diamond shapes, show how two entities share


information in the database. In some cases, entities can be self-linked. For example, employees can supervise other employees.

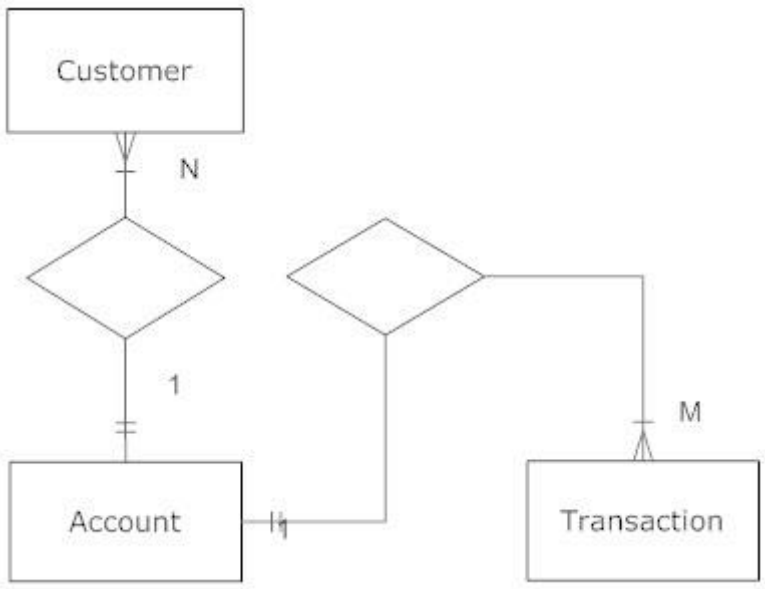



**Attribute:** Attributes, which are represented by ovals. A key attribute is the unique, distinguishing characteristic of the entity. For example, an employee's social security number might be the employee's key attribute.


A multivalued attribute can have more than one value. For example, an employee entity can have multiple skill values. A derived attribute is based on another attribute. For example, an employee's monthly salary is based on the employee's annual salary.

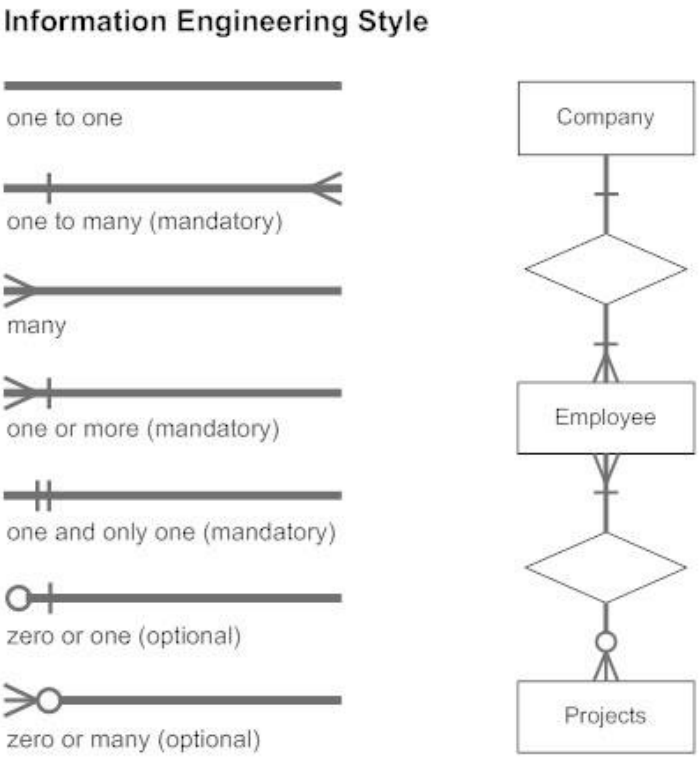
Connecting lines, solid lines that connect attributes to show the relationships of entities in the diagram.

Cardinality specifies how many instances of an entity relate to one instance of another entity. Ordinality is also closely linked to cardinality. While cardinality specifies the occurrences of a relationship, ordinality describes the relationship as either mandatory or optional. In other words, cardinality specifies the maximum number of relationships and ordinality specifies the absolute minimum number of relationships.

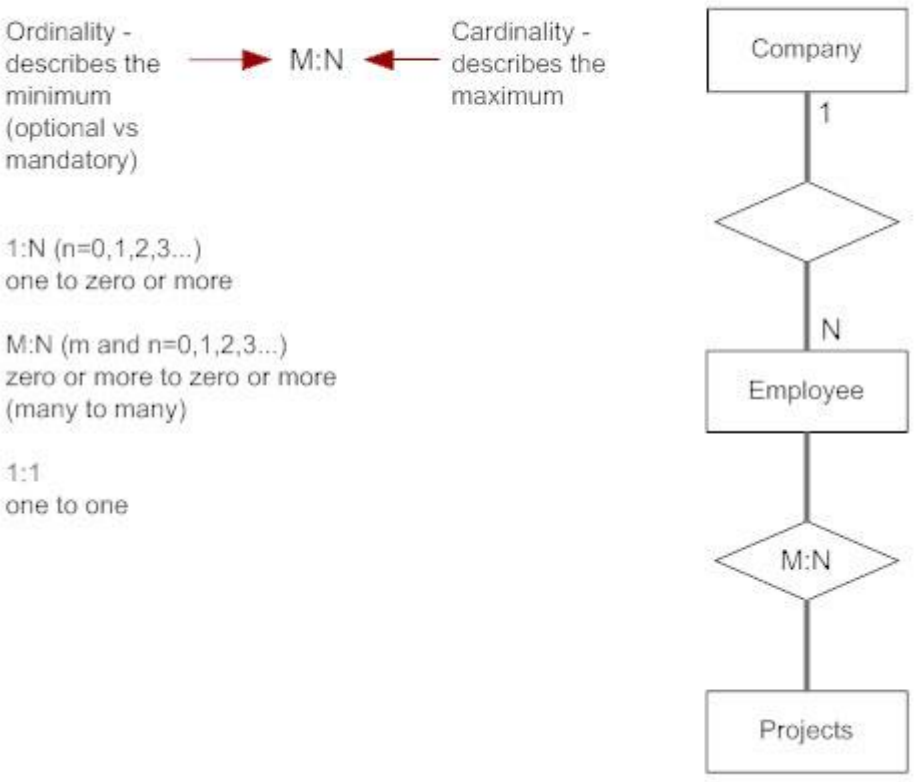



There are many notation styles that express cardinality.

Information Engineering Style

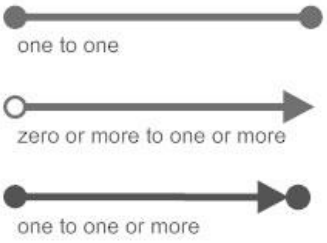


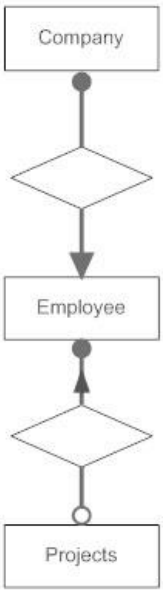

Chen Style




Bachman Style

**Bachman Style**




**Martin Style**

## Martin Style

1 - one, and only one (mandatory)

* - many (zero or more - optional)

1...* - one or more (mandatory)

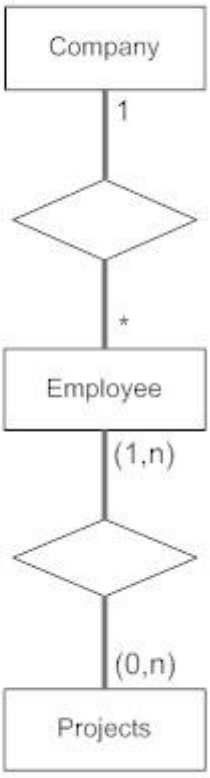
0...1 - zero or one (optional)

(0,1) - zero or one (optional)

(1,n) - one or more (mandatory)

(0,n) - zero or more (optional)

(1,1) - one and only one (mandatory)



## 2.5 Risk Identification

Risk identification is a systematic attempt to specify threats to the project plan (Estimates, schedule, resource loading etc). One method for identifying risk is to create a risk item checklist that can be used for risk identification and focuses on some subset of known and predictable risks in following generic subcategories.

- Product size – risk associated with the overall size of the software to be built.
- Business impact – risks associated with constraints imposed by management or the marketplace.
- Customer's characteristics – risk associated with the sophistication of the customer and the developer's ability to communicate with the customer in a timely manner.
- Process definition – risk associated with the degree to which the software process has been defined and is followed by the development organization.
- Development environment – risk associated with the availability and quality of the tools to be used to build the product.
- Staff size and experienced – risks associated with the overall technical and project experienced of the software engineers who will do this work.

### 2.5.1 Risk Mitigation, Monitoring and Management

All the risk analysis activities presented to this point have a single goal – to assist the project team in developing a strategy for dealing with risk. An effective strategy must consider three issues: Risk avoidance, Risk monitoring and Risk management contingency planning.

To mitigate the risk, project management must develop a strategy for reducing turnover. Among the possible steps to be taken are-

- Meet with current staff to determine causes for turnover.
- Mitigate those causes that are under our control before the project start.
- Once the project commences, assume turnover will occur and develop technique to ensure continuity when people leave.

- Organize project team so that information about each development activity is widely dispersed.
- Define documentation standard and establish mechanisms to ensure that documentations are developed in a timely manner.
- Conduct peer reviews of all work.
- Assign a backup staff member for every critical technology.

The project manager monitors factors that may provide an indicator of whether the risk is becoming more or less likely. The following factors can be monitored:

- General attributes of team members based on the project pressures.
- The degree to which the team has jelled.
- Interpersonal relationships among team members.
- Potential problems with compensation and benefits.

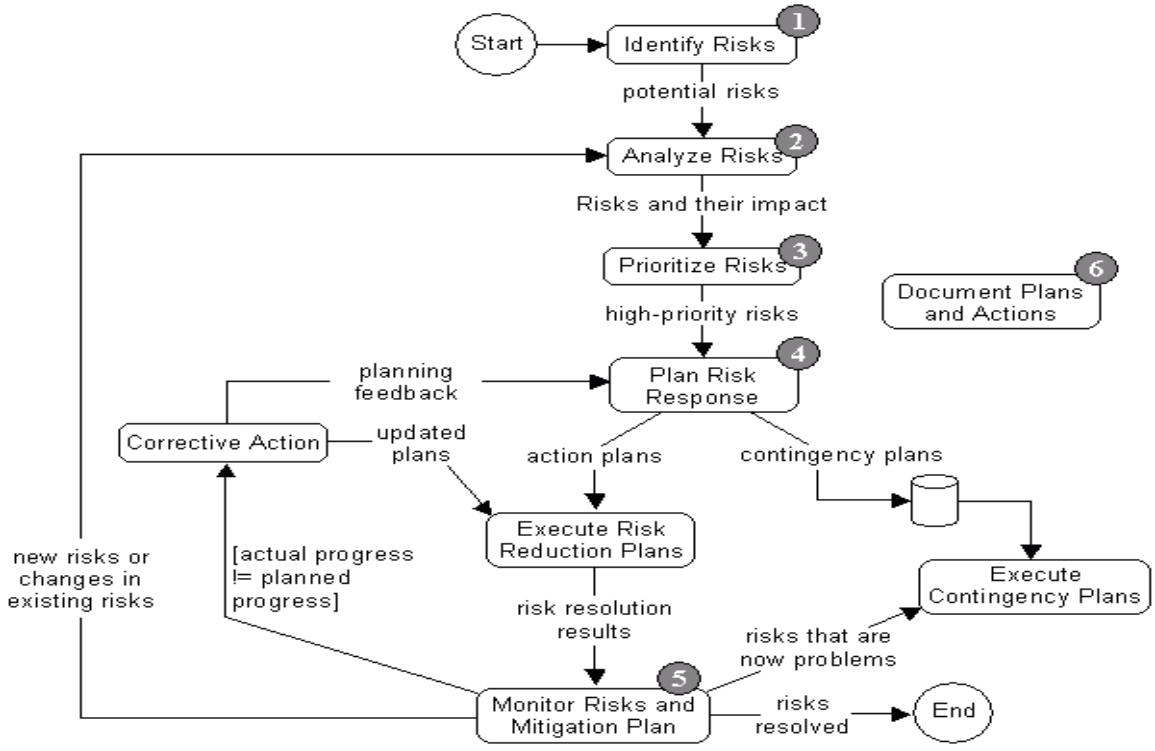



**Fig 2.4: Flowchart of Risk Analysis**

Here, Fig 2.1 shows the step by step functions of risk analysis. In step-1 risks are identified, step-2 analyzes those risks, step-3 prioritizes those risks according to some criteria, step-4 designs plan for those risks, and step-5 monitors risk and mitigation the plan. If any new risk occurs then again step-2 is executed and same procedures (discussed above) are followed. If there is no risk occurs then all risks are solved.

## 2.6 Summary:

Systems analysis and design is a systematic approach to identifying problems, opportunities and objectives, to analysing the information flow of the organization. End user involvement is critical to success. SDLC is identifying problems, opportunities and objectives, determining information requirements, analysing system needs, designing the recommended system and implementing and evaluating the system. Object oriented system UML objects are created include not only code, but also instruction to be performed on the data. Entity – relationship diagrams help the systems analyst understand the entity and relationship that compromise the organizational system. E-R diagrams can depict a one to one relationship, one to many relationship and many to many relationship. Three broad organizational fundamentals to consider while analysing and designing information systems, the various level of management and overall organizational culture. This chapter has discussed about the Requirement Engineering & Software Process Model. Next chapter will discuss about the Existing System Analysis and Proposed System.

# CHAPTER THREE

# EXISTING SYSTEM ANALYSIS & PROPOSED SYSTEM

## 3.1 Requirement Analysis

Requirement analysis is a software engineering task that bridges the gap between system engineering and software design. It allows the software engineer to refine the software allocation and builds the models of the data, functional and behavioral domains that will be treated by software. Among the entire software process models that have been discussed in the chapter 2, the waterfall process model has been chosen for this particular project because of its relatively simple structure. Another feature of this model is that it is easy to follow. For this particular project, the requirement analysis was done through continuous developer-client meetings. The solution or modification needed is designed and the modules that have been designed are the outcome of the requirement analysis and engineering.

## 3.2 Existing System Analysis

As we told in chapter-1 that we have chosen an Android based mobile application to collect gas meter reading with appropriate customer's approval. Gas distribution companies in our country, depends heavily on manual process for calculation of bills and writing meter reading in book. Manual procedure have some manual faults in bill generation process as well the process is time consuming. This procedure makes a lot of burden on meter reader and customer, as well create inconvenience for the customer to lodge any complaints. With advancement of technology things are becoming simpler and easier for us. In that software has three modes: User mode, Admin mode and meter readers mid user mode. In user mode visitors are allowed to see the entire website only, they are not permitted to add, delete or modify something. In admin mode user allow for all kind of permission, they are authorized for everything's. In mid user mode, users are permitted for some specific things. They are limited user. The existing system analysis is the first step for requirement analysis. **We have studied many utility companies billing system. But we found that no one organization using any Android based billing system and storing billing information in database system.**

## 3.3 The Proposed System

The proposed system means a new system that should have the ability to overcome the problems of existing one. After analyzing the problem of existing system, the following modules for the proposed system have been established. The Proposed system means a new system that should have the ability to overcome the problems of existing manual procedure. After analyzing the problem of existing system, the following rules & regulation for the proposed system have been established-

1. To develop the mechanism of easy and trustworthy and transparent billing system.
2. To develop a Android based software system that will help a company for a consolidated billing system for meter based customers.
3. Customer approval is necessary and will be implemented through proper valuation.
4. To develop the methodology of changing the policy for Billing System in conjunction with the changed scenario of the banks where the deposit have already been made and finally.
5. To develop a Billing System when emergency occurs.

### 3.3.1 Modules of the Proposed System

Modularity of software helps better management during development stages and later in maintenance. In order to do so, *"Mobile Application To Collect Gas Meter Reading"* project has been broken down into several modules. Each module has subdivided into functions.

### 3.3.2 User Groups of the Proposed System

Two types of user groups have been proposed for this project. These are Administrative user & Normal user.

- Administrative user
    - Add new user and give authority to normal users.
    - Add, modify, delete and all of their relevant information.
    - Generate all status of Gas Billing
    - Check all kinds of information.
- Normal User
    - Add meter reading data

- Checking own billing Status
- Finding present offer, notification and email

The following are the major operations in this application.

- Registration of User
- Forgot Password
- Login
- Change password
- Add Meter Reading
- Update Meter Reading
- Billing Status

## 3.4 Summary

In the gas industry of Bangladesh, gas distribution and revenue collection system is composed of interrelated and interdependent subsystems. Organizational culture and subcultures are important determinant of how people use information and information systems. The increasing popularity of internet and rapid digitalization is the driving force of proposed system. Thischapter has discussed about the Existing System Analysis and Proposed System. Next chapter will discuss about the Project Estimation.

# CHAPTER FOUR

# SYSTEM DESIGN & TOOLS

## System Design

System design is the most creative and challenging phase in software development. System design describes the final system and the process by which it is developed. It refers to the technical specifications that can be applied to implement the new system.

## 4.1 Analysis Modeling

To accomplish analysis model objectives we had to consider some basic schema such as – data flow diagram (DFD), entity relationship diagram (ERD) and normalized scheme (NS).

### 4.1.1 Data Flow Diagram (DFD)

Data flow diagram is a picture of the movement of data between external entities and the process and data stores within a system. DFD helps the analyst to understand the entire system and helps in preparing the coding phase. There are two standards for drawing DFD:

- The DeMarco&Yourdan Symbol set
- Gane&Sarson DFD Symbol set

The Gane&Sarson Symbol set have been used to draw the DFD for our project.These are:

The preparation of DFDs can go upto several levels deep. In each level the breakdown of a process of the previous level is shown in details. The DFDs for this project goes upto Level 1, starting from the context level.

### 4.1.2 Entity Relationship Diagram (ERD) of Proposed Android System:

A detailed, logical representation of the entities, associations, and elements for an organization or a business area is known as E-R model and the graphical representation of E-R model known as entity relationship diagram. It will be use for designing the database. It will make a clear understanding of the database for the system.

**Cardinality:**

The number of instances of one entity that can be associated with each instances of other entity.

The cardinality ratios are:

1. One to One: ( | …. | )
2. One to Many: ( | …. < )
3. Many to Many: ( > ….. < )

**Primary Key (PK)**

A primary key is an attribute or a collection of attributes that allow us to identify an entity uniquely.

**Foreign Key (FK)**

A foreign key is an attribute of a relation which refers to an exciting attribute of another relation.

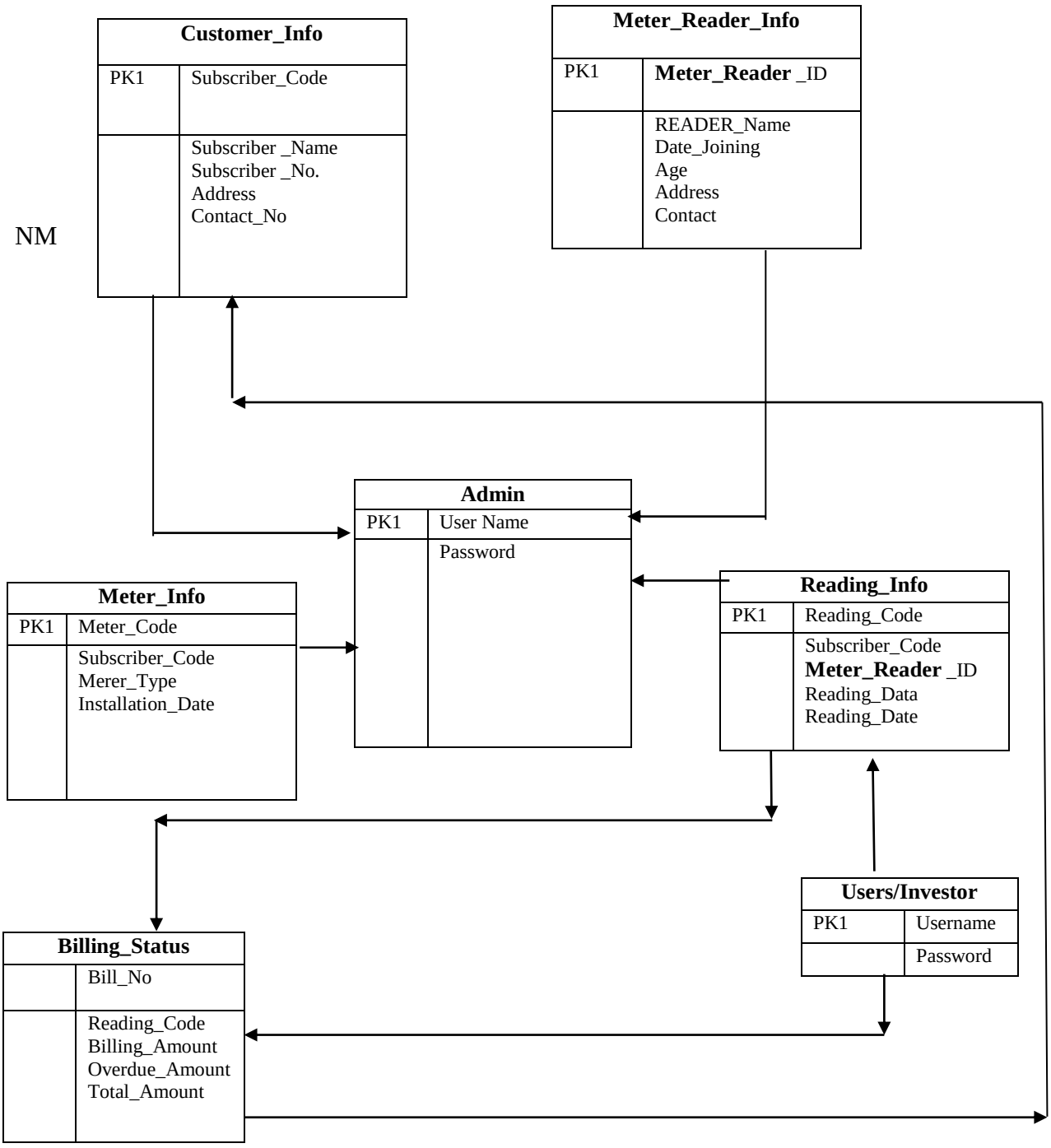



Fig 4.1: E-R Diagram of Android Based Metering System

By taking all the entities from ERD following tables have been designed for the database.

Table 4.1: Admin Information

**ADMIN**

| SL No | Name | Data Type | Width | Constrain |
|---|---|---|---|---|
| 01. | USERNAME | VARCHAR | 15 | PRIMARY KEY |
| 02. | PASSWORD | VARCHAR | 15 | |

Table 4.2: Admin Create Subscriber Info

**ADMIN CREATE SUBSCRIBER INFO**

| SL No | Name | Data Type | Width | Constrain |
|---|---|---|---|---|
| 01. | Subscriber_Code | VARCHAR | 15 | PRIMARY KEY |
| 02. | Subscriber _Name | VARCHAR | 50 | |
| 03. | ADDRESS | VARCHAR | 50 | |
| 04. | Contact_No. | VARCHAR | 11 | |

Table 4.3: Admin Create Meter Reader Info

**ADMIN CREATE METER READER INFO**

| SL No | Name | Data Type | Width | Constrain |
|---|---|---|---|---|
| 01. | READER_Name | VARCHAR | 50 | PRIMARY KEY |
| 02. | Date_Joining | VARCHAR | 15 | |
| 03 | Age | VARCHAR | 15 | |
| 04 | Address | VARCHAR | 50 | |
| 05 | Contact | VARCHAR | 15 | |

Table 4.4: Admin Create Reading Info

**ADMIN CREATE READING INFO**

| SL No | Name | Data Type | Width | Constrain |
|---|---|---|---|---|
| 01. | Reading_Code | VARCHAR | 15 | PRIMARY KEY |
| 02. | Subscriber_Code | VARCHAR | 15 | |
| 03. | **Meter_Reader** _ID | VARCHAR | 50 | |
| 04. | Reading_Data | VARCHAR | 50 | |
| 05. | Reading_Date | VARCHAR | 15 | |

Table 4.5: Admin Create Billing Info

**ADMIN CREATE BILLING INFO**

| SL No | Name | Data Type | Width | Constrain |
|---|---|---|---|---|
| 01. | BILL_NO | VARCHAR | 15 | PRIMARY KEY |
| 02. | READING_CODE | VARCHAR | 15 | |
| 03. | BILL_AMOUNT | VARCHAR | 15 | |
| 04. | OVERDUE_AMOUNT | VARCHAR | 50 | |

## 4.2 Process Model

The Process Model shows the overall functionality of the system. **Data Flow Diagrams** is the tools for process modeling. The Data Flow Diagram shows the sequence of events of a business operation.

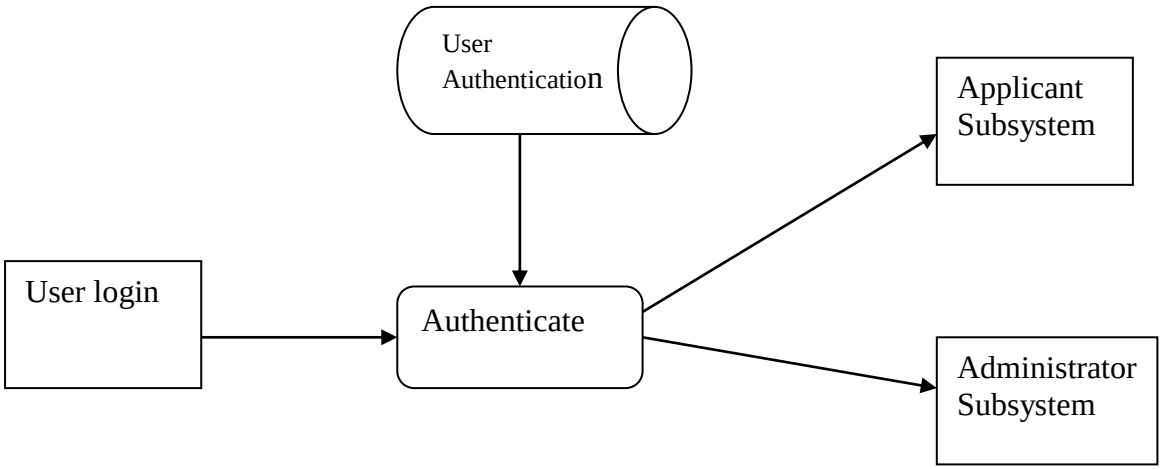



**Fig 4.2: User Authentication**

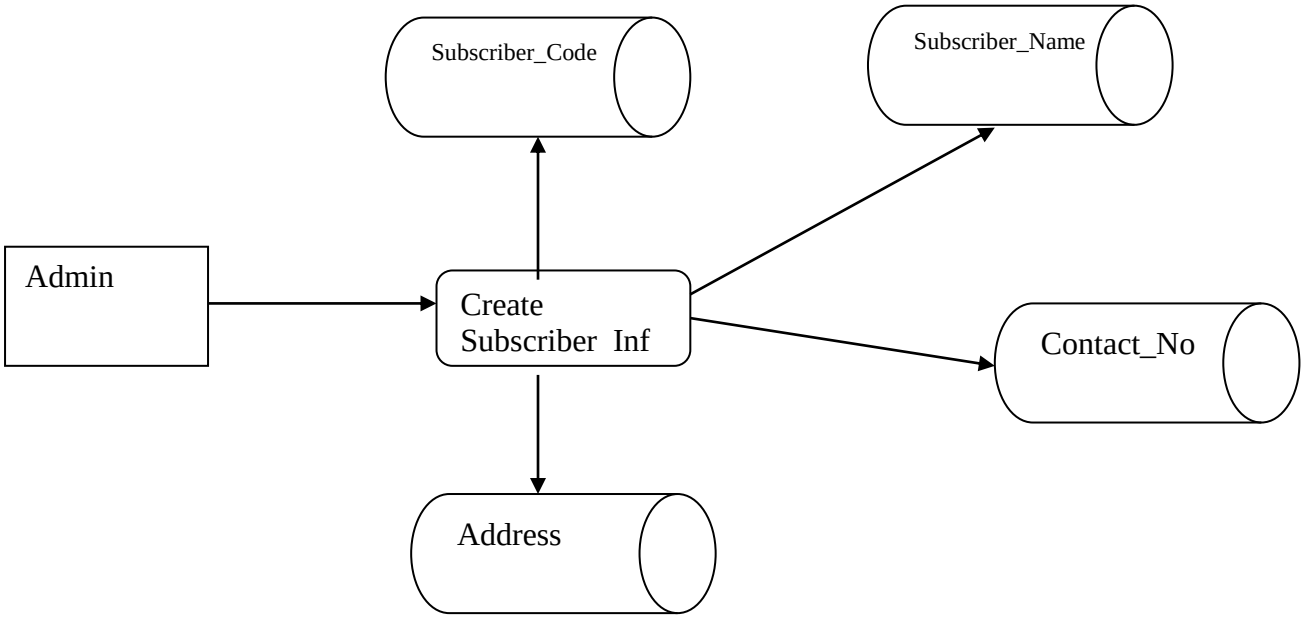



**Fig 4.3: Admin Create Subscriber_Info DFD**

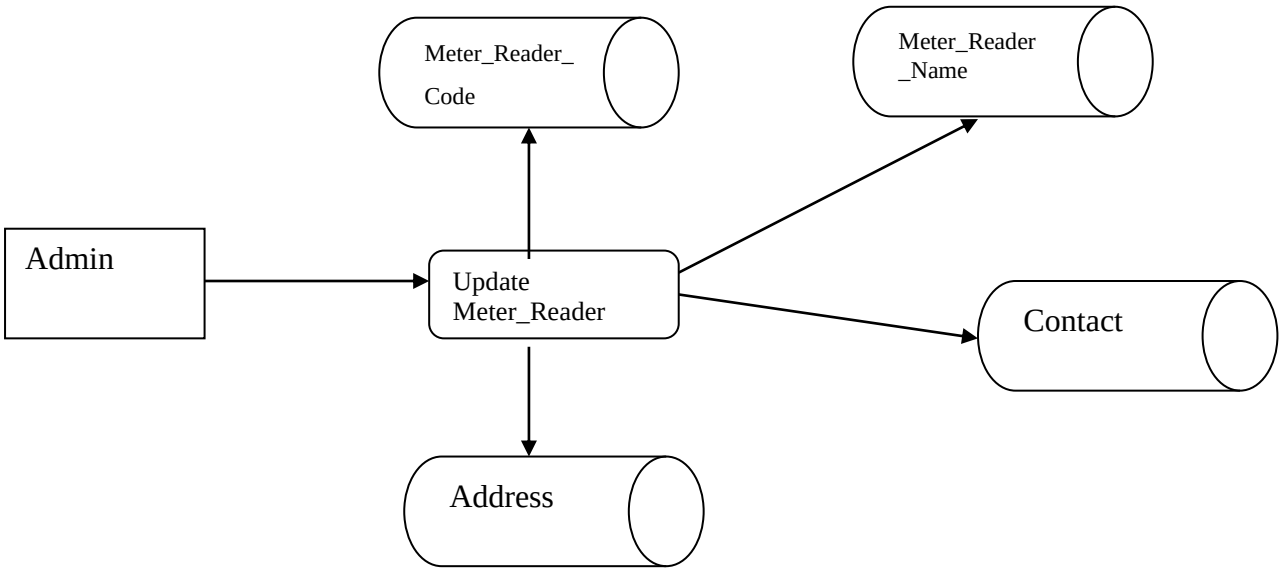



**Fig 4.4: Admin Update Meter_Reader_Info DFD**

### 4.3 UML Diagram

Object oriented analysis and design are implemented during the software design. Different software tools are used for designing different part of the software. UML is used for high level design of the proposed system. Different diagrams are drowning using MS vision. These diagrams help in visualizing the whole development process.

The Unified Modeling Language (UML) is a standard language for specifying, visualizing, constructing, and documenting the artefact of software systems, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

#### a) Use Case Diagram

A use case is a set of scenarios that describing an interaction between a user and a system. A use case diagram displays the relationship among actors and use cases. The two main components of a use case diagram are use cases and actors.

An actor is represents a user or another system that will interact with the system you are modeling. A use case is an external view of the system that represents some action the user might perform in order to complete a task.

In the bellow figures show the use cases for the user, the paid user and admin.

### 4.3.1. Use case diagram of Admin User

Fig: shows the use case diagram of Admin. Admin user can login their account for Create new information. Admin can Update and Delete the Information.

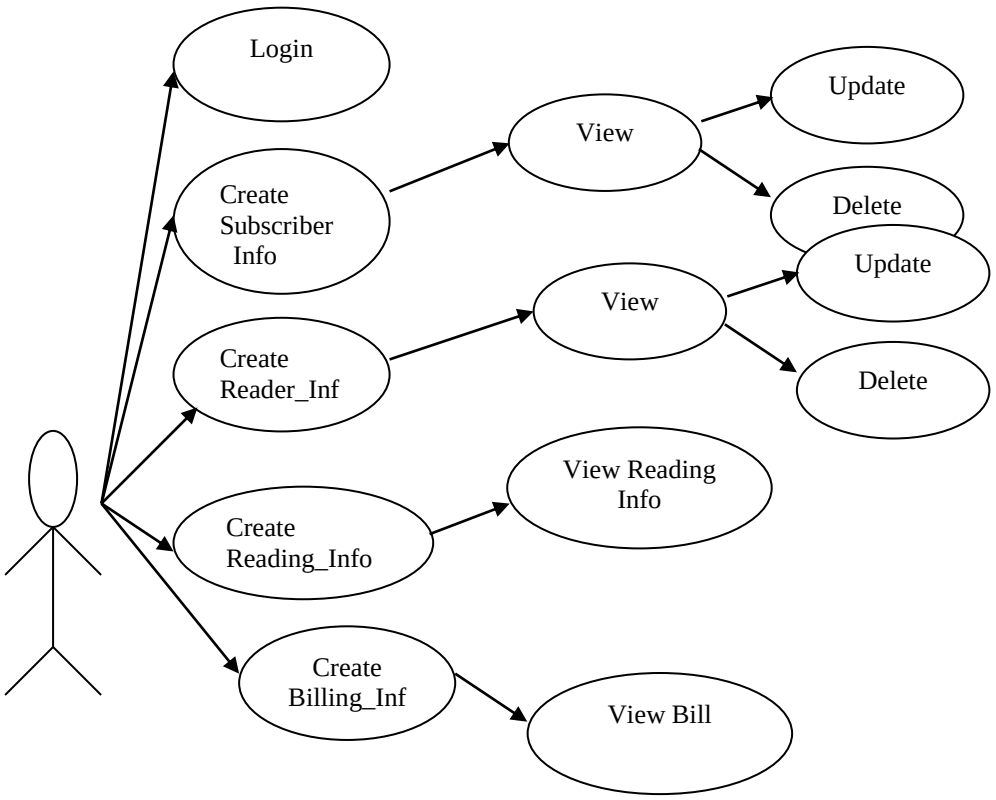



**Fig 4.5: Use case diagram of Admin**

### 4.3.2 . Use case diagram of Reader

Figure shows the use case diagram of Reader. Reader can upload meter reading and update and modify after log in his account.

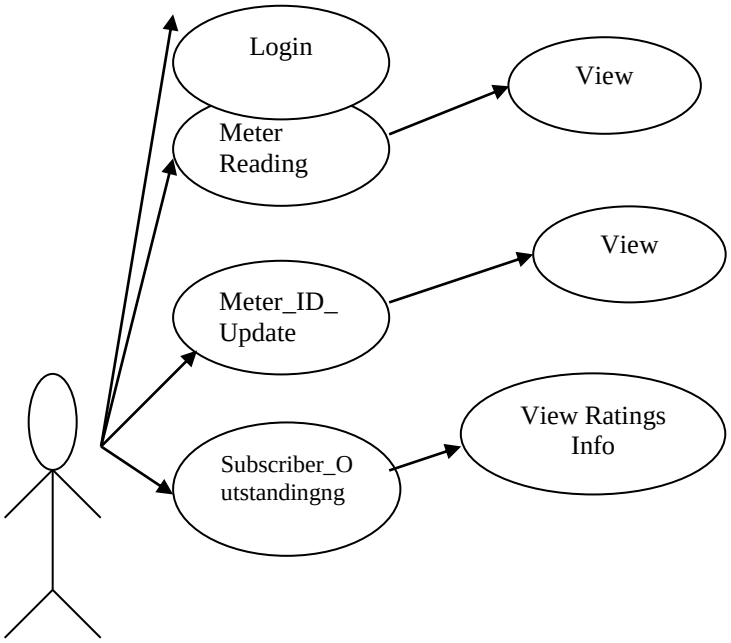



Fig 4.6: Use case diagram of Subscriber

## 4.3.3 Normalized Schema (NS)

Normalization is a step-by-step decomposition of complex record into simple record. Normalization reduces redundancy using the principles of non-loss decomposition. Non-loss decomposition is the reduction of a table to smaller tables without loss of information.

Redundancy is the unnecessary repetition of data. It causes problems with storage and retrieval of data. Redundancy can lead to, Inconsistencies, because errors are more likely to occur when data are repeated.

Update anomalies because of inserting, modifying and deleting data may cause inconsistencies. A fully normalized record consists of a primary key that identifies an entity a set of attributes that describe the entity.

There are few rules for database normalization. Each rule is called normalization. If the first rule is observed, the database is said to be first normal form (1NF). Similarly, if the second rule is observed, the database is said to be second normal form and if the third rule is observed, the database is said to be third normal form.

First Normal Form (1NF)

The purpose of the first normal (1NF) form is to eliminate repeating group of attributes of an entity. Remedy is to create a new relation for each repeating group and create primary key for new relation.

Second Normal Form (2NF)

The purpose of the second normal form (2NF) is to eliminate partial key dependencies for relations where primary key composed of more than one attributes, no non-key attributes should be functionally dependent on a part of the primary key. Remedy is to create a new relation for attributes which are not dependent on the whole key, copy the part of the primary key into the new relation that has link with the new attributes and create primary key for new relation.

Third Normal Form (3NF)

The purpose of the third normal form (3NF) is to eliminate interdependencies between non-key attributes. Relations should not have a non-key attribute functionally determined by another non key attribute.

## 4.4 Design Tools

The system is expected to serve all the purpose that it has been developed for. The design tools are used for designing the front end and back end interface and also generating the relevant report. We use the following tools for developing our project.

- Android Studio
- Node JS source code
- Cordova Platform
- Apache ANT
- Phonegap

- Bootstrap
- HTML,JAVASCRIPT, CSS (Front-end and Design)
- PHP and MYSQL ( for back end database design)
- Dream Waver , XAMPP server (Application Software)

### 4.4.1Android Studio

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as primary IDE for native Android application development.

The following features are provided in the current stable version:

- Gradle-based build support.
- Android-specific refactoring and quick fixes
- Lint tools to catch performance, usability, version compatibility and other problems
- ProGuard integration and app-signing capabilities
- Template-based wizards to create common Android designs and components
- A rich layout editor that allows users to drag-and-drop UI components, option to preview layouts on multiple screen configurations.
- Support for building Android Wear apps
- Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine
- Android Virtual Device (Emulator) to run and debug apps in the Android studio.
- Android Studio supports all the same programming languages of IntelliJ, and PyCharm e.g. Python, and Kotlin; and Android Studio 3.0 supports "Java 7 language features and a subset of Java 8 language features that vary by platform version." External projects backport some Java 9 features.

### 4.4.3 Node JS source code

Node.js is an open-source, cross-platform JavaScript run-time environment that executes JavaScript code server-side. Historically, JavaScript was used primarily for client-side scripting, in which scripts written in JavaScript are embedded in a webpage's HTML and run client-side by a JavaScript engine in the user's web browser. Node.js lets developers use JavaScript for server-side scripting—running scripts server-side to produce dynamic web

page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web application development around a single programming language, rather than different languages for server side and client side scripts.

Though .js is the conventional filename extension for JavaScript code, the name "Node.js" does not refer to a particular file in this context and is merely the name of the product. Node.js has an event-driven architecture capable of asynchronous I/O. These design choices aim to optimize throughput and scalability in web applications with many input/output operations, as well as for real-time Web applications (e.g., real-time communication programs and browser games).

The Node.js distributed development project, governed by the Node.js Foundation, is facilitated by the Linux Foundation's Collaborative Projects program.

Corporate users of Node.js software include GoDaddy, Groupon, IBM, LinkedIn, Microsoft, Netflix, PayPal, Rakuten, SAP, Tuenti, Voxer, Walmart and Yahoo!.

Overview of Node JS:

Node.js allows the creation of Web servers and networking tools using JavaScript and a collection of "modules" that handle various core functionality. Modules are provided for file system I/O, networking (DNS, HTTP, TCP, TLS/SSL, or UDP), binary data (buffers), cryptography functions, data streams, and other core functions. Node.js's modules use an API designed to reduce the complexity of writing server applications.

Node.js applications can run on Linux, macOS, Microsoft Windows, NonStop, and Unix servers. Alternatively, they can be written with CoffeeScript (a JavaScript alternative), Dart or TypeScript (strongly typed forms of JavaScript), or any other language that can compile to JavaScript.

Node.js is primarily used to build network programs such as Web servers. The biggest difference between Node.js and PHP is that most functions in PHP block until completion (commands execute only after previous commands finish), while Node.js functions are non-

blocking (commands execute concurrently or even in parallel, and use callbacks to signal completion or failure).

Platform Architecture:

Node.js brings event-driven programming to web servers, enabling development of fast web servers in JavaScript. Developers can create highly scalable servers without using threading, by using a simplified model of event-driven programming that uses callbacks to signal the completion of a task. Node.js connects the ease of a scripting language (JavaScript) with the power of Unix network programming.

Node.js was built on the Google V8 JavaScript engine since it was open-sourced under the BSD license, extremely fast, and proficient with internet fundamentals such as HTTP, DNS, TCP. Also, JavaScript was a well-known language, making Node.js immediately accessible to the entire web development community.

### 4.4.3Cordova Platform

Apache Cordova (formerly PhoneGap) is a mobile application development framework originally created by Nitobi. Adobe Systems purchased Nitobi in 2011, rebranded it as PhoneGap, and later released an open source version of the software called Apache Cordova. Apache Cordova enables software programmers to build applications for mobile devices using CSS3, HTML5, and JavaScript instead of relying on platform-specific APIs like those in Android, iOS, or Windows Phone. It enables wrapping up of CSS, HTML, and JavaScript code depending upon the platform of the device. It extends the features of HTML and JavaScript to work with the device. The resulting applications are hybrid, meaning that they are neither truly native mobile application (because all layout rendering is done via Web views instead of the platform's native UI framework) nor purely Web-based (because they are not just Web apps, but are packaged as apps for distribution and have access to native device APIs). Mixing native and hybrid code snippets has been possible since version 1.9.

The software was previously called just "PhoneGap", then "Apache Callback". As open-source software, Apache Cordova allows wrappers around it, such as Appery.io or Intel XDK.

PhoneGap is Adobe’s commercial version of Cordova along with its associated ecosystem. Many other tools and frameworks are also built on top of Cordova, including Ionic, Monaca,

TACO, Onsen UI, Visual Studio, GapDebug, App Builder, Cocoon, Framework7, Quasar Framework, Evothings Studio, NSB/AppStudio, Mobiscroll, the Intel XDK and the Telerik Platform. These tools use Cordova, and not PhoneGap for their core tools.

Contributors to the Apache Cordova project include Adobe, BlackBerry, Google, IBM, Intel, Microsoft, Mozilla, and others.

Design and rationale:

The core of Apache Cordova applications use CSS3 and HTML5 for their rendering and JavaScript for their logic. HTML5 provides access to underlying hardware such as the accelerometer, camera, and GPS. However, browsers' support for HTML5-based device access is not consistent across mobile browsers, particularly older versions of Android. To overcome these limitations, Apache Cordova embeds the HTML5 code inside a native WebView on the device, using a foreign function interface to access the native resources of it.

Apache Cordova can be extended with native plug-ins, allowing developers to add more functionalities that can be called from JavaScript, making it communicate directly between the native layer and the HTML5 page. These plugins allow access to the device's accelerometer, camera, compass, file system, microphone, and more.

However, the use of Web-based technologies leads some Apache Cordova applications to run slower than native applications with similar functionality. Adobe Systems warns that applications built with Apache Cordova may be rejected by Apple for being too slow or not feeling "native" enough (having appearance and functionality consistent with what users have come to expect on the platform).

Supported platforms:

Apache Cordova currently supports development for the operating systems Apple iOS, Bada, BlackBerry, Firefox OS, Google Android, LG webOS, Microsoft Windows Phone (7 and 8), Nokia Symbian OS, Tizen (SDK 2.x), and Ubuntu Touch. The table below is a list of supported features for each operating system.

### 4.4.4Apache ANT

Apache Ant is a Java library and command-line tool whose mission is to drive processes described in build files as targets and extension points dependent upon each other. The main known usage of ANT is the build of Java applications. ANT supplies a number of built-in tasks allowing to compile, assemble, test and run Java applications. ANT can also be used effectively to build non Java applications, for instance C or C++ applications. More generally, Ant can be used to pilot any type of process which can be described in terms of targets and tasks.

Ant is written in Java. Users of Ant can develop their own "antlibs" containing ANT tasks and types, and are offered a large number of ready-made commercial or open-source "antlibs".

Ant is extremely flexible and does not impose coding conventions or directory layouts to the Java projects which adopt it as a build tool.

Software development projects looking for a solution combining build tool and dependency management can use Ant in combination with Apache Ivy.The Apache Ant project is part of the Apache Software Foundation.

### 4.4.5Phonegap

Native mobile applications involved a costly and time consuming process. But this is no longer the case with Cordova developed applications. With Cordova, we use html, jquery, javascript and css to prototype and compile your app into a native mobile application, with full native capabilities.

### 4.4.6Bootstrap

Build responsive, mobile-first projects on the web with the world's most popular front-end component library.

Bootstrap is an open source toolkit for developing with HTML, CSS, and JS. Quickly prototype your ideas or build your entire app with our Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful plugins built on jQuery.

### 4.4.7 HTML, JAVASCRIPT & CSS

HTML, which stands for Hyper Text Markup Language, is the predominant markup language for web pages. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists etc as well as for links, quotes, and other items. It allows images and objects to be embedded and can be used to create interactive forms. It is written in the form of HTML elements consisting of "tags" surrounded by angle brackets within the web page content. It can include or can load scripts in languages such as JavaScript which affect the behavior of HTML processors like Web browsers; and Cascading Style Sheets (CSS) to define the appearance and layout of text and other material.

CSS: Cascading Style Sheets (CSS) is a style sheet language used to describe the presentation semantics (that is, the look and formatting) of a document written in a markup language. Its most common application is to style web pages written in HTML and XHTML, but the language can be applied to any kind of XML document, including SVG and XUL. CSS is designed primarily to enable the separation of document content (written in HTML or a similar markup language) from document presentation, including elements such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content (such as by allowing for table less web design). CSS can also allow the same markup page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or screen reader) and on Braille-based, tactile devices. While the author of a document typically links that document to a CSS style sheet, readers can use a different style sheet, perhaps one on their own computer, to override the one the author has specified. CSS specifies a priority scheme to determine which style rules apply if more than one rule matches against a particular element. In this so-called

*cascade*, priorities or *weights* are calculated and assigned to rules, so that the results are predictable.

JAVASCRIPT: JavaScript is an object-oriented scripting language used to enable programmatic access to objects within both the client application and other applications. It is primarily used in the form of client-side JavaScript, implemented as an integrated component of the web browser, allowing the development of enhanced user interfaces and dynamic websites. JavaScript is a dialect of the ECMAScript standard and is characterized as a dynamic, weakly typed, prototype-based language with first-class functions. JavaScript was influenced by many languages and was designed to look like Java, but to be easier for non-programmers to work with.

### 4.4.8 MYSQL & PHP

Hypertext Preprocessor, is a widely used, general-purpose scripting language that was originally designed for web development, to produce dynamic web pages. It can be embedded into HTML and generally runs on a web server, which needs to be configured to process PHP codeand create web page content from it. It can be deployed on most web servers and on almost every operating system and platform free of charge.PHP is installed on over 20 million websites and 1 million web servers. PHP was originally created by RasmusLerdorf in 1995 and has been in continuous development ever since. The main implementation of PHP is now produced by The PHP Group and serves as the *de facto* standard for PHP as there is no formal specification.PHP is free software released under the PHP License, which is incompatible with the GNU General Public License (GPL) because of restrictions on the use of the term *PHP*.PHP has evolved to include a command line interface capability and can also be used in standalone graphical applications.

MYSQL: MySQL is a relational database management system (RDBMS) which has more than 6 million installations. MySQL stands for "My Structured Query Language". The program runs as a server providing multi-user access to a number of databases. The project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL is owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now a subsidiary of Sun

Microsystems, As of 2009 Oracle Corporation began the process of acquiring Sun Microsystems; Oracle holds the copyright to most of the MySQL codebase

MySQL is often used in free software projects which require a full-featured database management system, such as Word Press, phpBB and other software built on the LAMP software stack. It is also used in very high-scale World Wide Web products including Wikipedia, Google and Facebook.

Advantage of SQL Server:

- Easy to use
- Familiar to the user

### 4.4.9 Dream Waver &Xampp Server

Adobe Dreamweaver (formerly Macromedia Dreamweaver) is a web development application originally created by Macromedia, and is now developed by Adobe Systems, which acquired Macromedia in 2005.Dreamweaver is available for both Mac and Windows operating systems. Recent versions have incorporated support for web technologies such as CSS, JavaScript, and various server-side scripting languages and frameworks including ASP, ColdFusion, and PHP.

XAMPP server: XAMPP is an easy to install Apache distribution containing MySQL, PHP and Perl. XAMPP is really very easy to install and to use – just download, extract and start.

## 4.5 Summary

This chapter has discussed about the System Design & tools. Next chapter will discuss the Implementation & Testing.

# CHAPTER FIVE

# IMPLEMENTATION & TESTING

In implementation and testing stage Total quality management through six sigma has been organized. Quality standards are tried to be maintained through feedback from consumer as well responsible person assigned for revenue purpose. Performance has been evaluated. In the preliminary stage database has been implemented with SQL. For mass data storage system will be upgraded through Oracle. With the high data speed this mobile application is very suitable for any android operating system.

## 5.1 Welcome Page

The login page is very simple. A Meter Reader through using User Name and Password can use this Android application. Training for user is very crucial as well vital. Android mobiles may be new for the meter reader as well consumer. There are some small tactics for which mobile, accommodated with android technology and proper twining, is very important. Figure 5.1 shows the login page.

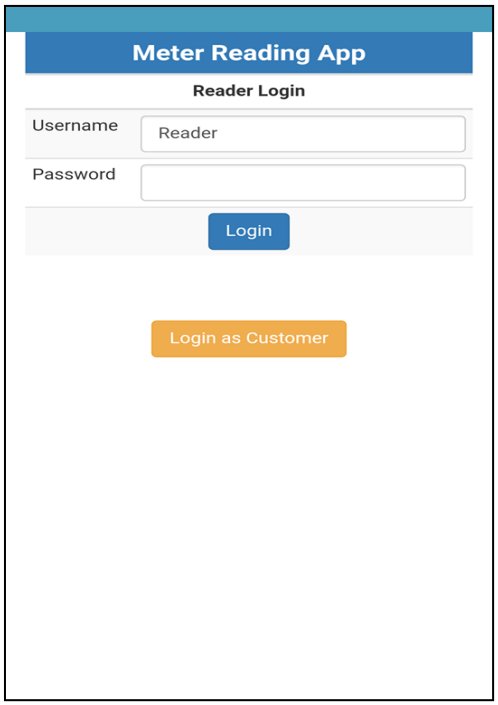



**Fig 5.1: Login Page**

Vendors himself, system analyst, external as well in house trainer are required for effective uses.  Upon meter reading taken by meter reader, for transparency and greater accountability, meter reading will be approved by customer. Customers will approve meter reading through a button given inside the same apps.

## 5.2 Available Task:

After login by the meter reader, a task window will be opened. In this window, there are two options for meter reader like uploading meter reading data, changing password and logout option. Figure 5.2 shows the task page. Strategies for converting from old system to the new one are incorporated in the design. As well this application will be filtered in google play store so that it is only available for users residing in Bangladesh.This app is compatible with all screen sizes and densities, because the system makes the appropriate adjustments to UI layout and image resources as necessary for each screen.

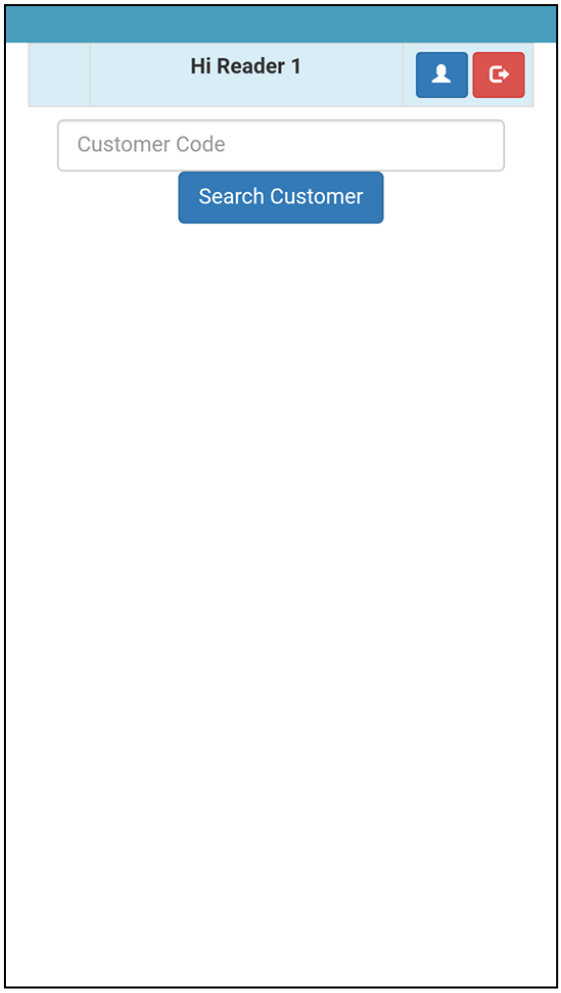



**Fig 5.2: Available Task Page**

## 5.3 Customer Info Form

This page contains customer code. Each customer has an individual customer code. Through inputting this code, we can enter each customer's concerned screen and can subsequently take snapshot of meter reading which will ultimately upload data upon concern of customer upon verbal confirmation of the customer. While designing, a systematic and useful form has been designed. This form is very easy for any preliminary educated person. Proper flow of data has been kept in mind. While inputting data for database storage, decreasing use of paper has been considered. Easy going and less instruction is required for this information fill up. Figure 5.3 shows the customer information page.

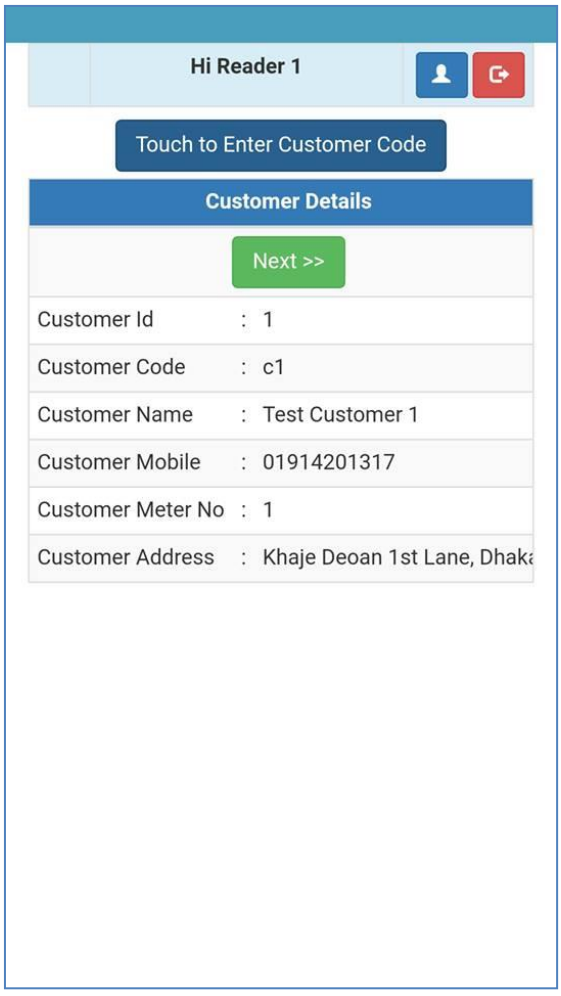



**Fig 5.3: Customer Info Form**

## 5.4 Capture Image Window:

The User Homepage contains unit data input option and customer PIN. In this page, image is uploaded and reading data is recorded. Image quality is ensured and mpeg format is defined. Src command has been defined via option of taking a photo directly with the camera, not choosing an existing image file. This camera cannot handle multiple images being selected/pasted/dropped at once. Option to stop using the camera when one no longer needs it has been applied. Not only will this save battery and processing power, it will also give users confidence in application. Figure 5.1 shows the meter reader's Camera App page.

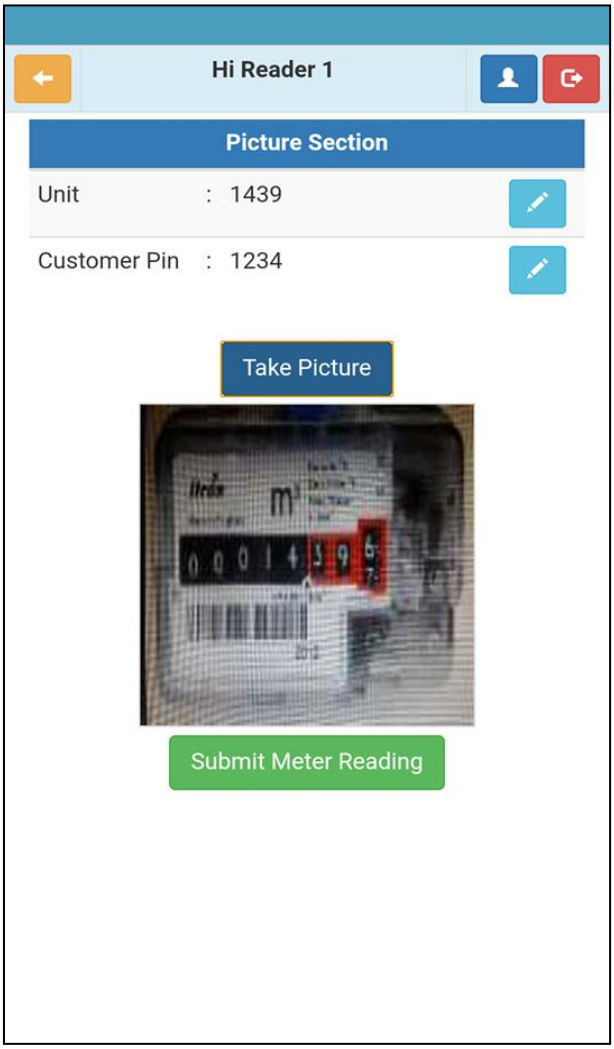



**Fig 5.4: Capture Image Window**

## 5.5 Uploading Image Window:

This page contains uploading image of Meter Reading. The FileUpload control is used to send files from the client to the web server, and saw how to present this binary data in a data Web control. The uploaded image file will be stored with the designated server directory. This directory is directory writable. This database is supported with any types of browsers and can be handled. The maximum file size that the server allows is 2MB. The user is not allowed to upload the same image more than once. Figure 5.5 shows the uploading of image.

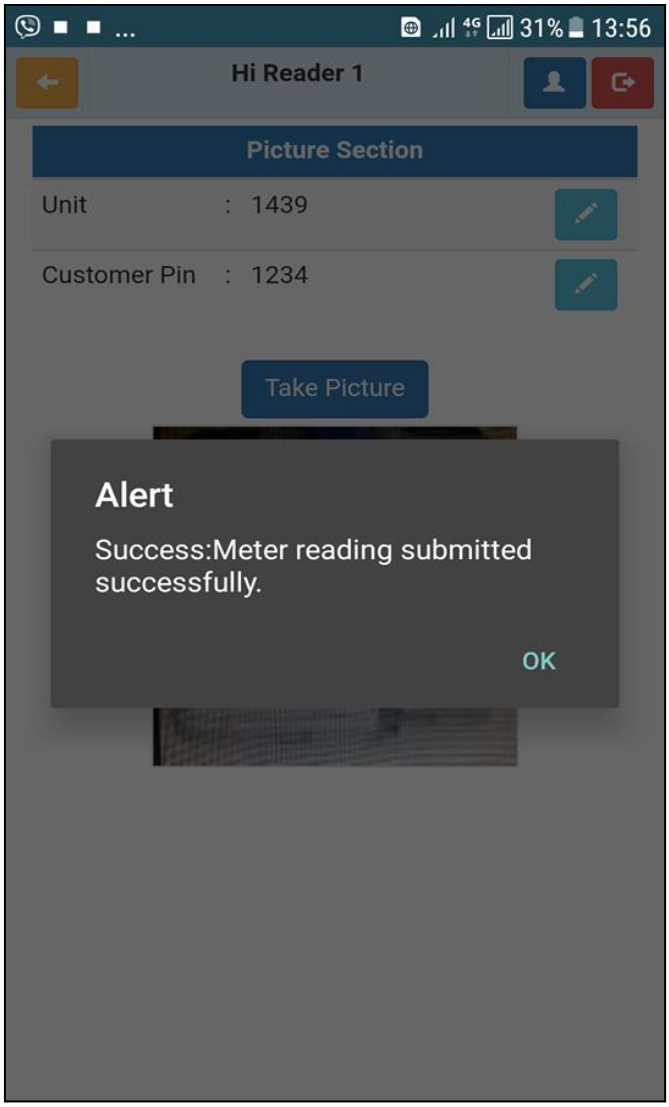



**Fig 5.5: Uploading of Image**

## 5.6 Meter Reading Approval Control

This screen in the apps is User's approval Control panel. Any user can pass through this screen through his respective user name and password. Data integrity is imposed within a database when it is designed and is authenticated through the ongoing use of error checking and validation by the customer through "Load Meter Reading" button. Figure 5.6 shows the Administration Panel.

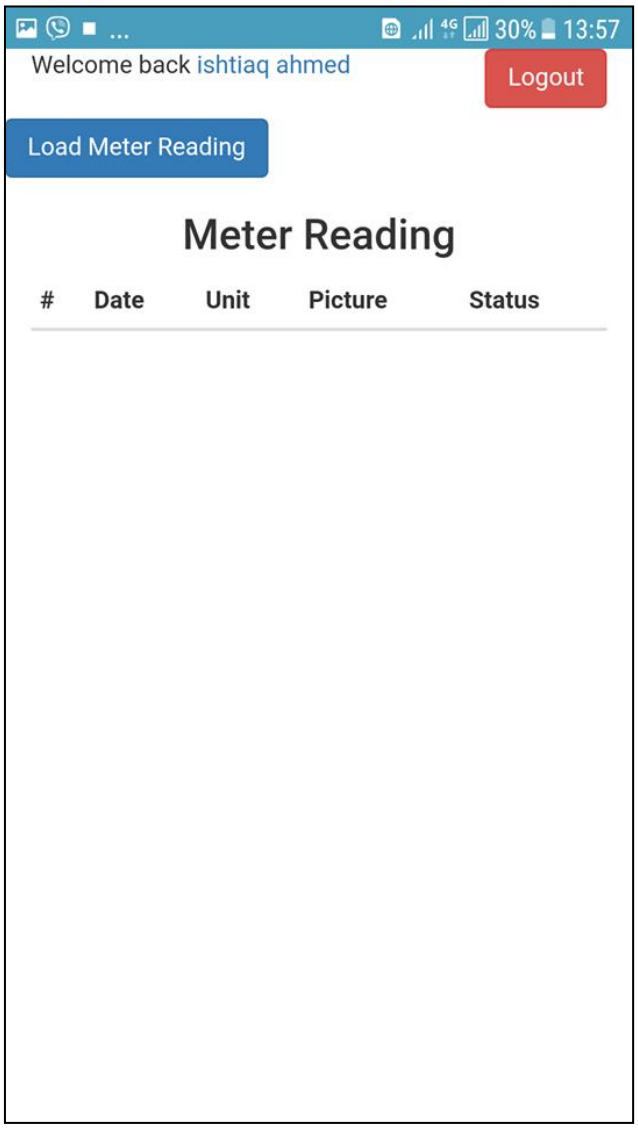



**Fig 5.6: Subscriber ApprovalScreen**

When load meter reading data has been activated then pending data screen come to screen.

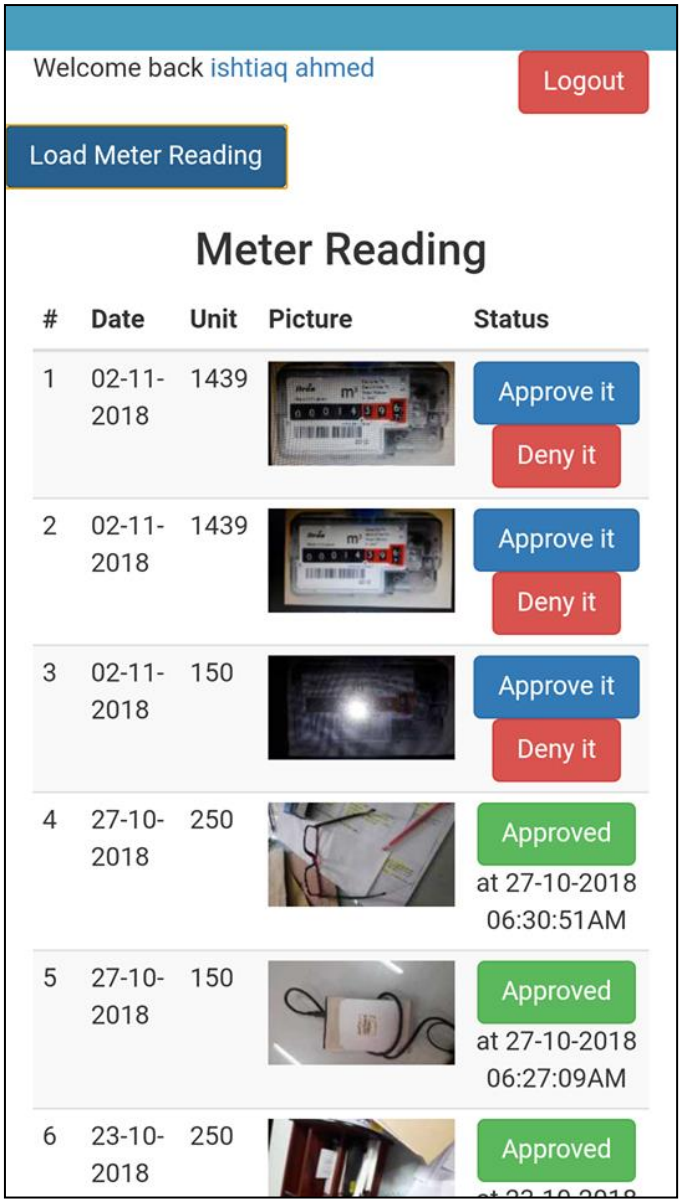



**Fig 5.7: ApprovalInterface Window for subscriber**

Users can connect to the database without authenticating a login at the Database Engine level through installed apps. Isolating the database from the Database Engine makes it possible to easily move the database to another instance of SQL Server. Including all the database settings in the database enables database owners to manage all the configuration settings for the database.After approving data, when meter reading is approved by subscriber, then meter reading is uploaded at the database.

## 5.7 Admin Login

This page is made for security purpose. Any admin can change his/her password from this page. Client/Server model has been implemented for remote server hosted in World Wide Web. Interaction of client with server through numeric and alphabetic data as well image data has been concerned prior to database design. it is a client based application as if any time server become down, then application will be on run ones it is recovered. On database part, customer –computer interface has been developed based on desk top computer. Data transmission is adopted to any type 2G, 3G or 4G technology, WiFi, Bluetooth technology. Figure 5.7 shows the Admin Login page.

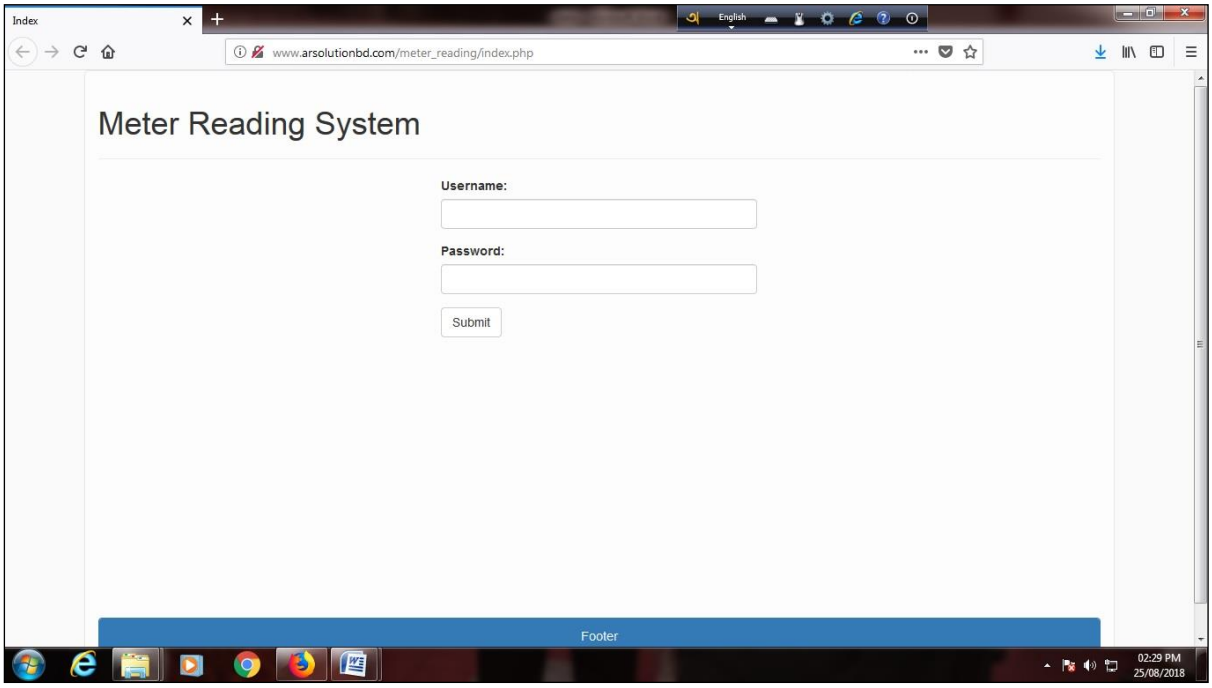



**Fig 5.8: Admin Login**

This screen in the apps is User's approval Control panel. Any admin can search the user from this page through name or email. Data integrity is a fundamental component of information security. In its broadest use, "data integrity" refers to the accuracy and consistency of data stored in a database, data warehouse, data mart or other construct. All characteristics of the data must be correct – including business rules, relations, dates, definitions and lineage – for data to be complete. Data integrity is imposed within a database when it is designed and is authenticated through the ongoing use of error checking and validation routines. Figure 5.9 shows the Administration Panel.

## 5.8 Admin Panel

This page is Admin Panel. Any admin can edit/delete required information. Figure 5.8 shows the Administration Panel. Firewalls, gateways and virtual private networks to prevent hackers from gaining backdoor access to the database will be incorporated to the server. Intrusion detection products (Symantec Intruder Alert) may be introduced to suggest any action required.

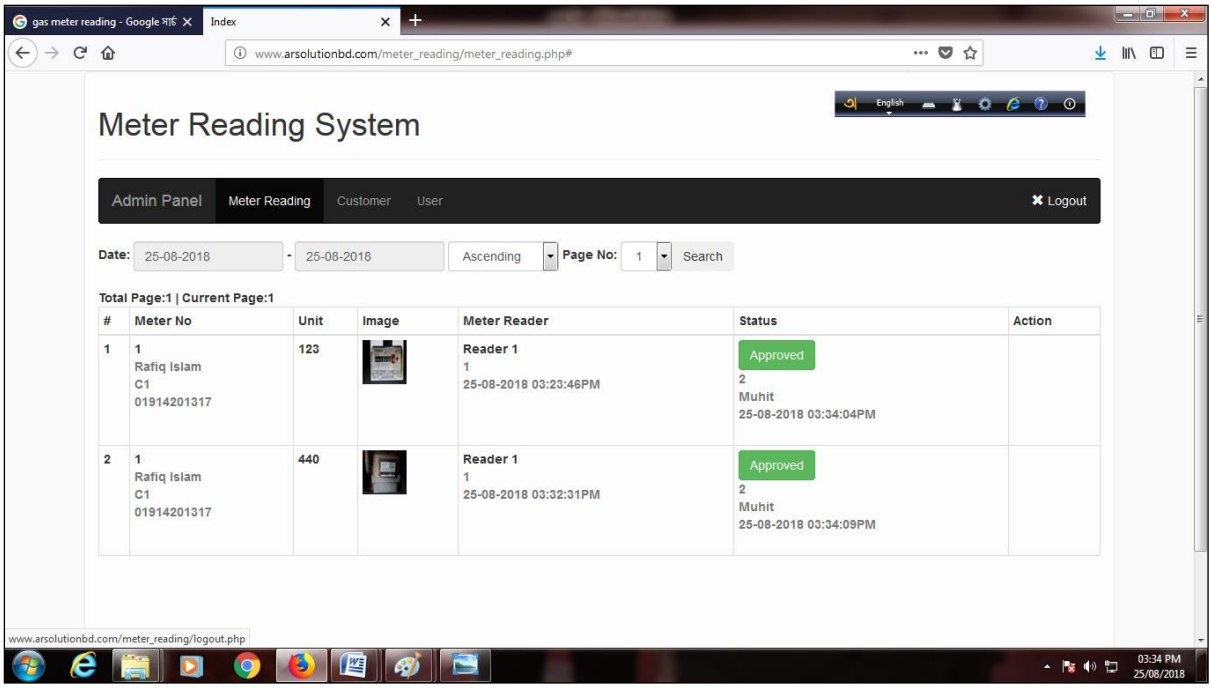



**Fig 5.9: Admin Panel**

## 5.9 Meter Reader Interface Panel

This page is Database management system (DBMS) for meter reader User Control. Any Admin can search the user from this page through name or email. Database management system (DBMS) is system software for creating and managing databases. This portion provides users and programmers with a systematic way to create, retrieve, update and manage data. It makes it possible for end users to create, read, update and delete data in a database.

Figure 5.10 shows the page for subscriber/meter reader info control.

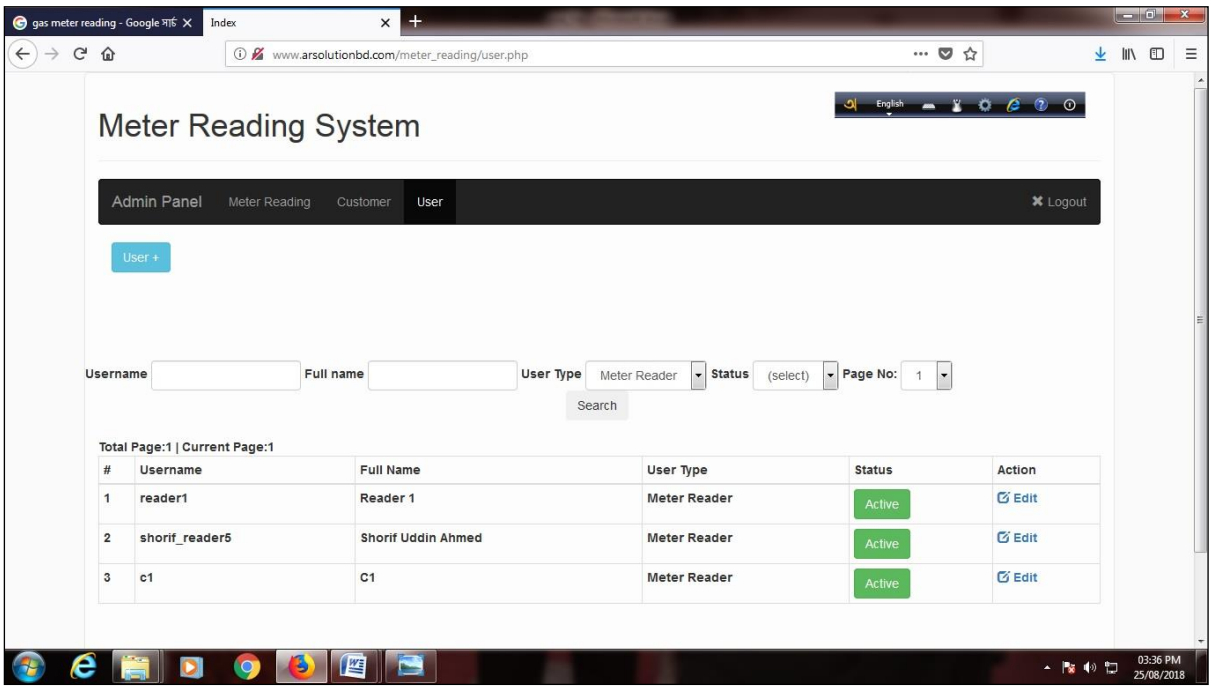



**Fig 5.10: Subscriber/ Meter Reader Info Control**

It can limit what data the end user sees, as well as how that end user can view the data, providing many views of a single database schema.That means it can protect users and applications from needing to know where data is stored or having to be concerned about changes to the physical structure of data (storage and hardware). As long as programs use the application programming interface (API) for the database that is provided by the DBMS, developers won't have to modify programs just because changes have been made to the database.

### 5.10 Subscriber Interface:

User interface page is General User Control. Any admin can search the subscriber from this page through name or email. Admin can add any user or delete any user through this interface. Admin can change any data sent through this window.  Figure 5.9 shows the User Interface Panel.

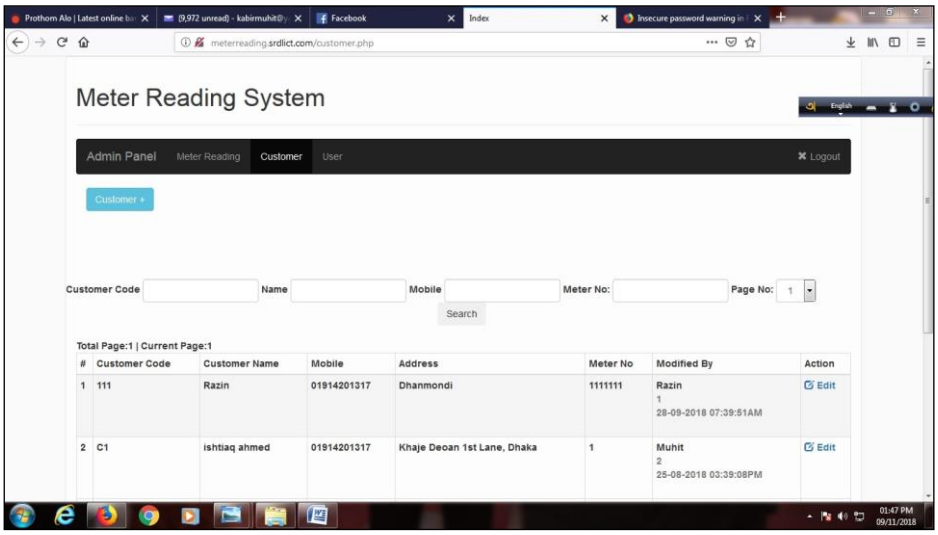



Fig: 5.11: Subscriber Interface Panel

# CHAPTER SIX

# CONCLUSION

## 6.1 Conclusion:

Development of an Android based mobile application to collect gas meter reading with appropriate customer's approval has been complete using Cordova platform. This application has an apk file, database interface and backend database. This apk file will be installed in meter reader Android set through which he will capture meter reading and upload reading online immediately through customer's confirmation with specific customer pin. This Android application is unique in its work and objectives of digitalization of meter reading are highlighted. There are some obvious limitations of the work which could not be completed within the timeframe are pointed out under this article.

In this project Software Development Life Cycle followed as a methodology. A full featured Android Based Meter Reading System has been developed. Most of the basic as well as advanced features are developed in this application.

This project work may be implemented in practical field for postpaid billing system as well to conduct energy audit door to door. This project is very much suited with Vision of Bangladesh Government of Digital Bangladesh in energy, gas and utility sector. In conclusion, both the achievement and limitation of our project work have been discussed below

### 6.1.1 Achievements

An overview of the existing gas distribution system of Bangladesh as well as the prevailing metering systems and data management systems are discussed in this project. In this connection, some problems of the present data management system used by the utility organizations are pointed out and there remedy has been designed to be solved through engagement of information technology.

The developed application is developed with vision of implementation in commercial purpose. Different types of advanced features like Meter reading database, uploading and updating billing database, verification and updating billing up to date information, add, update, delete, search billing info, keeping records of billing information are incorporated in

this application. Administrator of the software can control the user information, meter reader data and billing information etc. The developed software is user friendly. The features of the software are self-descriptive so that any new user can easily use this software. The developed software can easily be implemented for commercial purpose.

Finally the performance analysis of the proposed design has been carried out and described in the report. The analysis is made in terms cost, data volume and interoperability and have been investigated.

The rules and regulations followed in the gas meter reading system are summarized in this project. While discussing this issue, the difference between the conventional metering system and Android based metering system are depicted, framework is drown and implementation has been surfaced. This application has a common database for both pre-paid and postpaid system is designed. Common input and output for the two different prepaid system are identified and made compatible for this application.

**6.1.2 Limitation:** This project work has been developed as a prototype Android apk in limited scale. This application has a database which has limited data volume capacity as well as limited server capacity. This system cannot handle huge data traffic load. Besides this system has no report format version. As a result this database cannot generate bills automatically. Another limitation is, OMR capacity for database has not been incorporated for which database cannot automatically update data reading from image. Dispatch system of outstanding month end billing system through email system has not been outlined due to server's limited capacity. These all have been consciously limited in testing phase of designed Android application. Moreover, the suitability of the suggestions with respect to issues related to software and database cannot be verified due to the absence of real implementation, time limitation, uncertainty and limited logistic support.

## 6.2 Future Works

This project has been developed in a limited scale as proto type basis. In the implementation phase many scenarios will have to be developed with greater capacity having larger coding with database of greater load bearing scope. Future works related to the present development will have to be carried out in implementation phase. Few areas of improvement are addressed below for future reference and research in case of implementation phase:

Firstly, a billing system will have to be developed as a report software which will have a direct linkage to the central database and customer will have scope to see his respective repayment schedule and billing update.

Secondly, An optical mark capacity may have been enclosed with database, so that billing system may have optical mark capacity by which after uploading image, database may extract billing data from image and automatically update database checking authenticity of image.
Billing system.

Thirdly, the designed generalized database have to be upgraded to oracle platform from MySQL database to make it compatible for thousands of customers billing system.

Fourthly, A prototype software module and report generation module should be developed for testing the efficacy of the overall new disintegrated system.

Fifthly, A proper distribution of billing through email system have to be developed through PDF version of billing and effective distribution of billing via email protocol system.

## REFERENCES

[1] M. F. Khan, A. Zoha, and R. L. Ali, "Design and Implementation of Smart Billing and Automated Meter Reading System for Utility Gas", International Conference on Information and Emerging Technologies, pp. 1-6, 2007.

[2] T. Meek, and P. Chilese, "Remote Meter Reading by Radio - A European Perspective" Schlumberger Industries, UK. pp. 196-198.

[3] B. S. Koay, S. S. Cheah, Y. H. Sng, P. H. J. Chong, P. Shum, and Y. C. Tong, "Design and Implementation of Bluetooth Energy Meter", International Conference on Information, Communications and Signal Processing, vol.3, no., pp.1474-1477, 2003.

[4] Y. Liu, R. A. Fischer, and N. N. Schulz, "Distribution System Outage and Restoration Analysis Using A Wireless AMR System", Power Engineering Society Winter Meeting, IEEE, vol. 2, pp. 871-875, 2002.

[5] M. I. H. B. Asad, M. L. Ali, and M. S. Islam, "Development of a novel prepaid gas metering system", International Conference on Electrical and Computer Engineering (ICECE), pp. 77 – 80, 2014.

[6] S. K. Ali Zaidi, H. Masroor, S. R. Ashraf, and A. Hassan, "Design and Implementation of Low Cost Electronic Prepaid Energy Meter", IEEE International Multitopic Conference, pp. 548-552, 2008.

[7] Chao-Tung Yang, Wei-Sheng Chen, Kuan-Lung Huang, Jung-Chun Liu, Wen-Hung Hsu, and Ching-Hsien Hsu "Implementation of Smart Power Management and Service System on Cloud Computing", International Conference on Ubiquitous Intelligence & Computing and International Conference on Autonomic & Trusted Computing (UIC/ATC), pp. 924 – 929, 2012.

[8] M. S. Parvin, and S. M. L. Kabir, "Standard and interoperable database for pre-paid electricity metering systems in Bangladesh", International Conference on Informatics, Electronics & Vision (ICIEV), pp. 356 – 361, 2012.