**CHAPTER 1**

**1. INTRODUCTION**

* 1. **GENERAL INTRODUCTION**

The construction industry has been criticized as an "old-school" industry, because of being a slow adopter of mobile technologies. Due to the rapid growth in the smartphone applications market, new applications become available every day for use in different industries. Given the large number of choices, both companies and individuals in the construction industry must beware when selecting and purchasing smartphone applications. The business needs of the potential users and the expectations from the applications must be well identified, and the selection must be made accordingly. Properly selecting and deploying smartphone applications for construction-related tasks is expected to improve communication, enhance workflow with real time information, and increase productivity. Our software is an innovative management solution created for the construction . The system can mirror your company's structure and optimize the entire workflow, giving you full control over your business.

Construction professionals use tablets and smartphones to increase job-site efficiency. Both tablets and smartphones combine mobile accessibility with mobile construction management applications, with the added advantage for smartphones that they can fit in a pocket and provide the same benefits. The main advantage gained from the use of such mobile devices is that they enable construction professionals to work interactively and dynamically; data collected from the site is shared in real time among the project participants with visual attachments, and site reports are generated with more accurate and up-to-date information. It is apparent that construction companies need to go mobile in order to stay competitive in the industry. However, it is essential to establish a mobile device strategy before selecting a mobile device and applications. Since carrying multiple devices is not ideal for construction professionals, the selected applications must be compatible for the device used for work purposes. Different mobile devices with different operating systems limit the choice of applications.Thus a cross platform based mobile application development will be more compatible and preferable.

Currently, construction information exchange is done through traditional information and communication methods that are non-automatic and paper-based. However, having construction information digitalized by using a mobile computing technology that runs automatic information management activities is ideal and much desired.Even though the use of mobile applications is gaining popularity in the construction industry, its adoption is a vital process that must be conducted properly. The study showed that there are five elements that affect adoption: the user, the organization, the technology, the project, and the environment. A successful implementation of new technologies requires training and participation of users. Top management executives’ involvement in new technologies facilitates the adoption process. The technology adopted by organizations must be easy to use, compatible, reliable, and provide information security. Project type, cost, duration, specifications, and location must also be considered in selecting and adopting technologies. The environment affects the use of mobile technologies indirectly.In recent years, construction companies have become aware of the potential in using mobile applications. Some companies are developing their own mobile applications. The reason behind this shift is that a large amount of information is produced during construction, but this information is not managed efficiently

* 1. **GOAL OF PROJECT**

The overall goal of **Data Acquisition in Construction Sites with Remote Monitoring** is to provide , work monitoring in construction sites. The company performing various works at various location and each location may be farway. Each site may be handled by a different site enginner, the site enginner visits daily to the construction site inorder to verify and check the daily work progress.this app can be used by the site engineer to update the daily work updates of the site.the updates provided by those site engineer can be accesed by the clients, the main goal of the project is to provide a platform which can be accessible by the users and get proper and valid information about their sites, the site engineer can update the details of daily work progress and share images of the work, this may lead good communication between the client and the customers.

**CHAPTER 2**

**2. LITERATURE SURVEY**

**2.1 STUDY OF SIMILAR WORK**

The **Data Acquisition in Construction Sites with Remote Monitoring** is an android application which is helpful for the making customer relation smoother.this application will helps to exchange of information from construction company to the clients about their ongoing project. In existing system all the activities are done manually. It is very much costly and time consuming. The existing the method followed is in a traditional way or by sending messages or email over a network, a proper communication is not followed by any of the companies.In our proposed system there is so many advanced feathure which can over come those issues.The old way of manual work are being changed by the proposed system.The important factor is that anyone having smartphone with valid login credentilas can use in any platform .

**2.1.1 EXISTING SYSTEM**

Currently, construction information exchange is done through traditional information and communication methods that are non-automatic and paper-based. Social media application like whatsapp are used by site engineers for image and video sharing .Normally communications are via those applications .Most of the companies are following such manner inorder to exchange of information or data to the customer.the traditional way exchanging of information can lead to mis-communication between the customers and the companies. Some of the companies are providing web applications to know the facility and infrastructure of the construction company,but those web applications are just for their business purpose.those applications can be used by any user on the internet and can acces the details of the construction company. For an individual who is giving a project to a company cannot view the updates through websites,some of them having the final project result in the website to boost the company. You cannot get closer to the client and build a trusting relationship.All those things are in a traditional way of communication

**2.1.2 DRAWBACKS OF EXISTING SYSTEM**

* monitoring the entire system is a tedious task.
* It is very time consuming to manage the necessary activity schedule of various operation and maintenance works details.
* It is difficult to calculate or keep track of the amount of work done, make records, and retrieve previous records.
* Data and records are not organized
* Since third party applications are data loss and security issues may occur
* No location based services are there so cant track site engineer activities
* No daily updates are reports available
* Co ordination with entire team and transfer of data -diagrams ,plans etc are difficult
* Searching or sorting or finding data on day basis or name bases is difficult

**CHAPTER 3**

**3. OVERALL DESCRIPTION**

**3.1 PROPOSED SYSTEM**

Most of the information and communication related activities in the construction industry are time consuming when conducted manually. It is stated that it takes time to obtain an organized and complete report of visual inspection of damaged structures because engineers capture photos and gather information on site individually, and because the detection and quantification of the damage is measured through cumbersome manual approaches. Thus, system focus on mobile application to improve the current practice in team-based visual inspection of civil structures. A smartphone provides a constant feed of information among people that interact with each other on a daily basis. Using a smartphone for work purposes is a decision that many contractors are considering today.One of the potential benefits is that a construction manager can monitor job-site productivity, and take necessary actions to maximize efficiency. Built-in cameras allow project participants to exchange site data with photos and videos which, in some cases, are much more detailed and informing than a conversation. Real time job site data with visual attachments can even increase a contractor’s credibility with the owner. With a variety of selections, construction companies and professionals can find commercially available applications that can assist with day-to-day operations, increase productivity, credibility, and cost efficiency

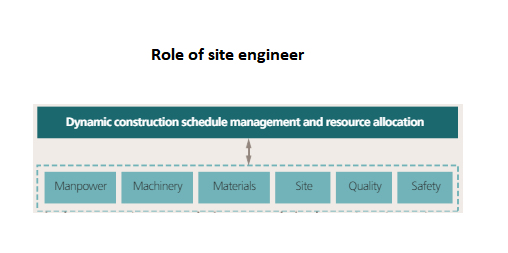
Thus, the proposed system is an Android Based Mobile Application to Monitor Works at Remote Sites”. It has been proposed for the betterment of the construction company. By developing this application, the Construction Company can easily record their progress of various works and their day-to-day expenditures that are made at various sites. Also, the system integrates Ip cameras placed in the constitution site. Thus, the construction company will get all the updates of construction without any time delay. By providing mobile applications for Site engineers, and owners(clients)the system makes easier ways to complete critical workflows, these technologies can significantly reduce delays, improve quality, and increase profits.

Objectives of the system includes:

* One app for all platforms – Android, iOS operating system, mobile or tablet interfaces
* Real-time backend integration – directly into your organization's core
* Built-in Security and User Management
* Use smart device capabilities, including GPS, camera activation and barcode scanning within the app.

**3.2 FEATURES OF PROPOSED SYSTEM**

Site engineer Mobile Application:



* Site engineer can login with credentials
* Can view construction site and details
* Can view diagrams and information
* Can update daily work reports
* Can upload images and videos regarding work
* Can upload human resource details
* Can upload materials details
* Can view milestones and to do works
* Can get notifications
* Can view client queries
* Can view ip camera
* checking technical designs and drawings to ensure that they are followed correctly
* supervising contracted staff
* ensuring project packages meet agreed specifications, budgets and/or timescales
* providing technical advice and solving problems on site
* Location based updates are marked on daily report so that company can track site engineer efficiency
* Locate diff site in google map,get map view for travelling
* Details from site engineer are uploaded to server via web service.These details can be used in the web application for AI based perdition of work efficiency
* Can communicate with team members
* Can contact client and can give updates to client queries
* Can get machinery and tools details in the site
* Can upload petty expenses and bill details
* Integrated Bar code scanner to scan different barcodes in machines,equipment's etc
* can view safety features and worker details
* Can update issues and changes
* Can save details offline in case of internet or sever issues

**3.3 FUNCTIONS OF PROPOSED SYSTEM**

* Enhancement: The main objective of Data Acquisition in Construction Sites with Remote Monitoring is to enhance and upgrade the existing system by increasing its efficiency and effectiveness. The software improves the working methods by replacing the existing manual system with new feathures.
* Accuracy: The Data Acquisition in Construction Sites with Remote Monitoring provides quick response & very accurate information regarding the Onsite progress etc. Any details or system in an accurate manner, as and when required.
* User-Friendly: The Data Acquisition in Construction Sites with Remote Monitoring p has a very user-friendly interface. Thus, the users will feel very easy to work on it. The Application provides accuracy along with a pleasant interface. Make the present manual system more interactive, speedy and user friendly.
* Availability: The data which uploaded can be be retried as and when required.Thus, there is no delay in the availability of any information, whatever needed, can be obtained very quickly and easily.

**3.4 REQUIREMENTS SPECIFICATION**

A software requirements specification (SRS) is a detailed description of a software system to be developed with its functional and non-functional requirements. The SRS is developed based the agreement between customer and contractors. It may include the use cases of how user is going to interact with software system. The software requirement specification document consistent of all necessary requirements required for project development. To develop the software system we should have clear understanding of Software system. To achieve this we need to continuous communication with customers to gather all requirements.

A good SRS defines the how Software System will interact with all internal modules, hardware, communication with other programs and human user interactions with wide range of real life scenarios. Using the *Software requirements specification* (SRS) document on QA lead, managers creates test plan. It is very important that testers must be cleared with every detail specified in this document in order to avoid faults in test cases and its expected results.

It is highly recommended to review or test SRS documents before start writing test cases and making any plan for testing. Let’s see how to test SRS and the important point to keep in mind while testing it.

**3.5 FEASIBILITY ANALYSIS / STUDY**

The main aim of the feasibility study activity is to determine. Whether it would be financially and technically feasible to develop the product. The feasibility study activity involves analysis of the problem and collection of all relevant information relating to the product such as the different data items which would be input to the system the processing required to be carried out of these data, the output data required to be carried out of these data, the output data required to be produced by the system, as well as various constraints on the behaviour of the system.

In our application we would find the actual requirements of this software and add that features Such as monitoring, protocoring etc. For adding this feature, we will like take different ways to solving this last find the best way to complete these features.

Feasibility studies aim to objectively and rationally uncover the strengths and weakness of the existing business or proposed venture, opportunities and threats as presented by the environment, the resources required to carry through, and ultimately the prospects for success. In its simplest term, the two criteria to judge feasibility are cost required and value to be attained As such, a well-designed feasibility study should provide a historical background of the business or project, description of the product or vice, accounting statements, details of the operations and management, marketing research and policies, financial data, legal requirements and tax obligations. Generally, studies precede technical development and project implementation.

The feasibility study to be conducted for this project involves

**3.5.1 TECHNICAL FEASIBILITY**

The technical Feasibility depends on the technical aspects of the proposed system. The main consideration is to be given at the study of available resources of the organizations where the project is to be developed and implemented. Here the system analyst evaluates the technical merits of the given system emphasis on the performance, reliability, maintainability and productivity. In our project technical feasibility is implemented in such a way that the required resources and its availability was successfully studied and applied.

**3.5.2 OPERATIONAL FEASIBILITY**

Operational analysis is the most frequently used method for evaluating the effectiveness of a new system. More commonly known as cost/benefit analysis, the procedure is to determine the benefits and saving that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system. An entrepreneur must accurately weigh the cost versus benefits before taking an action. Cost-based study: It is important to identify cost and benefit factors, which can be categorized as follows:

1. Development costs.

2. Operating costs.

This is an analysis of the costs to be incurred in the system and benefits derivable out of the system. Time-based study: This is an analysis of the time required to achieve a return on investments the future value of a project is also a factor. The system is operationally Feasible

**3.5.3 ECONOMICAL FEASIBILITY**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

In case of new project, financial viability can be judged on the following parameters:

• Total estimated cost of the project

• Financing of the project in terms of its capital structure, debt equity ratio and promoter ‘s share of total cost

• Existing investment by the promoter in any other business

Projected cash flow and profitability.

The system is Economically Feasible

**3.5.4 BEHAVIOURAL FEASIBILITY**

The behavioral feasibility depends upon whether the system performed in the expected way or not.Feasibility study is a test of system proposal according to its workability, impact on organization ability to meet the user’s need and efficient use of resources. However, a feasibility study provides a useful starting point for full analysis. Our project checks whether the system is performed in the expected way or not. For this we have given inputs for checking whether the expected outputs where generated. Feasibility study is a test of system proposal according to its workability, impact on organization ability to meet the user’s need and efficient use of resources. However, a feasibility study provides a useful starting point for full analysis.

**CHAPTER 4**

**4. OPERATING ENVIRONMENT**

**4.1 HARDWARE REQUIREMENT**

Processor : Intel i5 8th Gen

RAM : **8** GB DDR4

Hard Disk : **512** GB SSD

Display Size : Compatible Size(Recommend 15’inch)

Screen Resolution : 1920\*1080 Pixels

Keyboard : Wireless Enabled Keyboard(Recommend :Logitech)

Keyboard Mouse : Wireless Enabled Mouse (Recommend :Logitech)

MONITOR : LED Monitor

Dedicated Graphics Card : Nvidia Geforce GTX 1050 4GB DDR5

**4.2 SOFTWARE REQUIREMENT**

Operating System : Windows (7/8/10)/Ubuntu (14/16/18/20)

Programming Language : Dart, Java

IDE : Android studio 4.1

Front-End : Flutter SDK, Android SDK,

Back-End : Mysql, Sqlite

**4.3 TOOLS AND PLATFORMS**

**4.3.1 DART:**

DART is an open-source general-purpose programming language. It is originally developed by Google and later approved as a standard by ECMA. Dart is a new programming language meant for the server as well as the browser. Introduced by Google, the Dart SDK ships with its compiler – the Dart VM. The SDK also includes a utility -dart2js, a transpiler that generates JavaScript equivalent of a Dart Script. This tutorial provides a basic level understanding of the Dart programming language.

**Features of DART**

* Open Source-Dart is an open-source programming language, which means it is freely available. It is developed by Google, approved by the ECMA standard, and comes with a BSD license.

### Platform Independent-Dart supports all primary operating systems such as [Windows](https://www.javatpoint.com/windows), [Linux](https://www.javatpoint.com/linux-tutorial), Macintosh, etc. The Dart has its own Virtual Machine which known as Dart VM, that allows us to run the Dart code in every operating system.

### Object-Oriented-Dart is an object-oriented programming language and supports all oops concepts such as classes, inheritance, interfaces and optional typing features. It also supports advance concepts like mixin, abstract, classes, reified generic, and robust type system.

* Concurrency-Dart is an asynchronous programming language, which means it supports multithreading using Isolates. The isolates are the independent entities that are related to threads but don't share memory and establish the communication between the processes by the message passing. The message should be serialized to make effective communication. The serialization of the message is done by using a snapshot that is generated by the given object and then transmits to another isolate for desterilizing.

### Extensive Libraries-Dart consists of many useful inbuilt libraries including DK (Software Development Kit), core, [math](https://www.javatpoint.com/math), async, math, convert, [html](https://www.javatpoint.com/html-tutorial), IO, etc. It also provides the facility to organize the Dart code into libraries with proper namespacing. It can reuse by the import statement.

### Easy to learn-As we discussed in the previous section, learning the Dart is not the Hercules task as we know that Dart's syntax is similar to [Java](https://www.javatpoint.com/java-tutorial), [C#](https://www.javatpoint.com/c-sharp-tutorial), [JavaScript](https://www.javatpoint.com/javascript-tutorial), [kotlin](https://www.javatpoint.com/kotlin-tutorial), etc. if you know any of these languages then you can learn easily the Dart.

### Flexible Compilation-Dart provides the flexibility to compile the code and fast as well. It supports two types of compilation processes, AOT (Ahead of Time) and JIT (Just-in-Time). The Dart code is transmitted in the other language that can run in the modern web-browers.

### Type Safe-The Dart is the type safe language, which means it uses both static type checking and runtime checks to confirm that a variable's value always matches the variable's static type, sometimes it known as the sound typing.

### Objects-The Dart treats everything as an object. The value which assigns to the variable is an object. The functions, numbers, and strings are also an object in Dart. All objects inherit from Object class.

### Browser Support-The Dart supports all modern web-browser. It comes with the dart2js compiler that converts the Dart code into optimized JavaScript code that is suitable for all type of web-browser.

### Community-Dart has a large community across the world. So if you face problem while coding then it is easy to find help. The dedicated developers' team is working towards enhancing its functionality.

**4.3.2 JAVA:**

JAVA is a general-purpose programming language that is class-based, object-oriented, and designed to have as few implementation dependencies as possible. Java applications are typically compiled to byte code that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. & J2EE applications are made up of components such as Java Server Pages (JSP), Java servlets, and Enterprise JavaBeans (EJB) modules. These components enable software developers to build large-scale, distributed applications. Developers package J2EE applications in Java Archive (JAR) files (similar to zip files), which can be distributed to production sites. Administrators install J2EE applications onto the Application Server by deploying J2EE JAR files onto one or more server instances (or clusters of instances).

**Features of Java**

* Object Oriented − In Java, everything is an Object. Java can be easily extended since it is based on the Object model.
* Platform Independent − Unlike many other programming languages including C and C++, when Java is compiled, it is not compiled into platform specific machine, rather into platform-independent bytecode. This byte code is distributed over the web and interpreted by the Virtual Machine (JVM) on whichever platform it is being run on.
* Simple − Java is designed to be easy to learn. If you understand the basic concept of OOP Java, it would be easy to master.
* Secure − With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.
* Architecture-neutral − Java compiler generates an architecture-neutral object file format, which makes the compiled code executable on many processors, with the presence of Java runtime system.
* Portable − Being architecture-neutral and having no implementation dependent aspects of the specification makes Java portable. The compiler in Java is written in ANSI C with a clean portability boundary, which is a POSIX subset.
* Robust − Java makes an effort to eliminate error-prone situations by emphasizing mainly on compile time error checking and runtime checking.
* Multithreaded − With Java multithreaded feature it is possible to write programs that can perform many tasks simultaneously. This design feature allows the developers to construct interactive applications that can run smoothly.
* Interpreted − Java byte code is translated on the fly to native machine instructions and is not stored anywhere. The development process is more rapid and analytical since the linking is an incremental and light-weight process.
* High Performance − With the use of Just-In-Time compilers, Java enables high performance.
* Distributed − Java is designed for the distributed environment of the internet.
* Dynamic − Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving environment. Java programs can carry an extensive amount of run-time information that can be used to verify and resolve accesses to objects at run-time.

**4.3.3 MYSQL:**

MySql Server is the world's most used relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases. This stores data in the form of multiple related tables. The SQL phrase stands for Structured Query Language. The MySql development 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 was owned and sponsored by a single for-profit firm, the Swedish company MySql AB, now owned by Oracle Corporation.

**4.3.4 SQLLITE:**

SQLite is a C-language library that implements a [small](https://www.sqlite.org/footprint.html), [fast](https://www.sqlite.org/fasterthanfs.html), [self-contained](https://www.sqlite.org/selfcontained.html), [high-reliability](https://www.sqlite.org/hirely.html), [full-featured](https://www.sqlite.org/fullsql.html), SQL database engine. SQLite is the [most used](https://www.sqlite.org/mostdeployed.html) database engine in the world. SQLite is built into all mobile phones and most computers and comes bundled inside countless other applications that people use every day.  The SQLite [file format](https://www.sqlite.org/fileformat2.html) is stable, cross-platform, and backwards compatible and the developers pledge to keep it that way [through the year 2050](https://www.sqlite.org/lts.html). SQLite database files are commonly used as containers to transfer rich content between systems  and as a long-term archival format for data  There are over 1 trillion (1e12) SQLite databases in active use .SQLite [source code](https://sqlite.org/src) is in the [public-domain](https://www.sqlite.org/copyright.html) and is free to everyone to use for any purpose.

**4.3.5 ANDROID SDK:**

SDK is a collection of tools which make easiness and help in app development. It is an essential part of the android application development.

**4.3.6 ANDROID STUDIO:**

Android Studio is an official IDE for Android Operating System of Google. It is specially built for Android Development and is based on IntelliJ IDEA software. Mostly Java is used but app can also be built using C++. It targets all size of screen devices like android smart phones, Tablets, Smart TVs and Wearable devices. For coding, there is a most featured editor and a layout designer. For the output, an Emulator is given which is also known as Android Virtual Device (AVD) which looks like real device

**4.3.6 FLUTTER SDK**

Flutter is an [open-source](https://en.wikipedia.org/wiki/Open-source_software) [UI](https://en.wikipedia.org/wiki/User_interface) [software development kit](https://en.wikipedia.org/wiki/Software_development_kit) created by [Google](https://en.wikipedia.org/wiki/Google). It is used to develop applications for [Android](https://en.wikipedia.org/wiki/Android_(operating_system)), [iOS](https://en.wikipedia.org/wiki/IOS), [Linux](https://en.wikipedia.org/wiki/Linux), [Mac](https://en.wikipedia.org/wiki/MacOS), [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows), [Google Fuchsia](https://en.wikipedia.org/wiki/Google_Fuchsia), and the web from a single [codebase](https://en.wikipedia.org/wiki/Codebase). Flutter apps are written in the [Dart](https://en.wikipedia.org/wiki/Dart_(programming_language)) language and make use of many of the language's more advanced features. On [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows), [macOS](https://en.wikipedia.org/wiki/MacOS), and [Linux](https://en.wikipedia.org/wiki/Linux) Flutter runs in the Dart virtual machine, which features a [just-in-time](https://en.wikipedia.org/wiki/Just-in-time_compilation) execution engine. While writing and debugging an app, Flutter uses Just In Time compilation, allowing for "hot reload", with which modifications to source files can be injected into a running application. Flutter extends this with support for [stateful](https://en.wikipedia.org/wiki/Stateful) hot reload, where in most cases changes to source code are reflected immediately in the running app without requiring a restart or any loss of [state](https://en.wikipedia.org/wiki/State_(computer_science)).

Release versions of Flutter apps are compiled with [ahead-of-time (AOT) compilation](https://en.wikipedia.org/wiki/Ahead-of-time_compilation" \o "Ahead-of-time compilation) on both Android and iOS, making Flutter's high performance on mobile devices possible.

**4.3.7 GOOGLE MAP**

Google map is a product of Google and it shows Map of the world with many great features. The features we have used for the development of this app include: Navigation, Direction, Markers, Current Location, Distance, Duration and Path. HTTP.

**4.3.8 JSON WEB SERVICE**

JSON-WSP (JavaScript Object Notation Web-Service Protocol) is a web-service protocol that uses JSON for service description, requests and responses. ... The description format has the same purpose for JSON-WSP as WSDL has for SOAP or IDL for CORBA, which is to describe the types and methods used in a given service.

**4.3.9 SAFETY REQUIREMENTS**

The application will not affect data stored outside of its servers nor will it affect any other applications installed in the system. It cannot cause any damage to the system or internal components. The application can be used in any browser or laptop, which meets minimum system specifications as mentioned above. As per the client request the application may use 1) Google Cloud Platform provides infrastructure as a service, platform as a service, and server less computing environments. It provides a series of modular cloud services including computing, data storage, data analytics and machine learning.18 2) Amazon Web Services is a subsidiary of Amazon that provides on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered pay-as-you-go basis

**4.3.10 SECURITY REQUIREMENTS**

Security relies on Google Cloud Platform Service Account/Data server maintained by client, for authentication, instead of the previously used client and developer access tokens. The Google security model is an end-to-end process, built on over 15 years of experience focused on keeping customers safe on Google applications like Gmail, Search and other Apps. With Google Cloud Platform your applications and data take advantage of the same security model.

**CHAPTER 5**

**5. DESIGN**

**5.1 SYSTEM DESIGN**

System can be defined, as an orderly grouping of interdependent components can be simple or complex. The most creative and challenging phase of the system life cycle is system design. The term design describes a final system and the process by which it is developed .It refers to the technical specifications that will be applied in implementing the candidate system .It also includes the construction of programs and program testing. The first step in the system design is to determine how the output is to be produced and in what format. Samples of the output and the inputs are also presented in the second step, input data and master files are to be designed to meet requirement of the proposed output .The processing phase’s system’s objectives and complete documentation.

System design has two phases:

➢ Logical

➢ Physical

The logical design reviews the present physical system, prepares the input and output and also prepares a logical design walk- through. We have to deal with how to take entries required and whether and how to process the user data.

Physical design maps out the details of the physical system, plans the system implementation, devices a test and implementation plan and new hardware and software.

We have to decide how and where to store the input data and how to process it so as to present it to the user in an easy, informative and attractive manner.

**Modules**

* Profile updates
* Work updates

(i) Quick Updates

(ii) Daily Reports

(iii) To do lists

* Construction details
* Album and gallery
* Google Map API
* Qr/bar code readers
* Expense Tracker
* ip cameras

ABOUT THE PROJECT (The main modules of the project are)

Data Acquisition in Construction Sites with Remote Monitoring is one the leading mobile application for an entire construction details regarding each construction site is  included in this module .Below are the sub module details

* **Profile updates**

In this module,it deals with the site engineer profle,in which all the details of site engineer has been given,it can be updated or can be viewedthe user can can update or edit any of the information in the profile

* **Work updates**

This module deals with the work updates of site engineers .Below are the sub modules

1. **Quick Updates:**

Quick updates deals with quick updates in the site.Site engineer must visit his/her construction sites multiple times a day ,during that time he/she must update the work details with images.Also if they found any issues ,those issues must be update with images.Application takes the location coordinates of site engineer while updating these details to server.which help to track site engineer work .

1. **Daily Reports:**

The daily work report deals with daily updates in the construction site.Site engineer must update the entire construction report at the end of the working day through this module.he/she must upload work details in a day which includes report,work status in the site,no of employees worked,upload documents & images.Also have to update complaints and messages  with images.

1. **To do lists:**

Manager will assign daily jobs to site engineer .These daily jobs are listed in this module.Site engineer view the details of this job and have to give a report after completion of each and ever task in todolist

* + - **Construction details :**

This module is related with construction projects and its details.Site engineer  login to the application and will view the entire construction details .He/she will view the details such as construction projects names,Locations,features of the construction ,view all documents related to the construction,details of the client etc.Also Location details on the construction site is implemented using google map.

* + - **Album and gallery:**

This module also deals with album and gallery.site engineer can create albums and can upload images of construction .These album can be viewed by client and the verification from manager web application.Also these albums are published in portfolio after the completion of the construction work

* **Google Map API:**

In this module Google Map API is integrated with the application which helps site engineer to view site location in google map with route maps,which helps site engineer to travel from to unknown construction location easily .

* + - **Qr/bar code readers:**

This module helps site engineer to scan QR codes /Bar codes which are used on different machines,employee idcards,Tool Kits etc .This application module uses android camera feature to scan qr/bar codes

* **Expense Tracker:**

This module also deals with expense details .Using expense details feature site engineer can update petty expenses that occurs in the site such as travel expense, purchase of small construction materials such a nail,clamps etc.

* **Ip cameras:**

This module deals with the integration of ip camera API in site engineer application,Which helps site engineer for remote monitoring .HTTP request session of the IP camera SDK is implemented in this module which helps to select construction sites and view IP camera installed in that remote locations

**5.2 Data Flow Diagram**

Data Flow Diagram (DFD) are directed graphs in which the nodes specify processing activities and the arcs that specify data items transmitted between processing nodes. Like flow charts DFD can be used at any desired level of abstraction. A DFD might represent data flow between individual statements or block of statements in a routine, data flow between concurrent process and data flow in a distributed computing system. Unlike flow charts DFD do not indicate a decision logic or condition under where various processing nodes in the diagram might be activated.

DFD is necessary for communicating for customer during requirement analysis; they are also widely used for representing external and internal design specifications. In the lack of structure DFD’s are quite valuable for establishing meaning, conventions and names of system components such as subsystems, files and data links. A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently to make the data access easy, inexpensive and flexible to the user

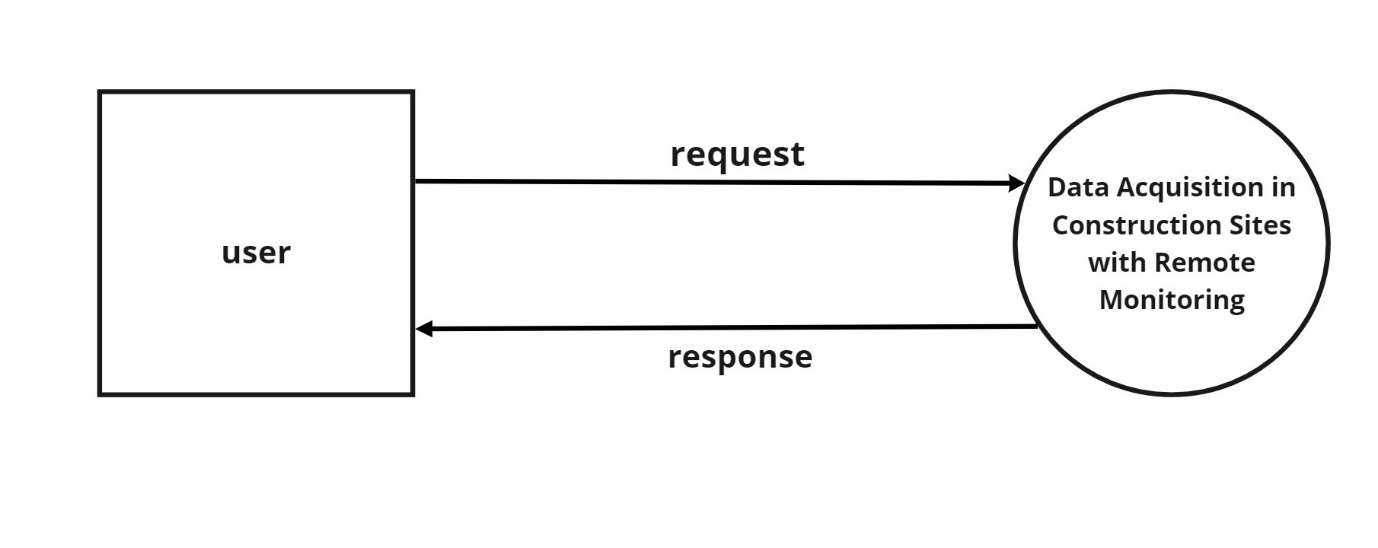
A DFD consists of a series of bubbles joined by lines. The bubble represents data transformation and line represents data flow in the system. In the normal convention a DFD has four major symbols:

* Square, this defines source or destination of data.
* Arrow, which shows data flow
* Circle, which represents a process that transforms incoming data into outgoing flow
* Open rectangle, which shows a data store.



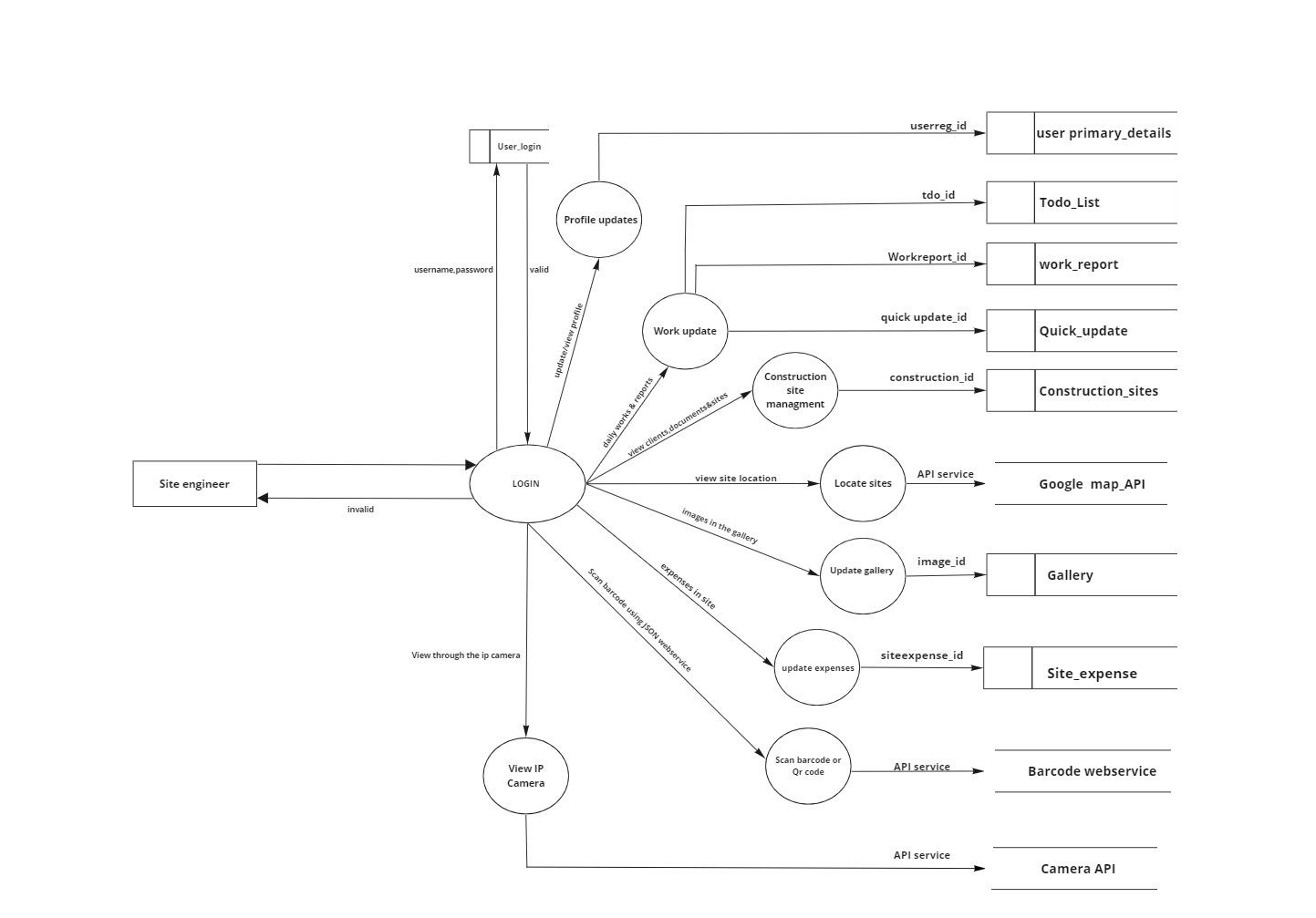
**Figure 5.1 DFD Components**

**5.2.2 PROJECT DFD**

****

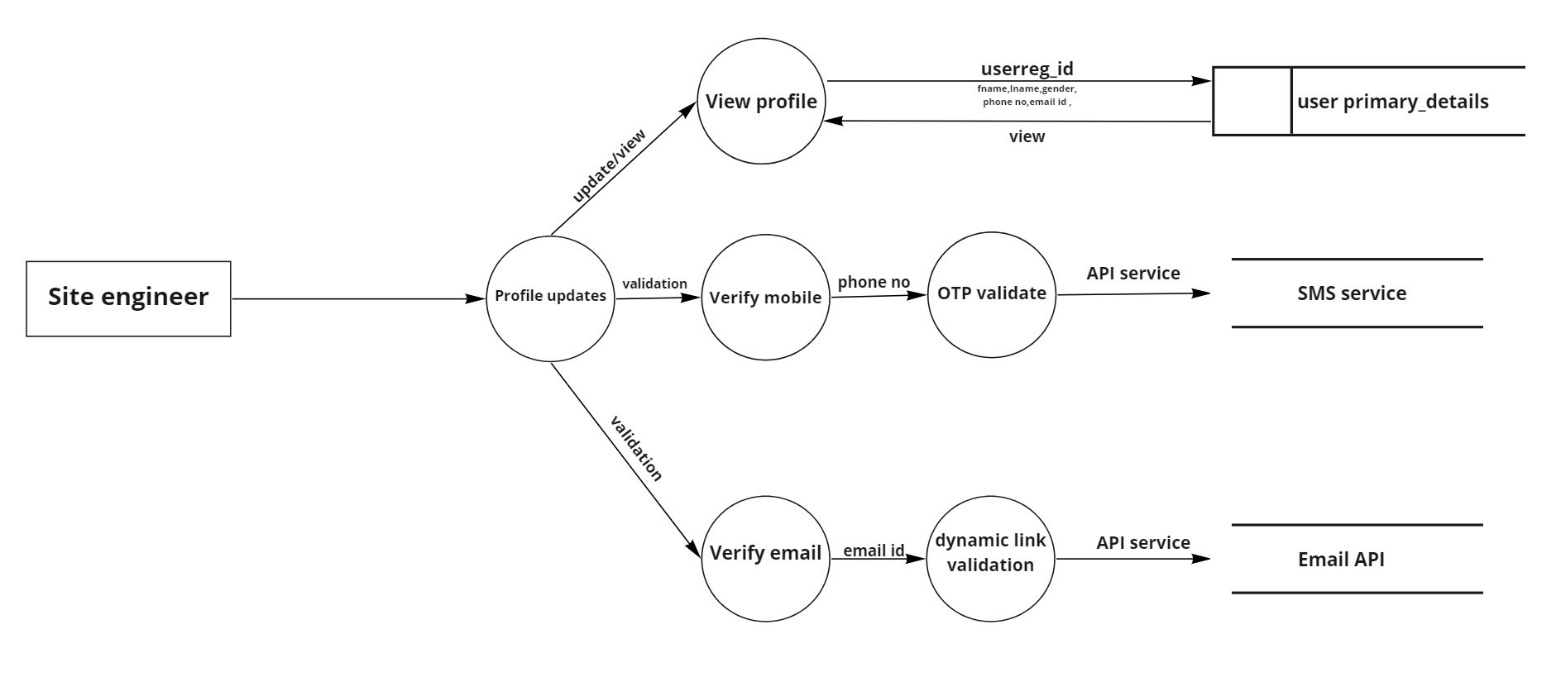
**Figure 5.2 Context level**

**Level 1**

****

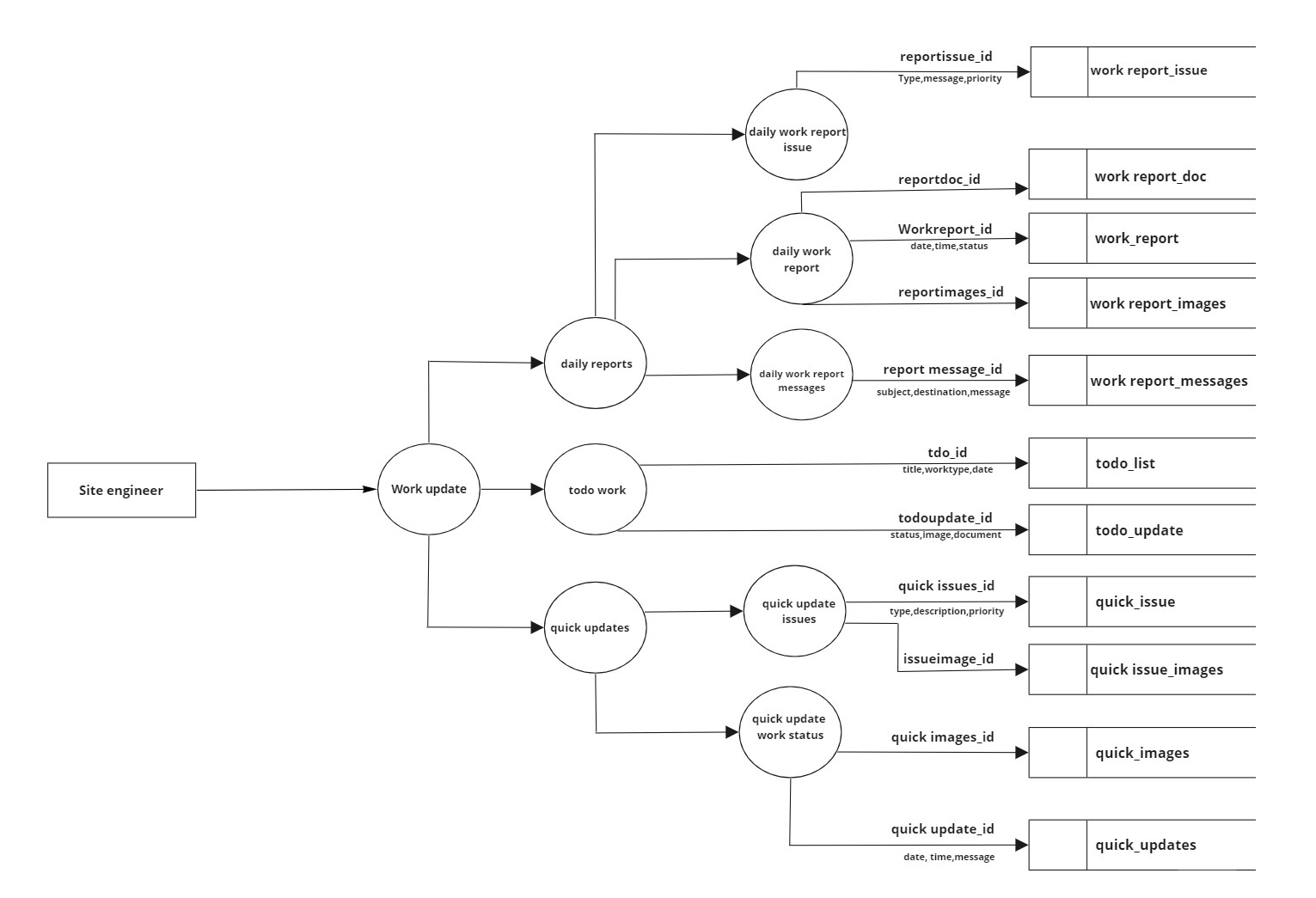
**Figure 5.3 Module Description (Level 1)**

**Level 2 (profile updates)**

****

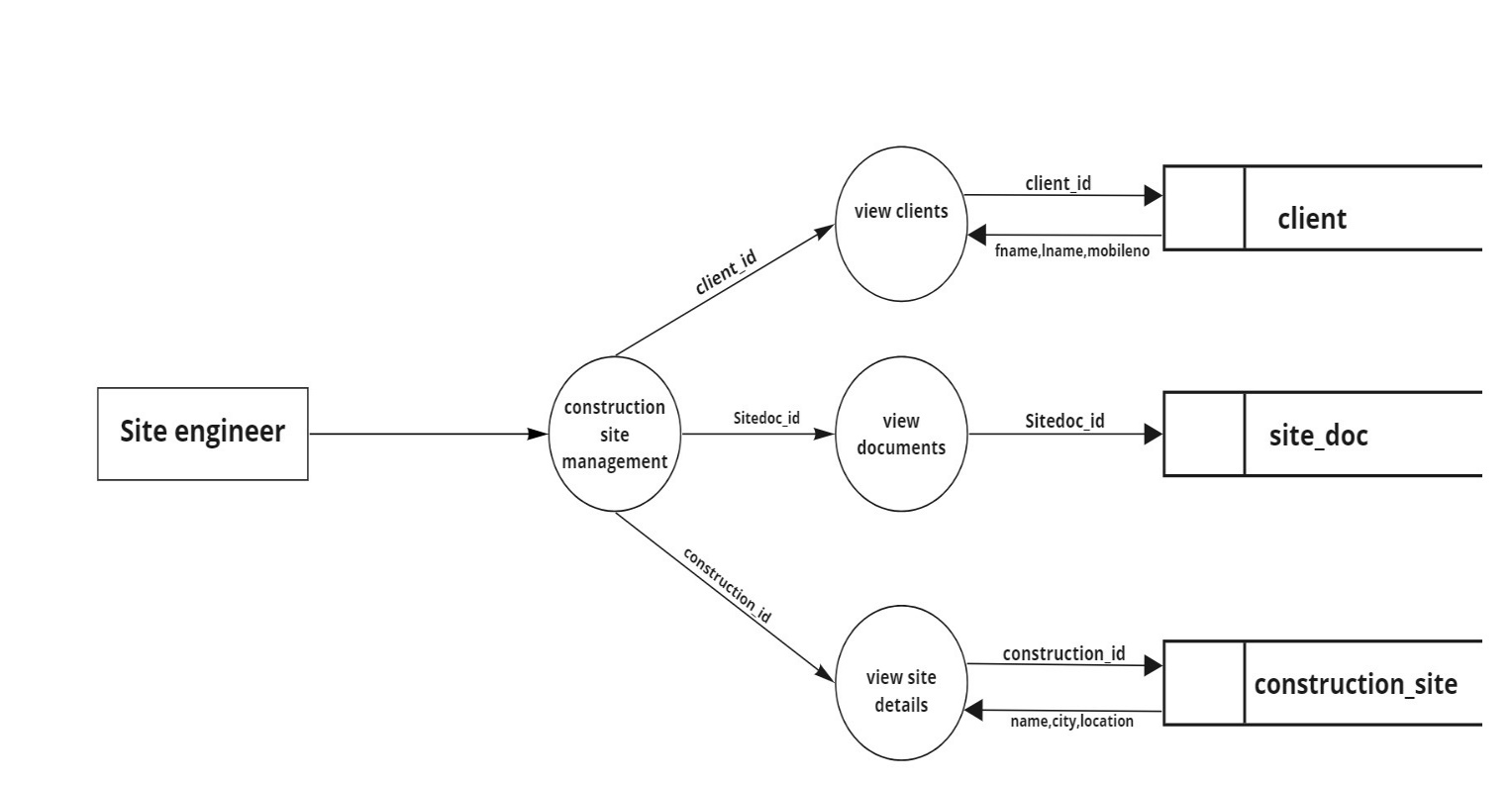
**Fig: 5.4 Profile DFD Level 2**

**Level 2 ( work updates)**

****

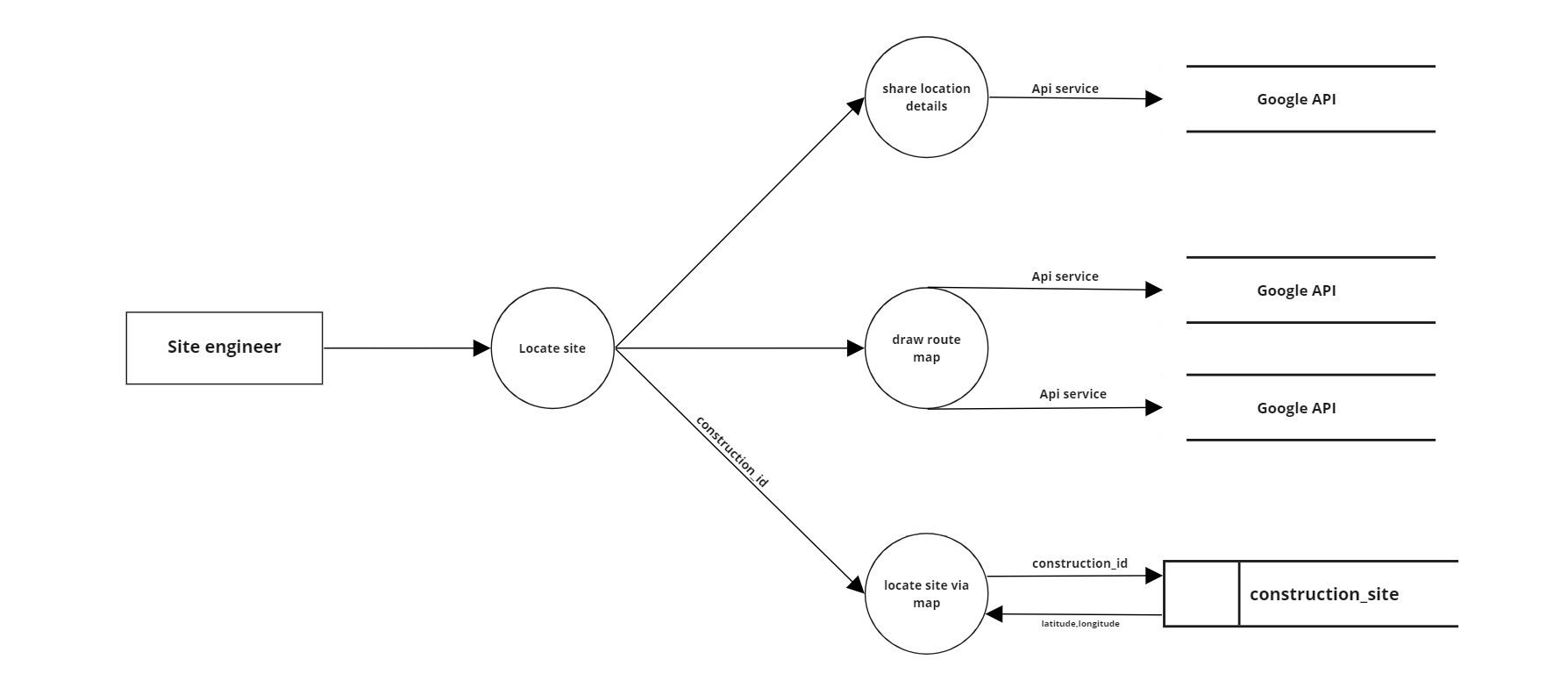
**Fig: 5.5 Work update DFD Level 2**

**Level 2 (Construction site management)**

****

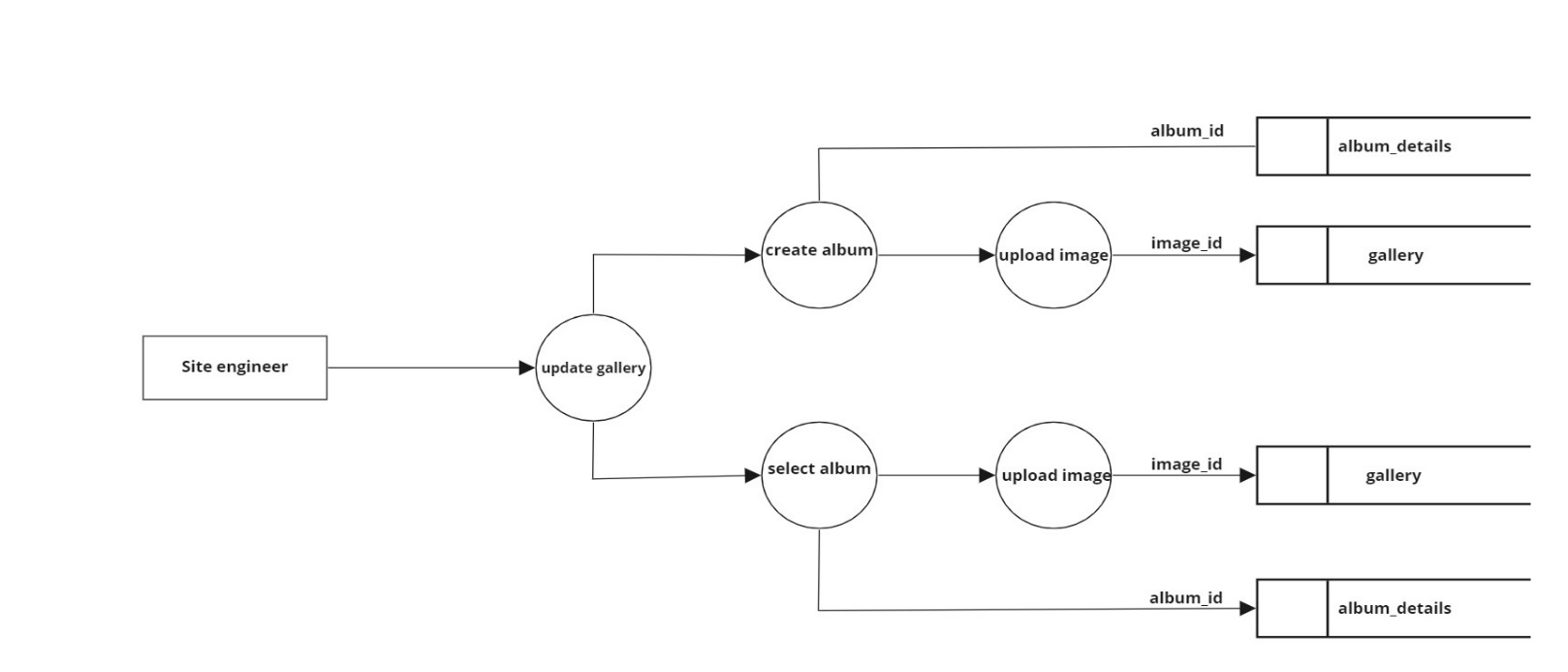
**Fig: 5.6 Construction site management DFD Level 4**

**Level 2 (Locate Site details )**

****

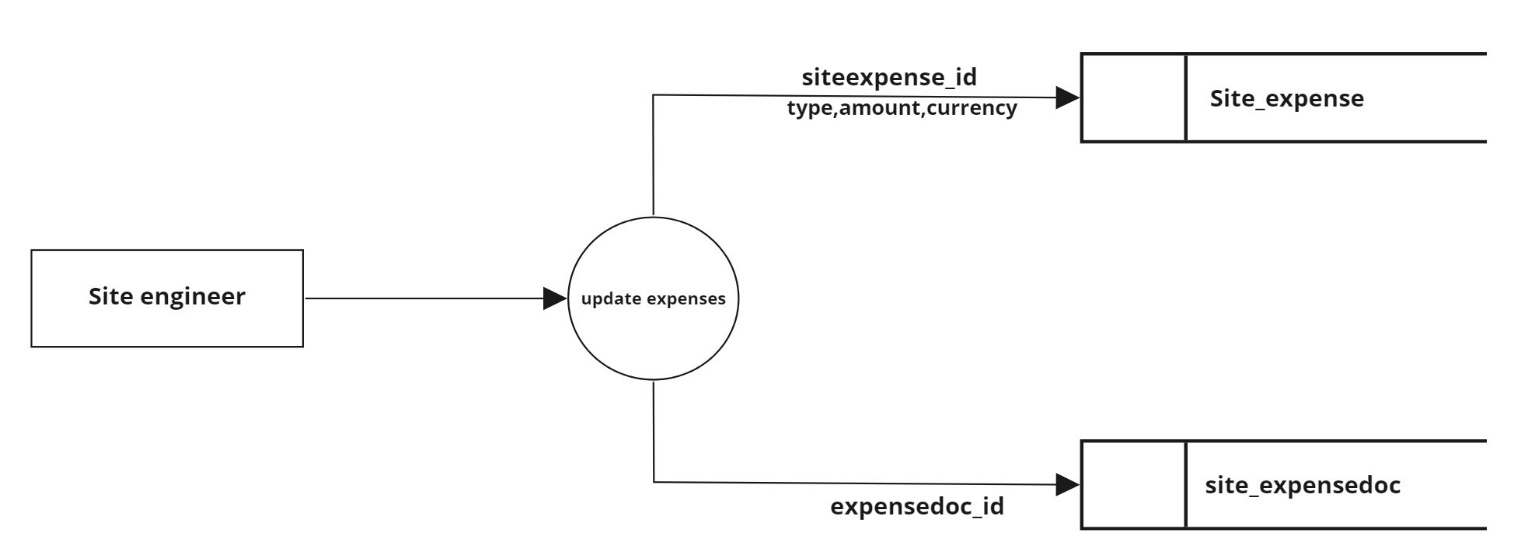
**Fig: 5.7 Locate Site details DFD Level 2**

**Level 2 (update gallery)**

****

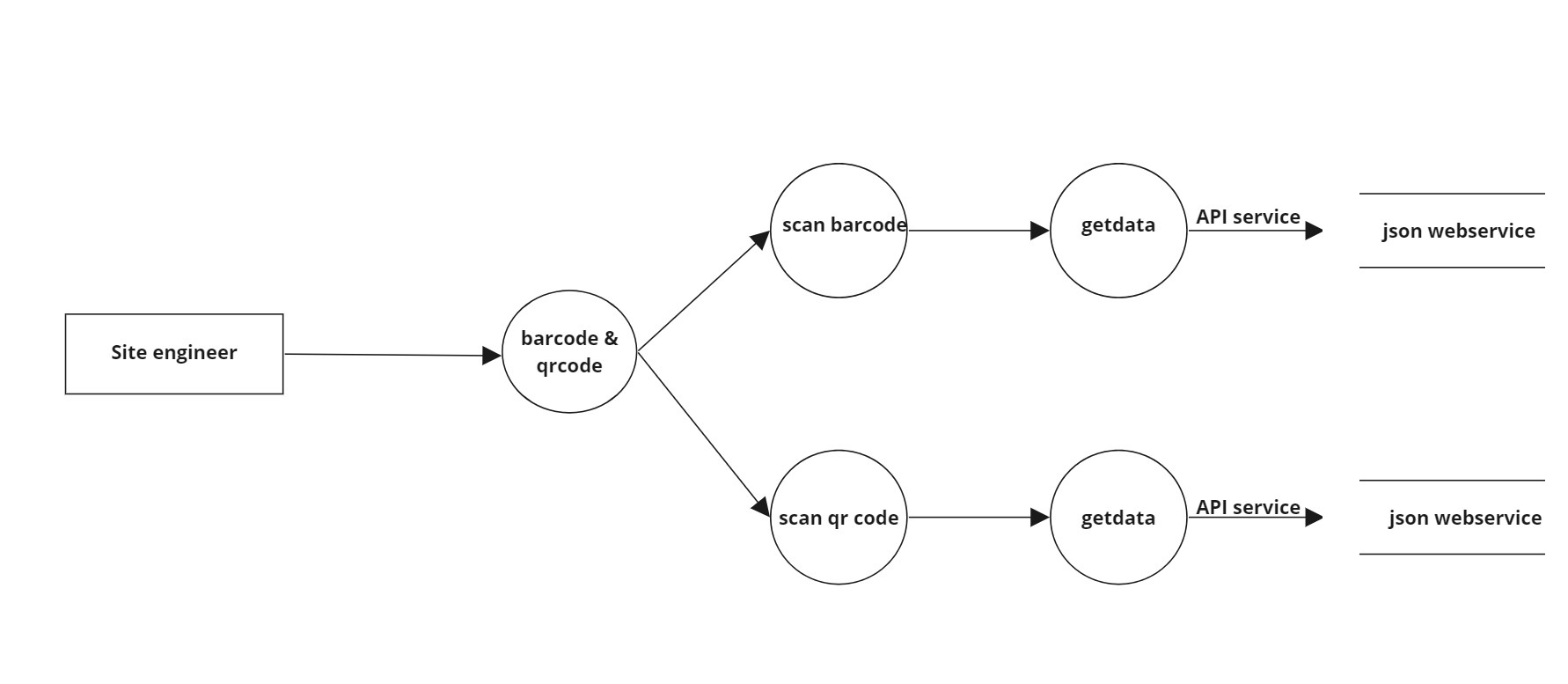
**Fig: 5.8 Gallery DFD Level 2**

**Level 2 ( Update expenses)**

****

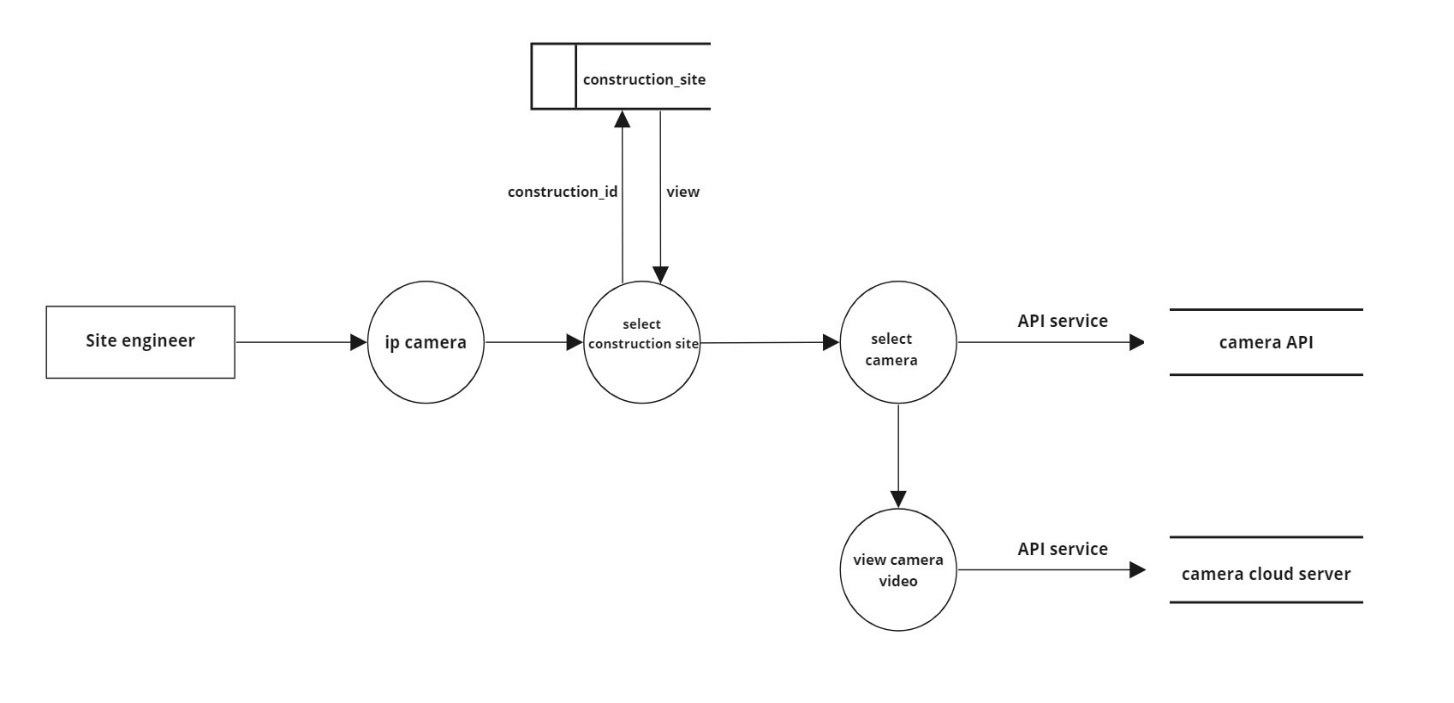
**Fig: 5.9 Update expenses DFD Level 2**

**Level 2 (barcode & Qr code )**

****

**Fig: 5.10 barcode & Qr code DFD Level 2**

**Level 2 (Ip camera )**

****

**Fig: 5.11 Ip camera DFD Level 2**

**5.3 DATABASE DESIGN**

A database is a collection of inter-related data stored with minimum redundancy, provides better data integrity and security and also to use many users quickly and efficiently. The general objective of database design is to make the data access easy, inexpensive and flexible to the user. Database design is recognized as a standard of management information system which is virtually available for every computer system.

Database design is the process of producing a detailed data model of a database. This data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity.

The term database design can be used to describe many different parts of the design of an overall database system. Principally, and most correctly, it can be thought of as the logical design of the base data structures used to store the data. In the relational model these are the tables and views. In an object database the entities and relationships map directly to object classes and named relationships. However, the term database design could also be used to apply to the overall process of designing, not just the base data structures, but also the forms and queries used as part of the overall database application within the database management system

The general theme behind a database is to integrate all the information. A database is the integrated collection of data and provides centralized access to data. Usually the centralized data managing software use relational database concepts and hence called RDBMS.

**Design Considerations**

The system is analyzed to the requirements and possible tables and fields are determined.

**Identifying** **Keys**

Once we have drawn upon the list of possible tables and fields, the next step in the logic database is to identify primary key and foreign key of the table.

**Primary Key**

The primary key (PK) of a relational table uniquely identifies each record in the table, it can either normal attribute that is guaranteed to be unique or it can be generated by the DBMS. Primary keys may consist of a single attribute or multiple attributes in combination.

**Foreign Keys**

A foreign key (FK) is a key comprised of a field or multiple field that to the primary key of another table. The concept of maintaining foreign keys is known as “referential integrity”.

**Defining Relationship**

A relationship is the term used to describe a connection between related tables. Stated another way, it means having shared fields in different tables that allow records to reference records in other tables. There are three possible types of relationships.

1. **One-to-One Relationships**

A one-to-one relationship indicates that each record in the table may relate to only one in another table.

1. **One-to-Many Relationships**

In a one-to-many relationship, any record in a table can relate to multiple records in a second table. 34

1. **Many-To-Many Relationships**

With many-to-many relationships many records in one table can link too many records in the second table.

1. **Normalization**

Normalization is a process of simplifying the database design to achieve the optimum structure. The steps in this process are known as normal form. These normal forms are a sequence of rules that are applied to progressively a database design. The higher the normal form of a database, the more efficient its underlying design. This is because, for a database to be simplified into third normal form, it must meet the criteria of first and second normal forms.

1. **First Normal Form**

To achieve first normal form, we must eliminate any repeating group. In the first normal form, we simplify our database structure to any repeating group. In other words first normal form include concept that field must be atomic or field represent one type of value for all the records.

1. **Second Normal Form**

To achieve second normal form, we must make sure that the non key fields depends on all the field in primary key every field in the table should be depend upon the entire primary key so that when new records are added, same value will not be repeated from records to records unnecessarily.

Full functional dependency indicates that if A and B are attributes of a relation, B is fully functionally dependent on A if B is functionally dependent on A, but not on any proper subset of A, Second normal form (2NF) is a relation that is in first normal form 35 and every non-primary-key attribute is fully functionally dependent on the primary key. The normalization of 1NF relations to 2NF involves the removal of partial dependencies. If a partial dependency exists, we remove the function dependent attributes from the relation by placing them in a new relation along with a copy of their determinant.

**(vi)Third Normal Form**

To achieve third normal form, we must make sure that the no fields depend on other non-key fields

**(vii) Fourth Normal Form**

An entity cannot have a one to one relationship between primary key columns and non key columns.

**(viii) Fifth normal Form**

Break all tables into the smallest possible pieces to eliminate all redundancy within a table. In conclusion, effective database design can help the development team reduce overall development time and costs. Undertaking the process of database design and creating a data model helps the team better understand the user’s requirements and thus enables them to build a system that is more reflective of the user’s requirements and business rules.

**DATABASE NAME: Data Acquisition in Construction Sites with Remote Monitoring**

Table Number : 5.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table NAME: quick\_update Primary key:**  **quick update\_id** | | | | |
| **Description: Details of updation** | | | | |
| Field | DataType | Size | Constraint | Descrption |
| quick update\_id | integer | 10 | Primary Key | Id of the quick update |
| quick update\_date | date |  | Not null | To update date |
| quick update\_time | integer | 10 | Not null | To update time |
| quick update\_heading | varchar | 20 | Not null | To update heading |

Table Number : 5.2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table NAME:User\_login Primary key: loginuser\_id** | | | | |
| **Description: Details of user login** | | | | |
| Field | DataType | Size | Constraint | Descrption |
| loginuser\_id | integer | 10 | Primary Key | Login id of user |
| loginuser\_name | varchar | 20 | Not null | Name of the user |
| loginuser\_password | password | 20 | Not null | Password of user |
| loginuser\_status | varchar | 20 | Not null | Status of user |
| loginuser\_section id | integer | 10 | Not null | Section id of user |
| loginuser\_latitude | integer | 10 | Not null | Latitude of the user |
| loginuser\_longitude | integer | 10 | Not null | Longitude of the user |
| loginuser\_recovery mail | varchar | 20 | Not null | Recovery email of the user |
| loginuser\_login attempt | varchar | 20 | Not null | No of attempts |
| loginuser\_active session | varchar | 20 | Not null | Active session of user |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| quick update\_desc | varchar | 20 | Not null | To update descrption |
| quick update\_latitude | integer | 10 | Not null | To update latitude |
| quick update\_longitude | integer | 10 | Not null | To update longitude |
| quick update\_location note | varchar | 20 | Not null | To update location note |
| Construction\_id | integer | 10 | Foreign Key | Id of the constuction |
| quick update\_message | varchar | 20 | Not null | To update message |

Table Number : 5.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table NAME: quick \_images Primary key:**  **quick images­\_id** | | | | |
| **Description:Details of images** | | | | |
| Field | DataType | Size | Constraint | Descrption |
| quick images­\_id | integer | 10 | Primary Key | Id of the image |
| quick images­\_path | varchar | 20 | Not null | Path of the image |
| quick images­\_type | varchar | 10 | Not null | Type of the image |
| quick images­\_size | integer | 10 | Not null | Size of the image |
| quick images­\_status | varchar | 20 | Not null | Status of the image |
| quick update\_id | integer | 10 | Foreign Key | Id of the quick update |

Table Number : 5.4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table NAME:quick\_issues Primary key:quick issues\_id** | | | | |
| **Description:Details of the quick issues** | | | | |
| Field | DataType | Size | Constraint | Descrption |
| quick issues\_id | integer | 10 | Primary Key | Id of the quick issues |
| quick issues\_type | varchar | 20 | Not null | Type of the quick issues |
| quick issues\_desc | varchar | 20 | Not null | Descrption of the quick issues |
| quick issues\_priority | varchar | 20 | Not null | Priority of the quick issues |
| quick issues\_status | varchar | 20 | Not null | Status of the quick issues |
| quick update\_id | integer | 10 | Foreign Key | Id of the quick update |

Table Number : 5.5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table NAME: quick issue\_images Primary key:**  **issueimage\_id** | | | | |
| **Description:** | | | | |
| Field | DataType | Size | Constraint | Descrption |
| issueimage\_id | integer | 10 | Primary Key | Id of the issue image |
| issueimage\_path | Varchar | 20 | Not null | Path of the issue image |
| issueimage\_type | Varchar | 20 | Not null | Type of the issue image |
| issueimage\_status | Varchar | 20 | Not null | Status of the issue image |
| issueimage\_size | integer | 10 | Not null | Size of the issue image |
| quick issues\_id | integer | 10 | Foreign Key | Id of the quick issues |

Table Number : 5.6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table NAME:Todo\_list Primary key: tdo\_id** | | | | |
| **Description: Details daily work** | | | | |
| Field | DataType | Size | Constraint | Descrption |
| tdo\_id | integer | 10 | Primary Key | Id of Todo Work |
| tdo\_title | Varchar | 20 | Not null | Title of Todo Work |
| tdo\_date | date |  | Not null | Date of Todo Work |
| tdo\_worktype | Varchar | 20 | Not null | Worktype of Todo Work |
| tdo\_priority | Varchar | 20 | Not null | Priority of Todo Work |
| tdo\_document | Varchar | 20 | Not null | Document of of Todo Work |
| tdo\_image | Varchar | 20 | Not null | Image of Todo Work |
| tdo\_employeetype | Varchar | 20 | Not null | Employee type of Todo Work |
| Construction\_id | integer | 10 | Foreign Key | Id of construction site |

Table Number : 5.7

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table NAME:Todo\_update Primary key:**  **todoupdate\_id** | | | | |
| **Description:Details of daily updation** | | | | |
| Field | DataType | Size | Constraint | Descrption |
| todoupdate\_id | integer | 10 | Primary Key | Id of Todo update |
| todo\_id | integer | 10 | Foreign Key | Id of Todo Work |
| todoupdate\_reply message | Varchar | 20 | Not null | Reply message of Todo update |
| todoupdate\_status | Varchar | 20 | Not null | Status of Todo update |
| todoupdate\_image | Varchar | 20 | Not null | Image of Todo update |
| todoupdate\_document | Varchar | 20 | Not null | Document of Todo update |
| loginuser\_id | integer | 10 | Foreign Key | Id of user |

Table Number : 5.8

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table NAME:work\_report Primary key:**  **Workreport\_id** | | | | |
| **Description:Details of daily work report** | | | | |
| Field | DataType | Size | Constraint | Descrption |
| Workreport\_id | integer | 10 | Primary Key | Id of workreport |
| Workreport\_date | date |  | Not null | Date of of workreport |
| Workreport\_time | integer | 10 | Not null | Time of workreport |
| Workreport\_subject | Varchar | 20 | Not null | Subject of workreport |
| Workreport\_report | Varchar | 20 | Not null | Report of workreport |
| Workreport\_status | Varchar | 20 | Not null | Status of workreport |
| Workreport\_employee number | integer | 10 | Not null | Employee number of workreport |
| Workreport\_type | Varchar | 20 | Not null | Type of workreport |
| loginuser\_id | integer | 10 | Foreign Key | Id of user |
| Construction\_id | integer | 10 | Foreign Key | Id of construction site |

Table Number : 5.9

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table NAME:** **Workreport\_images Primary key: reportimages\_id** | | | | |
| **Description:Details of work report images** | | | | |
| Field | DataType | Size | Constraint | Descrption |
| reportimages\_id | integer | 10 | Primary Key | Id of report images |
| reportimages\_path | Varchar | 20 | Not null | Path of report images |
| reportimages\_type | Varchar | 20 | Not null | Type of report images |
| reportimages\_size | integer | 10 | Not null | Size of report images |
| reportimages\_status | Varchar | 20 | Not null | Status of report images |
| Workreport\_id | integer | 10 | Foreign Key | Id of workreport |

Table Number : 5.10

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table NAME:work report\_doc Primary key: reportdoc\_id** | | | | |
| **Description:details of work report document** | | | | |
| Field | DataType | Size | Constraint | Descrption |
| reportdoc\_id | integer | 10 | Primary Key | Id of Reportdoc |
| reportdoc\_path | Varchar | 20 | Not null | Path of Reportdoc |
| reportdoc\_type | Varchar | 20 | Not null | Type of Reportdoc |
| reportdoc\_size | integer | 10 | Not null | Size of Reportdoc |
| reportdoc\_status | Varchar | 20 | Not null | Status of Reportdoc |
| Workreport\_id | integer | 10 | Foreign Key | Id of workreport |

Table Number : 5.11

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table NAME:Workreport\_issue Primary key: reportissue\_id** | | | | |
| **Description:Details of work report issue** | | | | |
| Field | DataType | Size | Constraint | Descrption |
| reportissue\_id | integer | 10 | Primary Key | Id of work report issue |
| reportissue\_type | Varchar | 20 | Not null | Type of work report issue |
| reportissue\_message | Varchar | 20 | Not null | Message of work report issue |
| Workreport\_id | integer | 10 | Foreign Key | Id of workreport |
| reportissue\_status | Varchar | 20 | Not null | Status of work report issue |
| reportissue\_priority | Varchar | 20 | Not null | Priority of work report issue |
| reportissue\_image | Varchar | 20 | Not null | Image of work report issue |

Table Number : 5.12

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table NAME:workreport\_messaage Primary key: report message\_id** | | | | |
| **Description: Details of work report message** | | | | |
| Field | DataType | Size | Constraint | Descrption |
| report message\_id | integer | 10 | Primary Key | Id of report message |
| report message\_subject | Varchar | 20 | Not null | Subject of report message |
| report message\_designation | Varchar | 20 | Not null | Designation of report message |
| report message\_message | Varchar | 20 | Not null | Message of report message |
| report message\_type | Varchar | 20 | Not null | Type of report message |
| Workreport\_id | integer | 10 | Foreign Key | Id or workreport |

Table Number : 5.13

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table NAME: construction\_site Primary key:** **construction\_id** | | | | |
| **Description:Details of construction site** | | | | |
| Field | DataType | Size | Constraint | Descrption |
| construction\_id | integer | 10 | Primary Key | Id of construction site |
| construction\_name | Varchar | 20 | Not null | Name of construction site |
| Client\_id | integer | 10 | Foreign Key | Id of client |
| construction\_city | Varchar | 20 | Not null | City of construction site |
| construction\_location | Varchar | 20 | Not null | Location of construction site |
| construction\_address | Varchar | 20 | Not null | Address of construction site |
| construction\_construction type | Varchar | 20 | Not null | Type of construction site |
| construction\_work type | Varchar | 20 | Not null | Work type of construction site |
| construction\_work desc | Varchar | 20 | Not null | Work Descrption of construction site |
| construction\_date of start | date |  | Not null | Date of start of construction site |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| construction\_latitude | integer | 10 | Not null | Latitude of construction site |
| construction\_longitude | integer | 10 | Not null | longitude of construction site |
| construction\_total area | integer | 10 | Not null | total area of construction site |
| construction\_land type | Varchar | 20 | Not null | land type of construction site |
| construction\_total floors | integer | 10 | Not null | Total floors of construction site |
| construction\_description | Varchar | 20 | Not null | Descrption of construction site |

Table Number : 5.14

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table NAME: site\_doc Primary key: Sitedoc\_id** | | | | |
| **Description:Details of site document** | | | | |
| Field | DataType | Size | Constraint | Descrption |
| Site doc\_id | integer | 10 | Primary Key | Id of site document |
| Site doc\_type | Varchar | 20 | Not null | Type of of site document |
| Site doc\_name | Varchar | 20 | Not null | Name of site document |
| Site doc\_date | date |  | Not null | Date of site document |
| Site doc\_desc | Varchar | 20 | Not null | Descrption of site document |
| Site doc\_size | integer | 10 | Not null | Size of site document |
| Site doc\_status | Varchar | 20 | Not null | Status of site document |
| construction\_id | integer | 10 | Foreign Key | Id of construction |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table NAME:client Primary key: client\_id** | | | | |
| **Description:Details of client** | | | | |
| Field | DataType | Size | Constraint | Descrption |
| client\_id | integer | 10 | Primary Key | ID of client |
| client\_fname | Varchar | 20 | Not null | First name of client |
| client\_Lname | Varchar | 20 | Not null | Second name of client |
| client\_primary mobile | integer | 10 | Not null | Mobile number of client |
| client\_primaryWhatsapp | integer | 10 | Not null | Whatsapp number of client |
| client\_primary email | Varchar | 20 | Not null | Email id of client |
| client\_ address | Varchar | 20 | Not null | Address of client |
| client\_location | Varchar | 20 | Not null | Location of client |
| client\_city | Varchar | 20 | Not null | City of client |
| client\_state | Varchar | 20 | Not null | State of client |
| client\_photo | Varchar | 20 | Not null | Photo of client |
| client\_identitycard | Varchar | 20 | Not null | Identity card of client |
| client\_secondary mobile | integer | 10 | Not null | Secondary mobile number of client |
| client\_secondary email | Varchar | 20 | Not null | Secondary email of client |
| client\_secondary whatsapp | Varchar | 20 | Not null | Secondary whatsapp number of client |

Table Number : 5.15

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table NAME:album\_details Primary key: album\_id** | | | | |
| **Description:Details of Album** | | | | |
| Field | DataType | Size | Constraint | Descrption |
| album\_id | integer | 10 | Primary Key | Id of album |
| album\_name | Varchar | 20 | Not null | Name of album |
| album\_desc | Varchar | 20 | Not null | Descrption of album |
| album\_date of create | date |  | Not null | date of creation of album |
| album\_status | Varchar | 20 | Not null | status of album |
| album\_total image | integer | 10 | Not null | Total image of album |
| album\_viewer | Varchar | 20 | Not null | viewer of album |
| construction\_id | integer | 10 | Foreign Key | Id of construction |

Table Number : 5.16

Table Number : 5.17

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table NAME: gallery Primary key: image\_id** | | | | |
| **Description:Details of Gallery images** | | | | |
| Field | DataType | Size | Constraint | Descrption |
| image\_id | integer | 10 | Primary Key | Id of the image |
| image\_name | Varchar | 20 | Not null | Name of the image |
| image\_desc | Varchar | 20 | Not null | Descrption of the image |
| image\_type | Varchar | 20 | Not null | Type of the image |
| image\_size | integer | 10 | Not null | Size of the image |
| image\_status | Varchar | 10 | Not null | Status of the image |
| image\_date of creation | date |  | Not null | Date of creation of the image |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| album\_id | integer | 10 | Foreign Key | Id of the album |
| loginuser\_id | integer | 10 | Foreign Key | Id of the user |

Table Number : 5.18

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table NAME: Site\_expense Primary key: siteexpense\_id** | | | | |
| **Description:Details of site expense** | | | | |
| Field | DataType | Size | Constraint | Descrption |
| siteexpense\_id | integer | 10 | Primary Key | Id of site expense |
| siteexpense\_head | Varchar | 20 | Not null | Head of site expense |
| siteexpense\_type | Varchar | 20 | Not null | Type of site expense |
| siteexpense\_desc | Varchar | 20 | Not null | Descrption of site expense |
| siteexpense\_date of expense | date |  | Not null | Date of expense of site expense |
| siteexpense\_date of update | date |  | Not null | date of update of site expense |
| siteexpense\_amount | integer | 10 | Not null | Amount of site expense |
| siteexpense\_currency | integer | 10 | Not null | Curreny of site expense |

Table Number : 5.19

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table NAME: Site\_expensedoc Primary key:expensedoc\_id** | | | | |
| **Description:Details of site expense document** | | | | |
| expensedoc\_id | integer | 10 | Primary Key | Id of expensedoc |
| expensedoc\_type | Varchar | 20 | Not null | Type of expensedoc |
| expensedoc\_size | integer | 10 | Not null | Size of expensedoc |
| expensedoc\_extension | Varchar | 20 | Not null | Extension of expensedoc |
| expensedoc\_path | Varchar | 20 | Not null | Path of expensedoc |
| siteexpense\_id | integer | 10 | Foreign Key | Id of site expense |

Table Number : 5.20

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table NAME:user\_doc Primary key: userdoc\_id** | | | | |
| **Description: Details of user documents** | | | | |
| Field | DataType | Size | Constraint | Descrption |
| userdoc\_id | integer | 10 | Primary Key | Id of userdoc |
| userdoc\_type | Varchar | 20 | Not null | type of document |
| userdoc\_name | Varchar | 20 | Not null | Name of document |
| userdoc\_docpath | Varchar | 20 | Not null | path of document |
| userdoc\_docextension | Varchar | 20 | Not null | extension of document |
| userdoc\_status | varchar | 20 | Not null | status of document |

Table Number : 5.21

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table NAME:user primary\_details Primary key: userreg\_id** | | | | |
| **Description:Details of user Registeration** | | | | |
| Field | DataType | Size | Constraint | Descrption |
| userreg\_id | integer | 10 | Primary Key | Id of user |
| userreg\_fname | Varchar | 20 | Not null | Fname of user |
| userreg\_Lname | Varchar | 20 | Not null | Lname of user |
| userreg\_gender | Varchar | 20 | Not null | Gender of user |
| userreg\_date of birth | date |  | Not null | Date of birth of user |
| userreg\_date of join | date |  | Not null | Date of join of user |
| userreg\_primary email | Varchar | 20 | Not null | primary Email of user |
| userreg\_ primary mobile no | integer | 10 | Not null | primary Mobile no of user |
| userreg\_secondary mobile no | integer | 10 | Not null | secondary mobile no of user |
| userreg\_whatsapp number | integer | 10 | Not null | Whatsapp no of user |
| userreg\_primary mobile status | integer | 10 | Not null | Primary mobile status no of user |
| userreg\_secondary mobile status | Varchar | 20 | Not null | Secondary mobile status no of user |
| userreg\_pe email status | Varchar | 20 | Not null | personal email of user |
| userreg\_offical email | Varchar | 20 | Not null | Official email of user |
| userreg\_off email status | Varchar | 20 | Not null | Email status of user |
| userreg\_secuirty pin | integer | 10 | Not null | Security pin of user |
| userreg\_address | Varchar | 20 | Not null | Address of user |
| userreg\_city | Varchar | 20 | Not null | City of user |
| userreg\_location latitude | integer | 10 | Not null | Latitude of user |
| userreg\_location longitude | integer | 10 | Not null | Longitude of user |
| userreg\_state | Varchar | 20 | Not nullS | State of user |
| userreg\_photo | Varchar | 20 | Not null | Photo of user |
| userreg\_idcard | Varchar | 20 | Not null | Idcard of user |
| userreg\_blood group | Varchar | 20 | Not null | Blood group of user |

**5.4 INPUT DESIGN**

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data into a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. In the input designing we had considered the following things

* What data should be given as input?
* How the data should be arranged or coded?
* The dialog to guide the operating personnel in providing input.
* Methods for preparing input validations and steps to follow when errors occur. It is achieved by creating user friendly screens for the data entry to handle large
* volume of data.
* The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be
* performed. It also provides record viewing facilities Input design is the process of converting a user oriented description of the input into a computer based system This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

When the data is entered it will check for its validity Data can be entered for its Validity. Mostly input data can be selected from a list of dataitems Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input is to create an input lay out that is easy to follow. For example in user registration form all the fields except some optional fields must be enter, otherwise display appropriate error message.

**5.5 OUTPUT DESIGN**

Output design is one of the most important features of the information system. When the output is not of good quality, the users will be averse to use the newly designed system and may not use the system. There are many types of outputs, all of which can be either highly useful or can be critical to the users, depending on the manner and degree to which they are used. Outputs from computer system are required primarily to communicate the results of processing to users. They are also used to provide a permanent hard copy of the results for later consultation

**Table Number: 5.22**

|  |  |  |
| --- | --- | --- |
| **PROCESS** | **INPUT DESIGN** | **OUTPUT DESIGN** |
| Login page for site engineer | Enter username and password | Site engineer login successfully |
| Add quick updates | Add details in the corresponding field | Quick updates succesfull |
| Add expenses | Add details in the corresponding field | Expenses updated successfully |
| Add daily report | Update in the corresponding fields | Daily report updated successfully |
| Todo list details | Lists of Todo jobs | Viewing successfully |
| Construction site details | List of construction sites | Viewing successfully |
| Client details | Display client details | View client details |
| Update work report issues | Update the issues | Updated successfully |
| Add pictures in gallery | Upload pictures | Successfully uploaded |
| View site loaction | List of site location | View the site Location |
| Scan QR code | Display QR code | Scan successful and show details |
| Scan Bar code | Display Bar code | Scan successful and show details |
| View Ip cameras | List of Site with ip camera | Viewing successful |

**Table 5.22 Input Output Design**

* 1. **PROGRAM DESIGN**

**(i) Site engineer**

Step 1: START

Step 2: Site engineer has a username and password to login.

Step 3: Site engineer can view profile

Step 4: Site engineer can add quick updates

Step 5: Site engineer can view Todo work

Step 6: Site engineer can add daily report

Step 7: Site engineer can add site expenses

Step 8: Site engineer can view client details

Step 9: Site engineer can view construction sites

Step 10: Site engineer can upload images

Step 11: Site engineer can scan Qr code & barcode

Step 12: Site engineer can view sites through Ip camera

Step 13: Site engineer can locate sites

Step 14: Site engineer can report work report issues

Step 15: STOP

**CHAPTER 6**

**6. FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS**

**6.1 FUNCTIONAL REQUIREMENTS**

In software engineering, a functional requirement defines a function of a software system or

its component. A function is described as a set of inputs, the behavior, and outputs. Functional

requirements may be calculations, technical details, data manipulation and processing and

other specific functionality that define what a system is supposed to accomplish. Generally,

functional requirements are expressed in the form "system must do requirement ".

Functional requirements for each of the uses cases described below :

* Site engineer should login with his credentials for doing any activity, To view profile or list of activities and the amount of the work can be viewed.
* User of the application needed  android /ios mobile-apk file/ipa file ,Internet connection and Permissions :GPS,SMS, Internet, Storage permission, Physical Activity, camera.

**6.2 NON-FUNCTIONAL REQUIREMENTS**

A non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours. Non-functional requirements are “system shall be requirement ". Non-functional requirements are often called qualities of a system. Other terms for non-functional requirements are "constraints", "quality attributes”, “quality goals", "quality of service requirements" and "non-behavioural requirements.

Some of the non-functional requirements are mentioned below:

* **Usability:** The system shall have a clean interface with only needed features, clear terminology and tool tips wherever necessary. Warning or alerts shall be specified in clear way.
* **Efficiency**: The system shall respond to different searches being conducted like searching particular product, search quantity, etc. in a very fast way.
* **Interoperability:** The system shall be able to interact with other systems.
* **Portability:** the system shall be independent of the specific technological platform used to implement it.
* **Reliability:** It defined as a measure of the time between failures occurring in a system (measure show frequently the system fails), so that the system shall operate without any failures for a particular period of time.
* **Availability:** It measures the percentage of time the system is in its operational state so that the system shall be available for use 24 hours per day and 365 days per year

**CHAPTER 7**

**7.TESTING**

**7.1 TESTING STRATEGIES**

An engineered product can be tested in one of these two ways.

These testing strategies include:

• Black box testing

• White box testing

**White box testing**

White-box testing is a method of testing the application at the level of the source code. White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality. In white-box testing an internal perspective of the system, as well as programming skills, are chooses inputs to exercise paths through the code and determine the expected outputs.

**Black box testing**

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing: unit, integration, system and acceptance. It is sometimes referred to as specification-based testing.

**7.2 UNIT TESTING**

In computer programming, unit testing is a software method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures are tested to determine whether they are fit for use intuitively, one can view a unit as the smallest testable part of an application. In procedural programming a unit could be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process. If forms the basis for component testing.

**7.3 INTEGRATION TESTING**

This is the final step in testing. In this case all the modules were combined and given the test data. The combined module works successfully without any side effect on other programs. Everything was found to be working correctly. In this the entire system was tested as a whole with all modules. This form of testing is popularly known as Black Box testing or system testing. Black Box testing methods focus on the functional requirement of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program. Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in data structures or external database access, performance errors and initialization errors and termination errors.

**7.4 SYSTEM TESTING**

Testing is a set activity that can be planned and conducted systematically. Testing begins at the module level and work towards the integration of entire computers based system. Nothing is complete without testing, as it is vital success of the system.

Testing Objectives:

There are several rules that can serve as testing objectives, they are

• Testing is a process of executing a program with the intent of finding an error

• A good test case is one that has high probability of finding an undiscovered error.

• A successful test is one that uncovers an undiscovered errors.

A test case is a specification of the inputs, execution conditions, testing procedure, and expected results that define a single test to be executed to achieve a particular software testing objective, such as to exercise a particular program path or to verify compliance with a specific requirement. Test cases underlie testing that is methodical rather than haphazard. A battery of test cases can be built to produce the desired coverage of the software being tested. Formally defined test cases allow the same tests to be run repeatedly against successive versions of the software, allowing for used to design test cases. The tester effective and consistent regression testing

**7.5 TESTING RESULTS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sl no.** | **Test case** | **Input** | **Expected output** | **Actual Results** | **Result** |
| 1 | Login Page(Site Engineer) | Username,  Password | Site Engineer Dashboard | As Expected | Pass |
| 2 | Add Quick Updates | Construction Site, Image, Issues,  Descrption | Details Added | As Expected | Pass |
| 3 | Add Expenses | Type, Date, Descrption, Message | Details Added | As Expected | Pass |
| 4 | Add Daily Report | Report, Image, Issues, Message | Daily Report Added Successfully | As Expected | Pass |
| 5 | Todo Lists | Update as Pending or Completed | Updated Successfully | As Expected | Pass |
| 6 | Gallery | Upload Image | Uploaded Successfully | As Expected | Pass |
| 7 | Work Report | Update Issues, Image | Updated Successfully | As Expected | Pass |
| 8 | Clients | Name, Phone No, Email Id | Viewed Successfully | As Expected | Pass |
| 9 | Construction Sites | Name Of The Client, Location | Viewed Successfully | As Expected | Pass |
| 10 | Scan Qr & Barcode | Shows Scanner | Shows The Data | As Expected | Pass |
| 11 | Locate Sites | Show Route Map , Locate Sites | Location Showed Successfully | As Expected | Pass |
| 12 | Ip Camera | Construction Site, Camera Name | Viewed Successfully | As Expected | Pass |

**Table number 7.1**

**CHAPTER 8**

**8.1 RESULTS AND DISCUSSION**

**8.1 RESULTS (SALIENT FEATURES)**

The main motivation and objective of Data Acquisition in Construction Sites with Remote Monitoring is to enhance and upgrade the existing system by increasing its efficiency and effectiveness. The software improves the working methods by replacing the existing manual system with new feathures. Enabling the user experiencing a friendly user interface.

The proposed system incorporated with the following features:

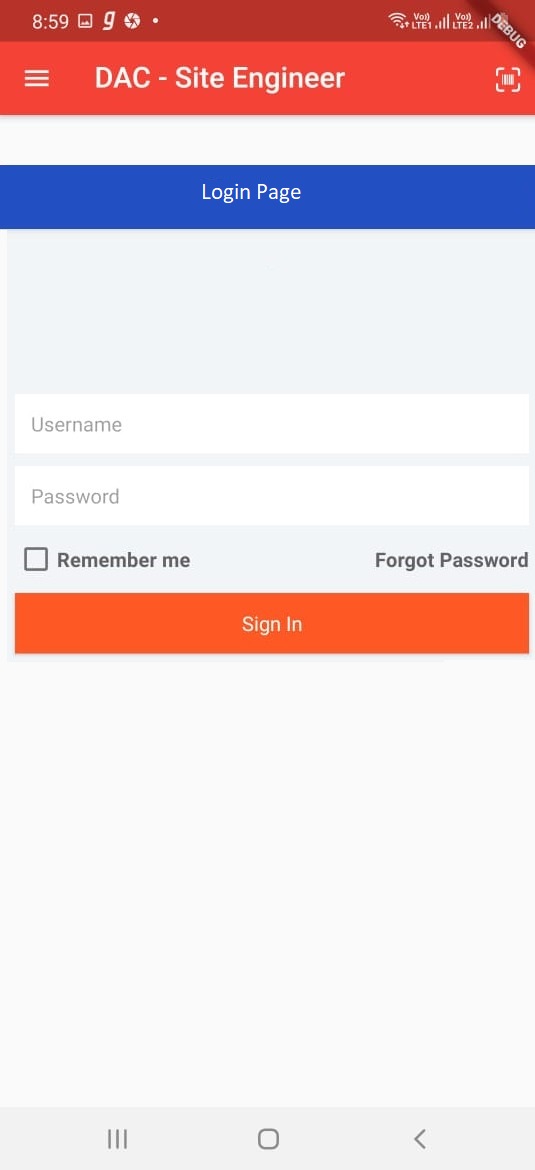
* Improved efficiency.
* Flexibility.
* Improved accuracy.
* Simplifies complex task.
* High scalability.
* User friendly and customized.
* Technologically updated.
* Less waiting time.
* Update feedback /complaints & rate.

Table above in chapter 7 summarizes the results of tests conducted. The table gives the main characteristics of each action, its expected output and actual output obtained .The table gives

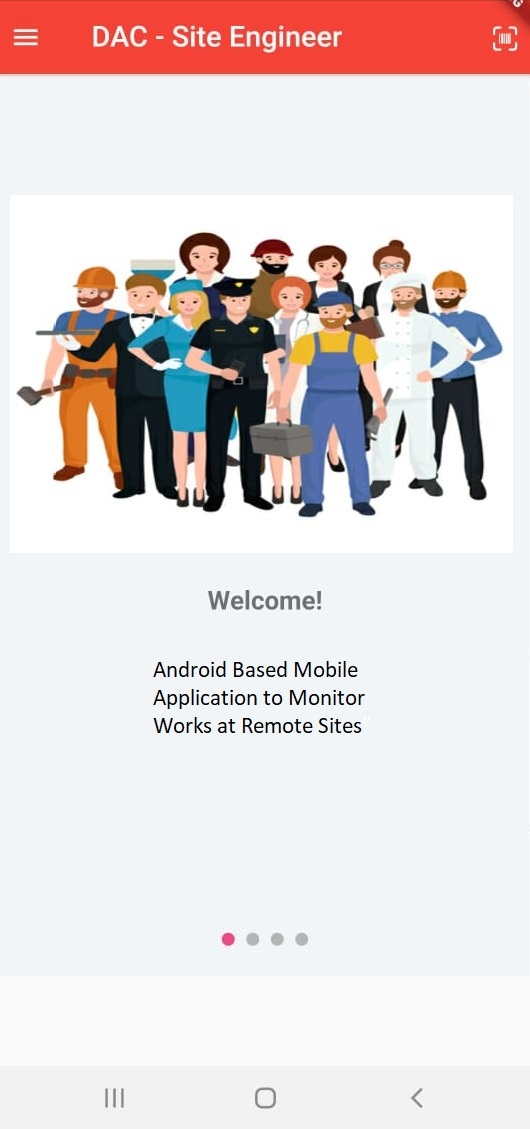
* The number of tests conducted
* Description of the test.
* Who conducted the test
* What was the expected output
* What is the actual output
* Was a pass or fail.

**8.2 SCREEN SHOTS**

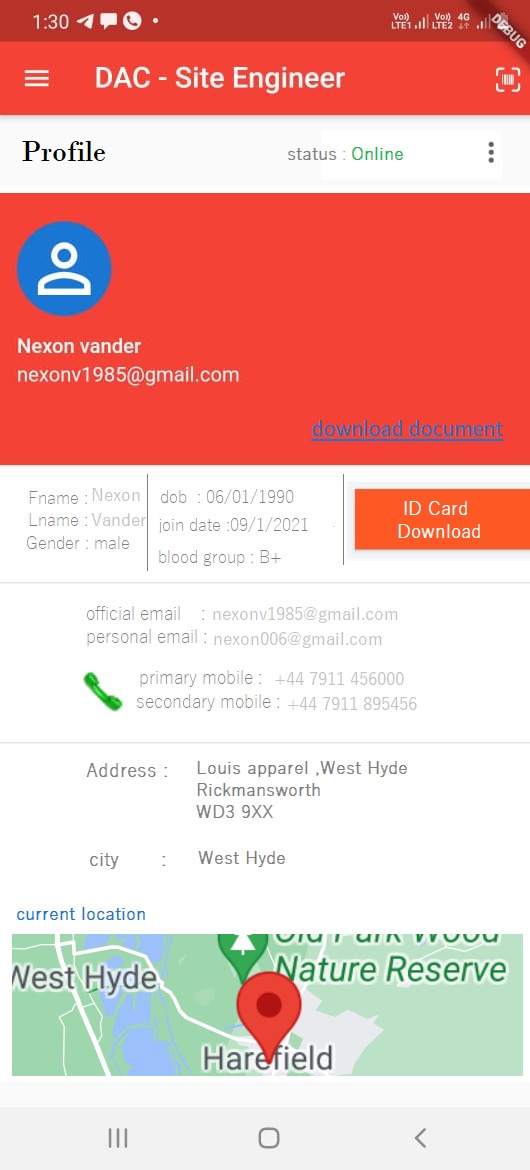
1. **Site Engineer Login**

****

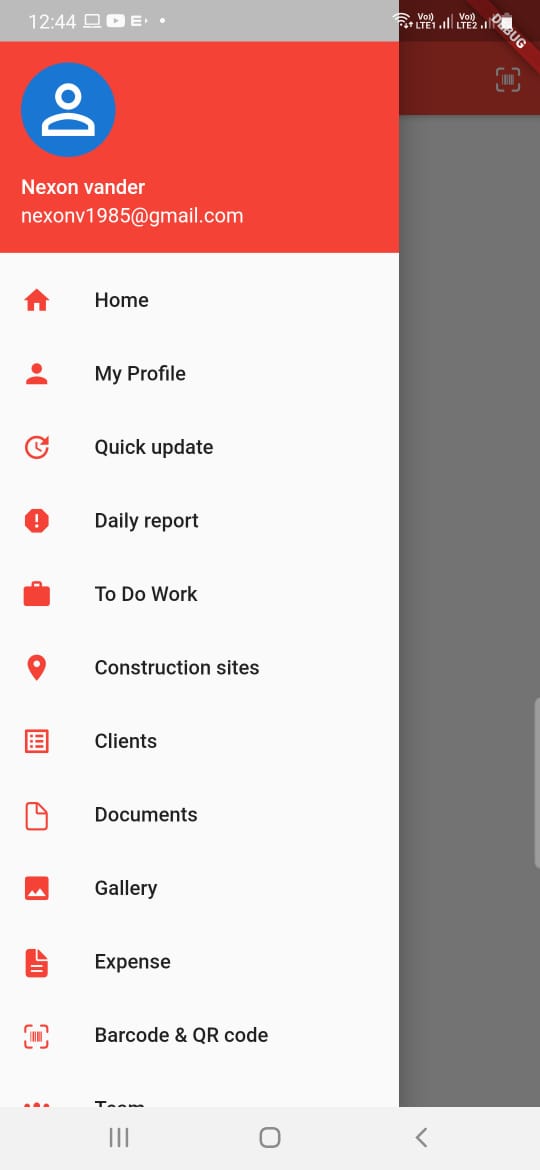
1. **Site engineer dashboard**

****

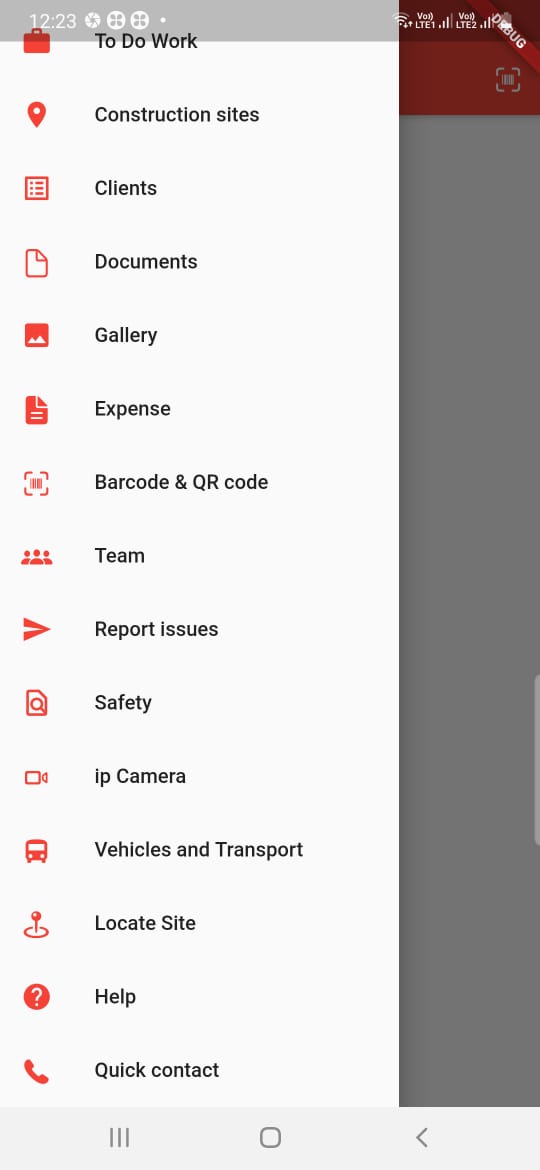
1. **Profile of site engineer**

****

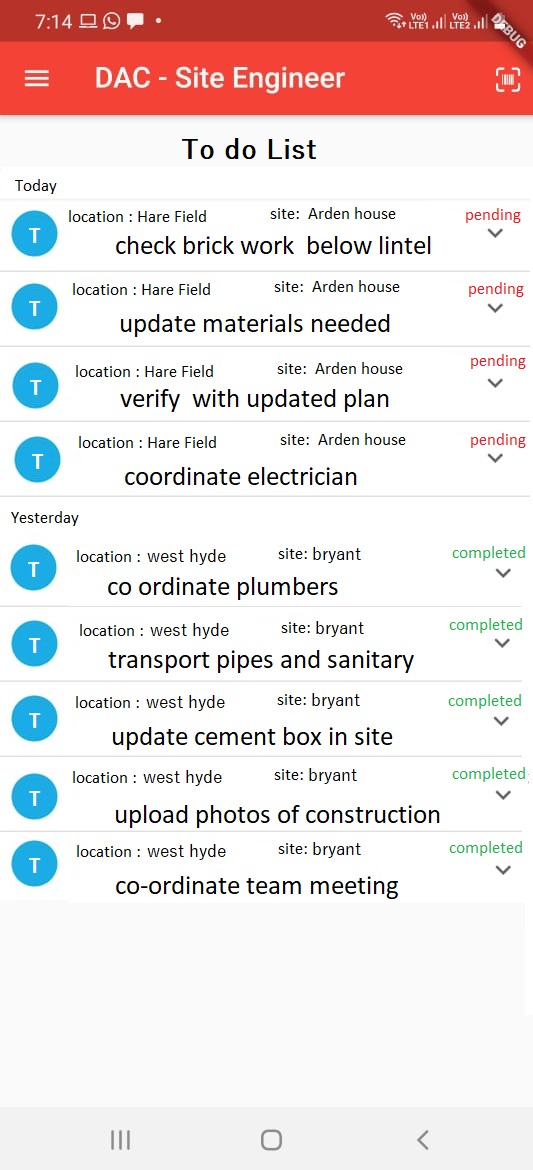
1. **menu**

****

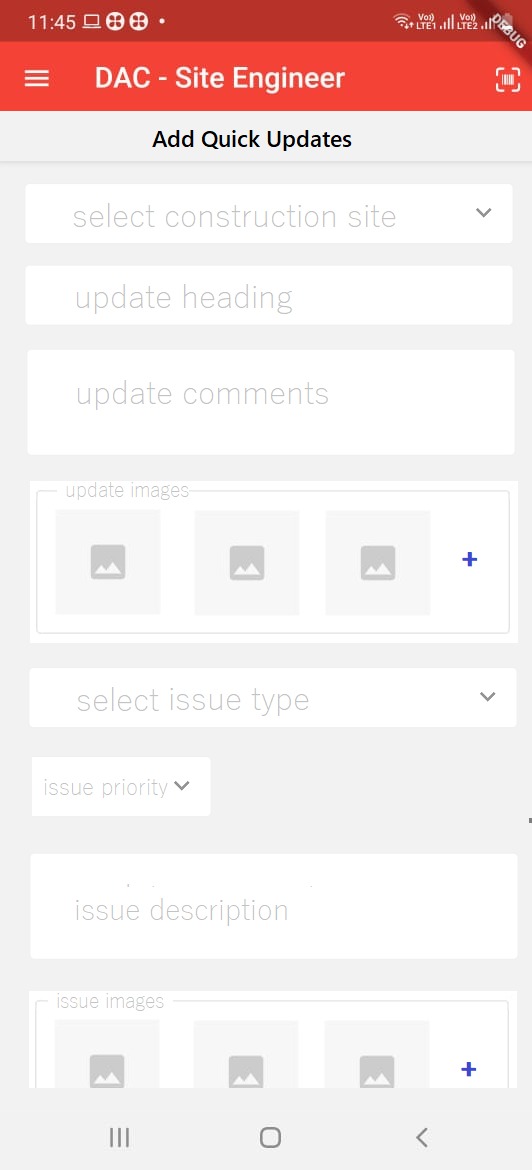
1. **menu**

****

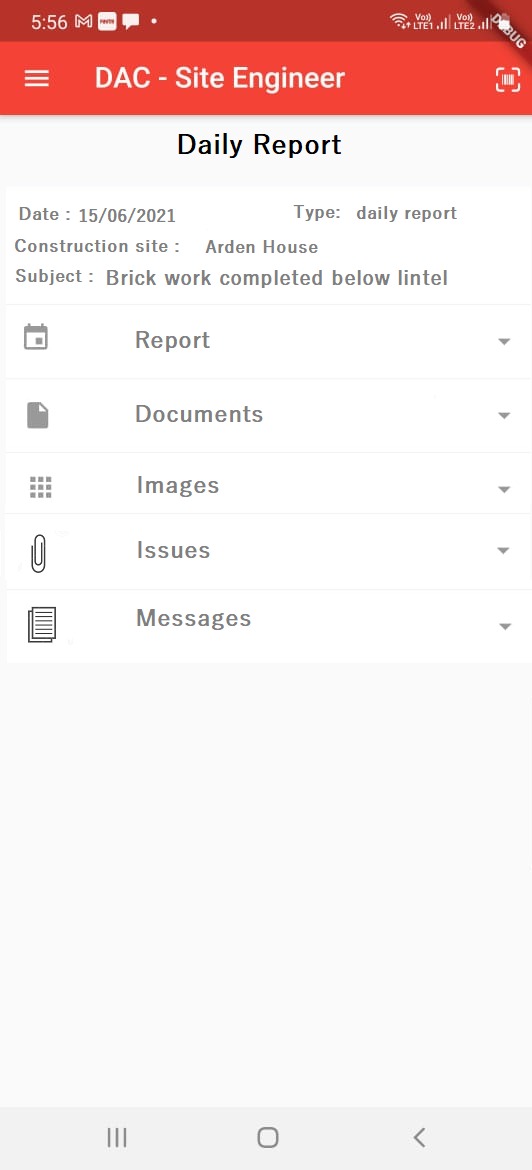
1. **Todo Works**

****

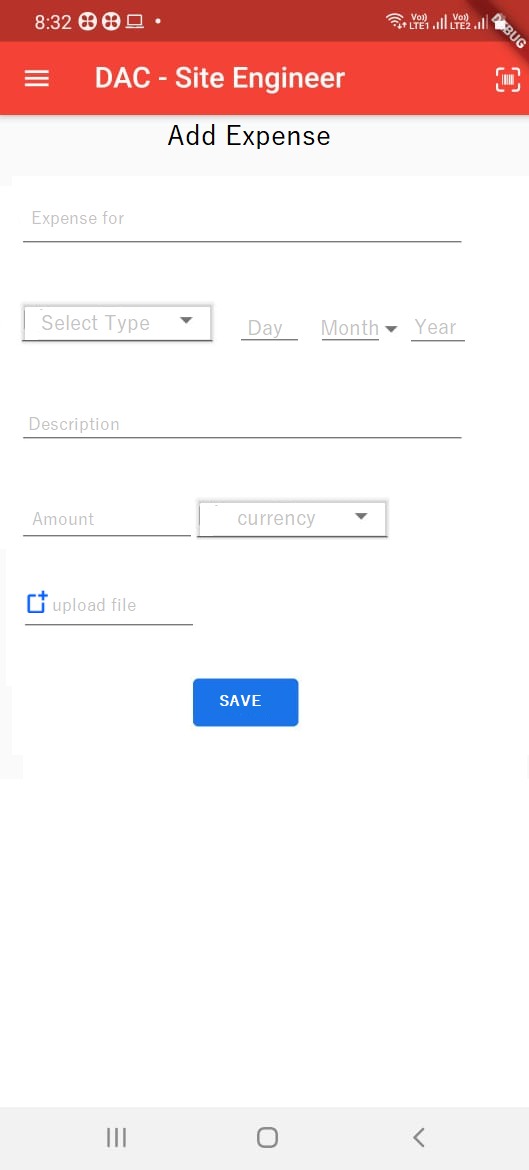
1. **Quick updates**

****

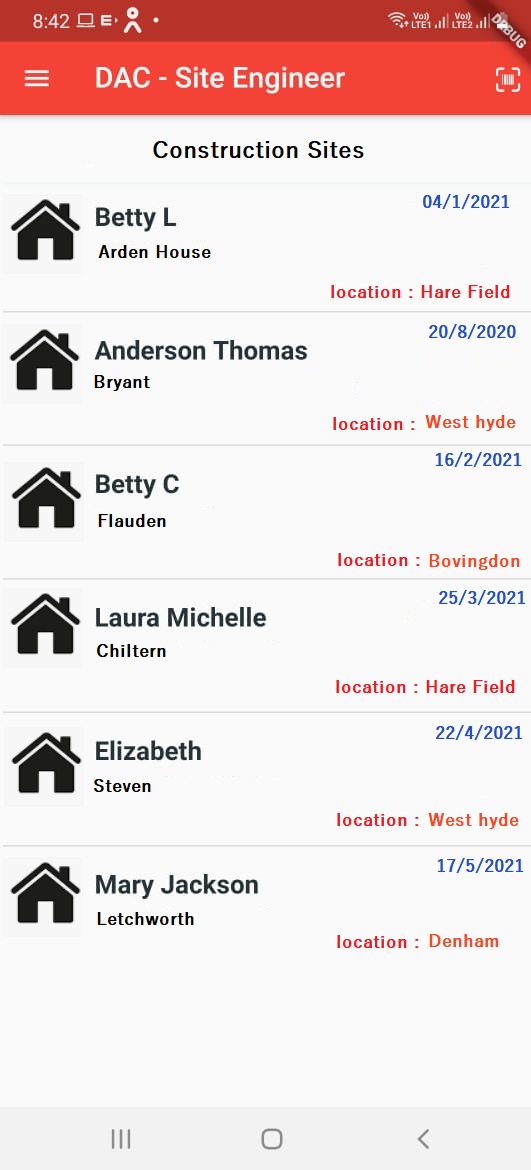
1. **Daily report**

****

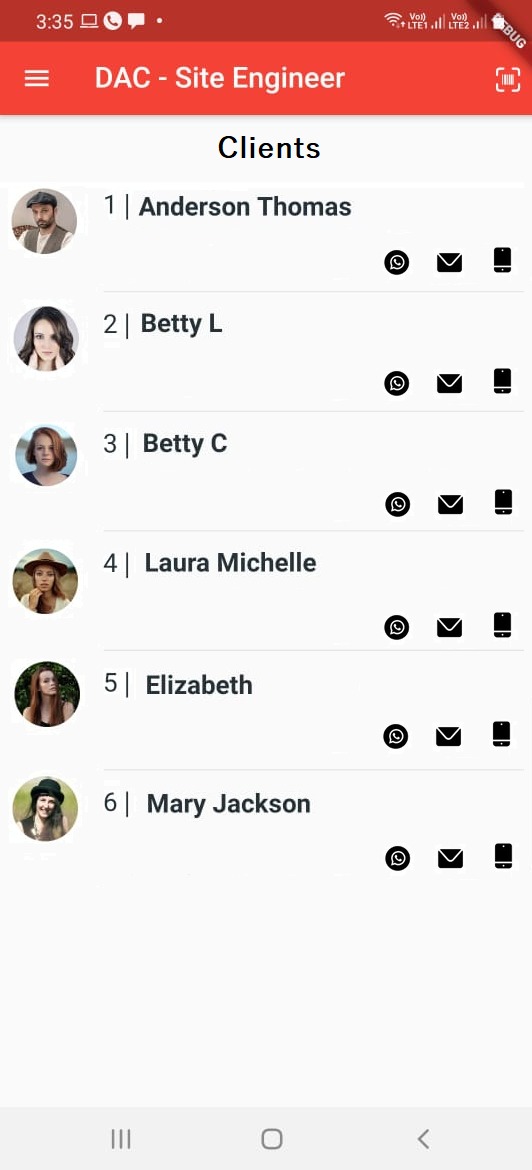
1. **Expenses**

****

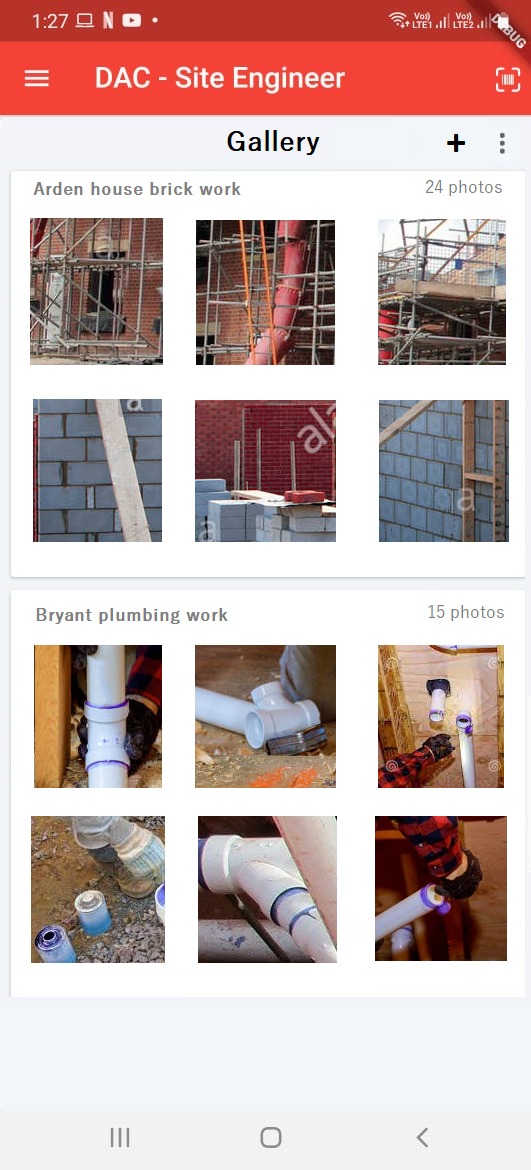
1. **Construction site details**

****

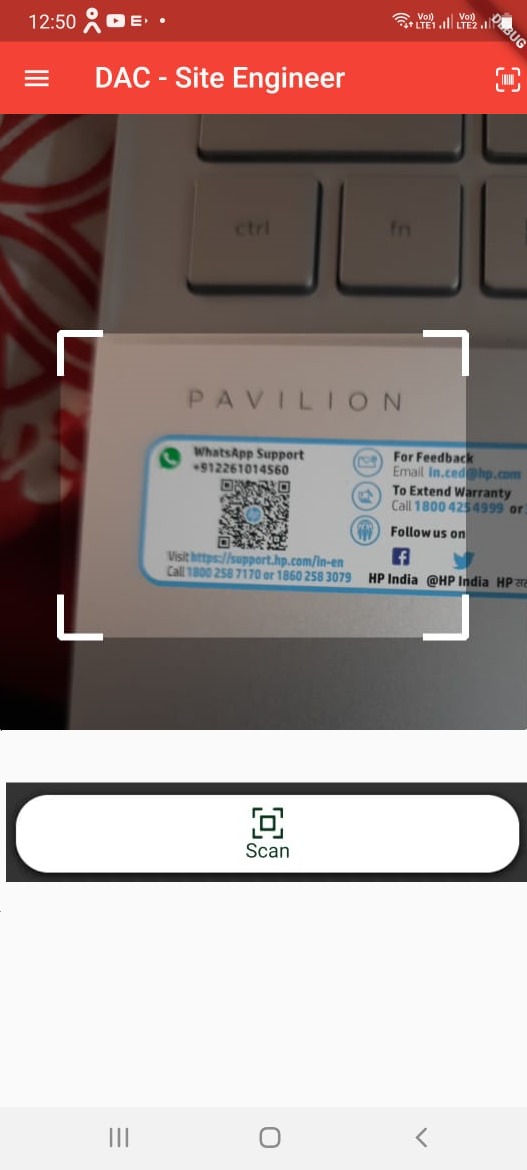
1. **Clients details**

****

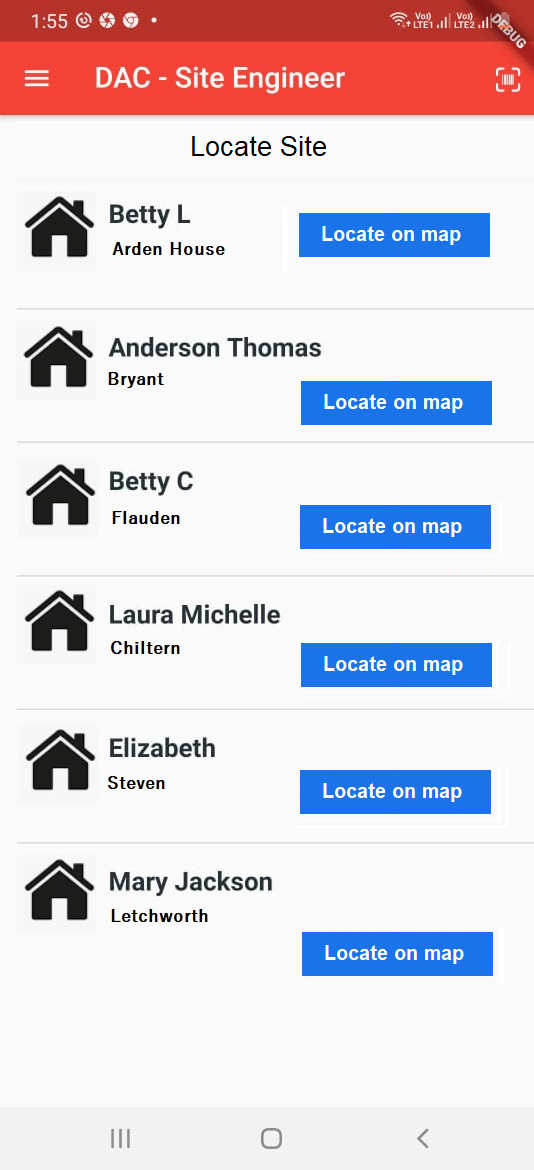
1. **Gallery**

****

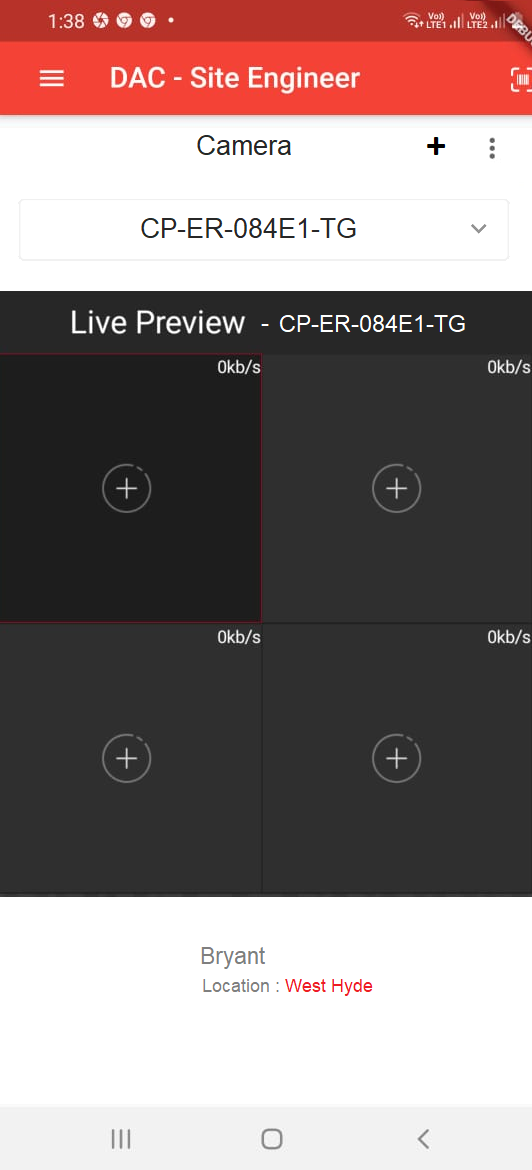
1. **Scanner**

****

1. **locate sites**

****

**(xv) Ip camera**

****

1. **report issue**

**CHAPTER 9**

**9. CONCLUSION**

**9.1 SYSTEM IMPLEMENTATION**

Implementation means converting a new design into iteration .During implementation there should be a strong interaction between the developer of the software and the users. Implementation involves installing hardware terminals and training the operating staff. In this phase, user training is critical for minimizing reluctance to change and giving the new system a chance to prove its worth. The new system may be totally new replacing the existing system, or it may be the modifications of existing system. In either case proper implementation is essential to provide a reliable system to meet organizational requirements.

The implementation involves following things:

* Careful planning.
* Investigation of the system considerations.
* Design the method to achieve the changeover.
* Evaluation of change over method.

Implementation of a new system requires the operating staff installing the software and creating computer files. There are many ways in which this can be achieved. The most common methods are the following.

* Direct change over
* Parallel running
* Pilot running change over The creation of the designed system takes place in the implementation phase.

This phase activities do the following:

• Development of phase overview

• Preparing for implementation

• Computer program development

• Development phase report and overview

It also performs activities like writing, testing, debugging and documenting the programs.

There are three types of implementations:

* Implementation of a computer system to replace a manual system. The problems encountered are converting files, training users, creating accurate files and verifying printouts for integrity.
* Implementation of a new computer system to replace an existing one. This is usually a difficult conversion. If not properly planned, there can be many problems. Some large computer systems have taken as long as a year to convert.
* Implementation of a modified application to replace the existing one, using the same computer. This type of conversion is relatively easy to handle, provided there are no major changes in the files. Every system requires periodic evaluation after implementation. This is to review the performance of the system and to evaluate against established standard or criteria. A study is conducted for measuring the performance of the system against pre-defined requirements. This study results a post-implementation review that determines how well the system continues to meet the performance specification.

The project “**Data Acquisition in Construction Sites with Remote Monitoring** “is tested properly and at the same time, users can be well trained in the new procedure. Proper implementation of the current system meets the requirements. The new application is implemented successfully with proper installation.

**9.2 CONCLUSION**

Considering the globalization of Mobile application we developed an application for site engineer who works on the company.This application “ **Data Acquisition in Construction Sites with Remote Monitoring”** gives process of work monitoring in any Construction Company. the company performs various works at different location at the same time hence all works are recored in manual in the current scenario, hence this application provides such a platform inorder to update all the daily tasks that has been done a site by a site engineer. Due to rapid advancements in technology, the smart phones have proved to be an imperative source of communication and now have become an integral part of our daily lives. In Current system , people are facing problems , To overcome this problem, it’s immensely important to design and develop a platform in order to bridge the communication gap between the technical workers and end-users which can provide an easy and understandable interface for both using the current trends of technology. To serve in this context, an mobile app based system is developed . We have developed an app for the provision for the site engineer to update the work flow of a construction site. Normal Testing of the completed UI is done. Now load testing is conducted for efficacy and functionality of the developed system.

**9.3 FUTURE ENHANCEMENT**

Over the last years the work updates in the construction sites has been recorded manually.Here the application which is developed for the site engineers who works for the construction company can update the work progress of a construction site.Our application comprises of several feathure to update the work flow of the site.As future enchancement of the system we will introduce live video streaming for clients ,while they can see these the visuals of their site. It will lead to improve the communication between the site engineer and the clients and the use of live stream will help the interaction more flexible.Using AI Work flow analysis can be done in the future.It will help the flow of work in the construction site.By introducing new modules that uses machine learing which can be used for material requirement for work,the amount of raw material can be analysed using this feathure. These are few features which can be integrated with this system to make it more flexible in future.

**BIBLIOGRAPHY**

1. **BOOKS**

# [Keith Makan](https://www.amazon.in/s/ref=dp_byline_sr_book_1?ie=UTF8&field-author=Keith+Makan&search-alias=stripbooks) , [Scott Alexander-Bown](https://www.amazon.in/Scott-Alexander-Bown/e/B00H1OX7WG/ref=dp_byline_cont_book_2) , Android Security Cookbook

# [Ryan Cohen](https://www.amazon.in/Ryan-Cohen/e/B00NGHT1T0/ref=dp_byline_cont_ebooks_1), [Tao Wang](https://www.amazon.in/Tao-Wang/e/B00J0LF0BE/ref=dp_byline_cont_ebooks_2) GUI Design for Android Apps 1st ed. Edition

* Bill Philips & Chris Stewart, (2018). Android programming 4th edition published by Big Nerd Ranch Guide
* Leonard Richardson & Sam Ruby, (2007). Restful Web Services published by O ‘Reilly Media

1. **WEBSITES**

* https://www.tutorialspoint.com/restful/restful\_quick\_guide.htm
* https://developer.android.com/training/basics/firstapp
* https://pub.dev/
* https://stackoverflow.com/
* https://www.javatpoint.com
* https://www.wikipedia.com

1. **JOURNAL AND PUBLICATIONS**

* Khalid, H., Nagappan, M., & Hassan, A. E. (2016). Examining the Relationship between FindBugs Warnings and App Ratings. IEEE Software
* Miss. Narmada N. Shahade, “Student Attendance Tracker System in Android”, INTERNATIONAL JOURNAL FOR ENGINEERING APPLICATIONS AND TECHNOLOGY- ISSN: 2321-8134. [5] Ramesh Shrestha, “Design of Secure Location and Message Sharing System for Android Platform”, IEEE-2012 on computer technology.
* Vishwakarma R Ganesh, “Android College Management System” International Journal of Advanced Research in Computer Engineering & Technology, vol. 5, Issue 4, April. 2016, ISSN 2278-1323.

**APPENDICES I**

**1.SCRUM MODEL**

**i.Git**

Git is a version-control system for tracking changes in computer files and coordinating work on those files among multiple people. It is primarily used for source-code management in software development, but it can be used to keep track of changes in any set of files. As a distributed revision-control system, it is aimed at speed, data integrity, and support for distributed, nonlinear workflows.

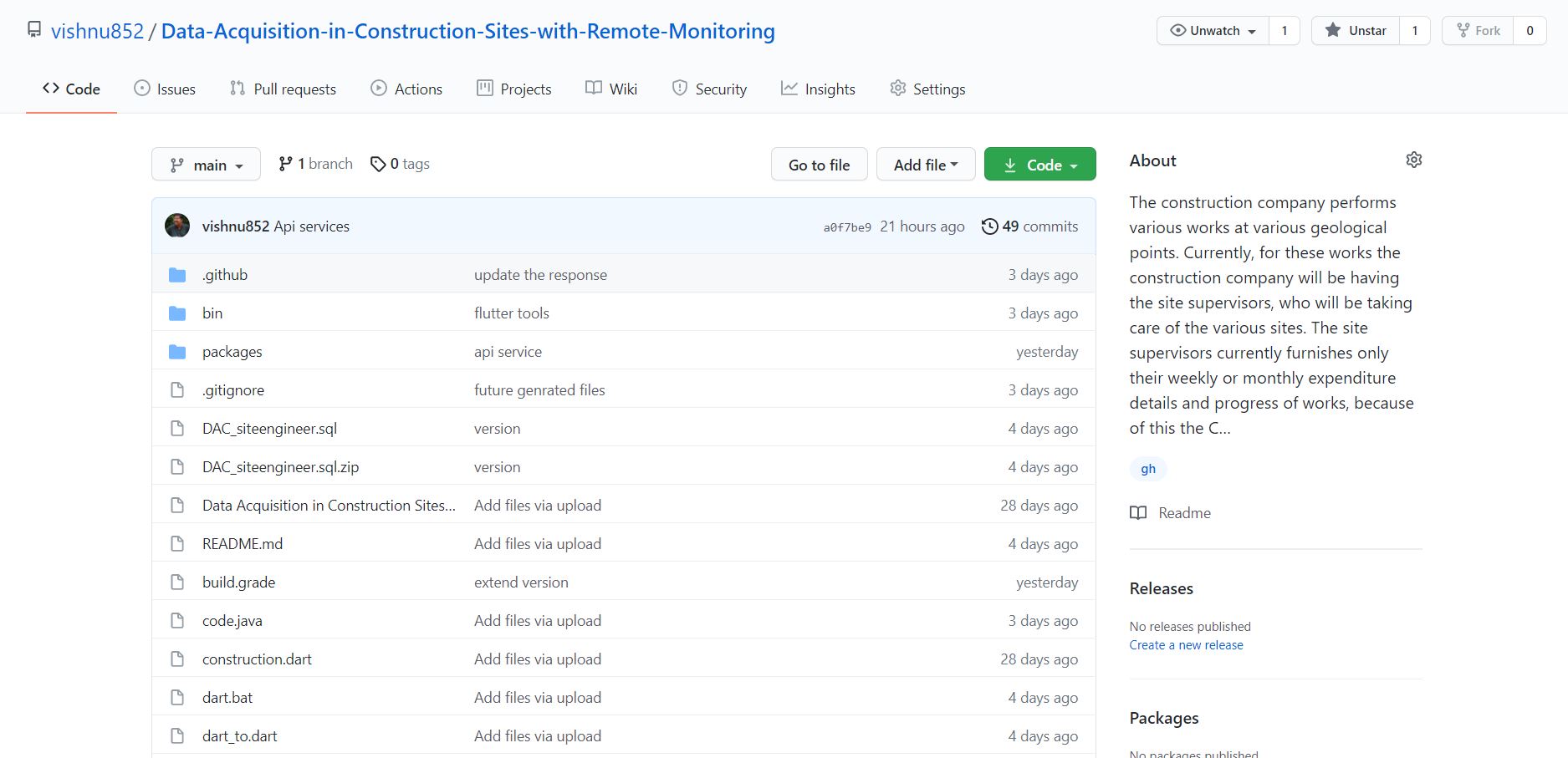
**ii.Git Repositories**

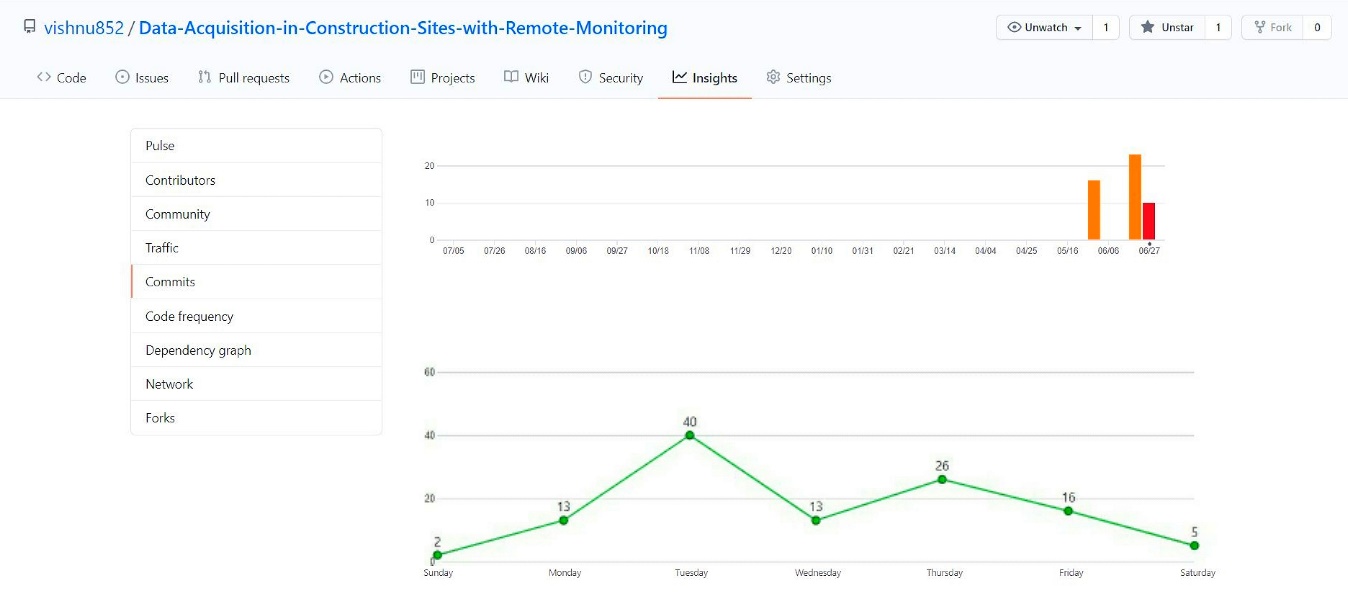
A Git repository contains the history of a collection of files starting from a certain directory. The process of copying an existing Git repository via the Git tooling is called cloning. After cloning a repository the user has the complete repository with its history on his local machine. Of course, Git also supports the creation of new repositories. If you want to delete a Git repository, you can simply delete the folder which contains the repository. If you clone a Git repository, by default, Git assumes that you want to work in this repository as a user. Git also supports the creation of repositories targeting the usage on a server.

**iii.Scrum**

Scrum is an agile way to manage a project, usually software development. Agile software development with Scrum is often perceived as a methodology; but rather than viewing Scrum as methodology, think of it as a framework for managing a process. In the agile Scrum world, instead of providing complete, detailed descriptions of how everything is to be done on a project, 77 much of it is left up to the Scrum software development team. This is because the team will know best how to solve the problem they are presented. In the agile Scrum world, 94 | P a g e instead of providing complete, detailed descriptions of how everything is to be done on a project, much of it is left up to the Scrum software development team. This is because the team will know best how to solve the problem they are presented. Within agile development, Scrum teams are supported by two specific roles. The first is a Scrum Master, who can be thought of as a coach for the team, helping team members use the Scrum process to perform at the highest level. The product owner (PO) is the other role, and in Scrum software development, represents the business, customers or users, and guides the team toward building the right product.

**iv. Git History**

****



**APPENDICES II**

**1. LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **TABLE NO** | **TABLE NAME** | **PAGE NO** |
| 5.1 | User\_login | 36 |
| 5.2 | quick\_update | 36 |
| 5.3 | quick \_images | 37 |
| 5.4 | quick\_issues | 37 |
| 5.5 | quick issue\_images | 38 |
| 5.6 | Todo\_list | 38 |
| 5.7 | Todo\_update | 39 |
| 5.8 | work\_report | 39 |
| 5.9 | Workreport\_images | 40 |
| 5.10 | work report\_doc | 41 |
| 5.11 | Workreport\_issue | 41 |
| 5.12 | workreport\_messaage | 42 |
| 5.13 | construction\_site | 42 |
| 5.14 | site\_doc | 43 |
| 5.15 | client | 44 |
| 5.16 | album\_details | 45 |
| 5.17 | gallery | 45 |
| 5.18 | Site\_expense | 46 |
| 5.19 | Site\_expensedoc | 47 |
| 5.20 | user\_doc | 47 |
| 5.21 | user primary\_details | 48 |
| 5.22 | Input and output design | 51 |
| 7.1 | Testing results | 59 |

**2. LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **TABLE NO** | **TABLE NAME** | **PAGE NO** |
| 5.1 | Dfd components | 27 |
| 5.2 | Context Diagram (Level 0) | 27 |
| 5.3 | Module Description (Level 1) | 28 |
| 5.4 | Profile updates (Level2) | 29 |
| 5.5 | Work Updates( Level 2) | 29 |
| 5.6 | Construction Site Management ( Level 2) | 30 |
| 5.7 | Locate Site Details ( Level 2) | 30 |
| 5.8 | update gallery ( Level 2) | 31 |
| 5.9 | Update expenses ( Level 2) | 31 |
| 5.10 | barcode & Qr code ( Level 2) | 32 |
| 5.11 | Ip Camera ( Level 2) | 32 |
| I | Site Engineer Login | 62 |
| Ii | Site Engineer Dashboard | 63 |
| Iii | Profile Of Site Engineer | 64 |
| Iv | Menu | 65 |
| V | Menu | 66 |
| Vi | Todo Works | 67 |
| Vii | Quick Updates | 68 |
| Viii | Daily Report | 69 |
| Ix | Expenses | 70 |
| X | Construction Site Details | 71 |
| Xi | Clients Details | 72 |
| Xii | Gallery | 73 |
| Xiii | Scanner | 74 |
| Xiv | Locate Sites | 75 |
| Xv | Ip Camera | 76 |
| Xvi | Report Issue | 77 |

**3. ABBREVIATIONS AND NOTATION**

1. **DFD** (Data Flow Diagram) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).
2. **DB** A database is an organized collection of data, generally stored and accessed electronically from a computer system. Where databases are more complex they are often developed using formal design and modeling techniques