TTND BUZZ

*Submitted in partial fulfillment of the requirements*

*for the award of the degree of*

**Master of Computer Application (MCA)**

Guru Gobind Singh Indraprastha University, Delhi

Guide: Submitted by:

Shoibal Chatterji Priya

Enroll No.:01919104413



**Gitarattan International Business School**

**Delhi - 110085**

**Batch (2013-2016)**

**Chapter-1**

Problem Definition

* 1. **Brief Description of the Organization**
  2. **Organization**



Figure No- 1.1: **Logo of Company**

TO THE NEW DIGITAL is a premium digital services company that combines the power of technology, analytics, creative and content for digital transformation. TO THE NEW Digital is also a strategic partner of many global companies including Google, Adobe, Amazon Web Services (AWS), YouTube, DataStax and MongoDB.

The Company was incorporated as a Pvt. limited company at Noida on 2008. It has a team of over 600 people, which includes technology evangelists, social media experts, content specialists, and creative mavericks who have transformed businesses of more than 300 companies spread across 30 countries worldwide.

The Company provides services to enterprises to conceive and develop business critical applications for web, mobile and cloud platforms using new technologies such as Java, Grails, MEAN, Adobe Marketing Cloud and AWS.

The Company also provides Omni channel analytics services that help clients get actionable insights from huge volume of social and enterprise data. It create & run digital marketing campaigns across-platforms that impact the bottom-line and goes beyond 'likes' and 'tweets'.

The resume of services offering is as under:

1. Creative Marketing
2. Social Media Marketing
3. Media Buying & Planning
4. Marketing Technology.
   1. **Vision**

TO THE NEW DIGITAL vision is simple and clear, for every problem there is a solution, it is only the matter how you think and whom you look for the same. We as a company want our client to be our friend and seek TO THE NEW DIGITAL for every solution.

* 1. **General Description of the System Under Study**

There are a large number of social sites like Face book, Twitter etc to share the thoughts or photos but this project is developed for a particular industry or company where only the employee of the organization can post on the site. There are two categories of posts: - Normal Buzz and Complaints. In the normal buzz (post) employee can write blog or can share images with other employee. Employee can use this application for any “lost and found” type post. In Complaints, category employee can send the complaint to the administrator and only the administrator can see the post. The administrator then sends the complaint to the particular department. The employee and the administrator can then keep the track of status of the complaint.

* 1. **User Requirements**

1. Removes overhead in retrieving records from database
2. Serve Clients/Customers—Work to satisfy customer’s expectations.
3. Make Posts online.
4. Customized Retrieval of Records.
5. Tracking of complaints.
6. Keeping information of Users, Posts and Complaints.

**1.4 Establish the Need of the New System**

1.41 Proposed system provides various functionalities like:

* 1. To facilitate easy maintenance of records of various Users posts and complaints details.
  2. To facilitate a good user interface.
  3. To check for details of the lost and found buzz.
  4. Quick access of all record.
  5. Prevent and reduce human error.

**1.5 Objectives of the Proposed System**

1.51 In the fast moving world, if people lack something, it is time. All are busy in their world. It will be welcomed if services are provided at their will.

The main objective is to create a TTND BUZZ by improving and promoting a single, consolidated Application presence for our major lines of Employee, easy-to-use application which helps us to:

1. Share blogs and images among other employee.
2. Make posts online.
3. Make complaints online.
4. User can posts lost and found category of post
5. Tracking status of the complaint online
6. Easily Maintained By Administrator.

vii) User can filter the blogs on the basis of their category.

**1.6 Methodology**

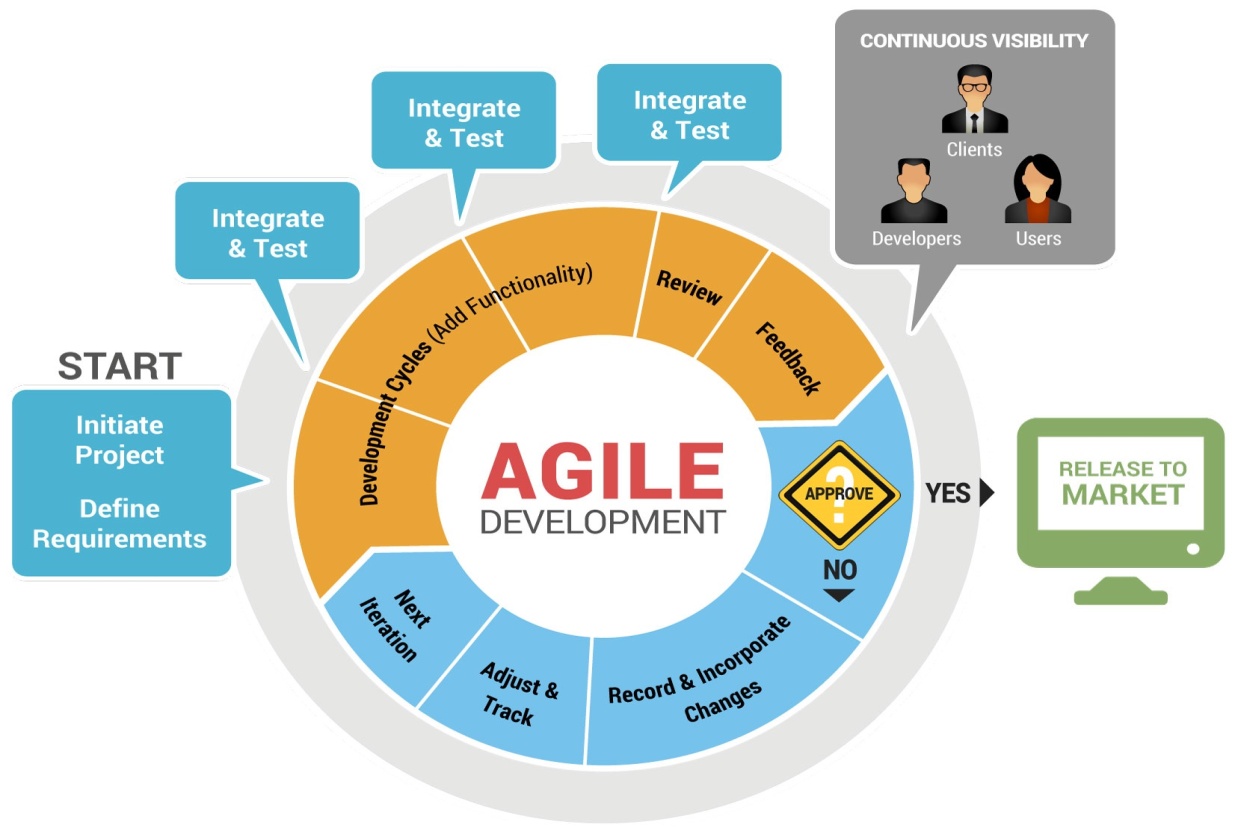
1.61 The entire project will be developed keeping in view a distributed client server computing technology. The user interfaces are application specific to provide easy accessibility of the system. At all levels due care will be taken to check that the system manages data consistency with proper validations. Authentication Mechanism at various levels will be the minimum level of zones to administrator the applications efficiency.

1.62 Analysis Model

The model being followed for this project is the Agile Model.

The Agile is based on twelve principles:

1. Customer satisfaction by early and continuous delivery of valuable software
2. Welcome changing requirements, even in late development
3. Working software is delivered frequently (weeks rather than months)
4. Close, daily cooperation between business people and developers
5. Projects are built around motivated individuals, who should be trusted
6. Face-to-face conversation is the best form of communication (co-location)
7. Working software is the principal measure of progress
8. Sustainable development, able to maintain a constant pace
9. Continuous attention to technical excellence and good design
10. Simplicity—the art of maximizing the amount of work not done—is essential
11. Best architectures, requirements, and designs emerge from self-organizing teams



**Figure 1.2: Agile model approach**

**1.7 Data Required and Data Collection Method**

In this project, for developing application data is required in respect of input, output, storage and processing requirements. Data is also required in respect of entities involved in the system & their connectivity/linkages and flow of data between the entities. The data has been collected through the following methods:

* 1. **Internet** Data collected for proposed system is from internet, by searching and

analyzing various social networking application like Twitter, Face book.

* 1. **Interviews** The real requirements actually reside in user’s mind. Hence existing system is assessed to collect the information and data by arranging Interview. In interviews afterreceiving problem statement from the employee, the first step is to arrange a meeting with the real user (employee). During meeting both the parties interact with each other and discuss about the requirements. Developer’s main focus will be on the user requirements, he will try to read as much as user’s mind. It is an informal kind of meeting where both parties are free to ask any thing to each other regarding project.
  2. **Brainstorming** technique is a group techniquethat may be used during data collection to understand the requirements. The group discussions may lead to new ideas quickly and help to promote creative thinking. Requirements in the long list can be categorized, prioritized and pruned. Every idea will be documented such a way that everyone can see it. Finally a final document will be prepared which will have list of requirements and their priority.
  3. **Facilitated Application Specification Technique** Similar to brainstorming sessions. Fast is a Team oriented approach. FAST involves creation of joint team of customers and developers. Hence this technique is used to bridge the gap between what the developers thinks they are supposed to build and what customers think they are going to get. Each FAST attendee is asked to make a list of objects that are as

1. Part of environment that surrounds the system.
2. Produced by the system.
3. Used by the system.
4. List of constraints
5. Performance criteria

**1.8 Project Planning**

* 1. When creating a software project schedule, the planner begins with a set of tasks (the work break down structure). Efforts, duration, and start date are then input for each task. In addition, task may be assigned to specific individuals. As a sequence of this input, a timeline chart, also called a Gantt chart, is generated. A Gantt chart can be developed for the entire project. Alternatively, separated it depicts a part of a software project schedule that emphasizes the concept scooping task for a new word processing software project. All project tasks are listed in the left hand column. The horizontal bars occur at the same time on the calendar, task concurrency is implied.

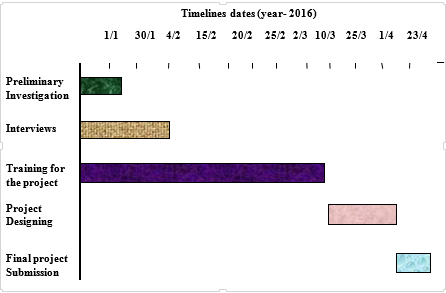


Figure No- 1.3: **Gantt Chart**

**Chapter-2**

**Systems Requirement Specifications**

1. **Introduction**
   1. A Software Requirement Specification (SRS) document is generated as the output of requirements analysis. Requirements analysis involves obtaining a clear and thorough understanding of the project to be developed. Thus, the SRS should be a consistent, correct, unambiguous and complete document. The developer of the system can prepare an SRS after detailed communication with the customer.

* 1. An SRS contains External interfaces of the system that identify the information that is to flow ‘from and to’ to the system. In addition it also contains functional and non-functional requirements of the system that are the findings of the run-time requirements. The functional requirementsof the system as documented in the SRS document should clearly describe each function, which the system would support along with the corresponding input and output data set. The non-functional requirementsdeal with the characteristics of the system that cannot be expressed as functions. Examples of non-functional requirements include aspects concerning maintainability, portability, and usability

* 1. **Scope of the Development Project**

**“TTND BUZZ”** is a web-based project which is made for is implemented in MEAN stack. Main aim of this project is to create a online blogging applicable like Google blogs …etc. Using this application users can create website with the tag name of blogger in this software you can find all features that are present in existing blogging software’s. Users can design modify by adding new templates and add posts. Initially users need to register with application like in Google blog sites we log with Gmail user id similarly user need to get unique user id and password. These posts which are posted in blogger site will be available in results and users who visit blog can like or dislike posts.

* 1. **Users of the System**

1. Employee
   1. **Hardware Requirements**
2. RAM 512MB and Above
3. Internal Memory 4 GB Hard Disk Space and Above
   1. **Software Requirements**
4. Windows or Linux OS
5. Front End : AngularJs, Javascript.
6. Back End  : Mongodb
7. Server side: NodeJs, Express.
   1. **Identify the Process**

There are different processes used in this project that covers user’s requirements.

* 1. **Employee Module**

1. Employee Login
2. Blog
3. Post the blogs
4. View blogs posted by other employee
5. Like and dislike the posts.
6. Email Verify
7. Preview of the blog
8. Upload images
9. Complaint
10. Post the Complaint
11. Preview of the complaint
12. Upload images
13. View all complaints
14. Complaint status
15. Lost and Found posts
16. Administrator
    1. **Input and Output of the Processes**
    2. **Module**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Employee Login | Login Form | Report retrieved from database |
|  | Make Posts | Hyperlink | Post View |
|  | Complaints | Hyperlink | Complaint View |
|  | Lost and found | Hyperlink | Lost and Found View |
|  | Email Verify | Entry | Verification Code |
|  | Logout | Hyperlink | Successful Logout from System |
|  | Store Manager | Selection | Database |

**Table No**- **2.1: Input/output of Each Process in Customer Module**

* 1. **Identify the Data Elements (Fields) in Each Input and Output**
  2. **Customer Module**

1. Employee Login
2. User Name (Textbox)
3. Password (Textbox)
4. Blogs
5. Heading (Textbox)
6. Content (Textbox)
7. Preview (Checkbox)
8. Type of buzz (Selection)
9. Upload image (Button)
10. Send post. (Button)
11. Complaints

i) Heading (Textbox)

ii) Content (Textbox)

iii) Preview (Checkbox)

iv) Upload image (Button)

v) Send complaint. (Button)

* 1. **Identify the Procedures/Rules/Mathematical Relationships Used for Converting Input into Output**
  2. **Employee Login:** It will authenticate the user to be an employee of the organization.
  3. **Blog:** It will manage the Employee posts. User can post the blog in the provided space and can view the post in view panel.
  4. **Complaints:** This module maintains the complaints of the user. User can post a complaint and this complaint is further processed by the other departments.
  5. **Administrator:** Administrator can maintain the status of the complaint.
  6. **Identify the Controls (For I/O & Access), Security Needs, Validation Rules and Codes Used for Data Elements**
  7. **Customer Module**

1. **Security Needs**
2. The User on the app should be verified by their Gmail.
3. The system will provide the relevant information depending on the query searched.
4. The password provided should be in encrypted format.
5. **Validation Rules**

A validation rule is a criterion used in the process of data validation, carried out after the data has been encoded onto an input medium and involves a data set or validation program.

1. For Login process, the user should be the employee of the organization.
2. User cannot post without giving a heading and content of the post.
3. All the entries should be filled; no entry should be left blank.
4. For login customer have to provide correct user name and password otherwise access cannot be granted.
5. Provide secure password with special characters.

**Chapter-3**

**Systems Design**

1. **Introduction**

After the analysis phase we have with us the details of the existing system and the requirements of the user for the new system. This phase diverts focus from the problem domain to the solution domain. It acts as a bridge between the requirement phase and its solution. The design phase focuses on the detailed implementation of the system recommended in the feasibility study. Systems design is the process or art of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering. Object-oriented analysis and design (OOAD) methods are becoming the most widely used methods for computer system design. The UML has become the standard language used in Object-oriented analysis and design. It is widely used for modeling software systems and is increasingly used for high designing non-software systems and organizations.

1. **Levels of Designs**
   1. **External Design**

External design consists of conceiving, planning out and specifying the externally observable characteristics of the software product. These characteristics include user displays or user interface forms and the report formats, external data sources and the functional characteristics, performance requirements etc. External design begins during the analysis phase and continues into the design phase.

* 1. **Logical Design**

The logical design of a system pertains to an abstract representation of the data flows, inputs and outputs of the system. This is often conducted via modelling, which involves a simplistic (and sometimes graphical) representation of an actual system. In the context of systems design, modelling can undertake the following forms, including:

1. Data Flow Diagrams
2. Entity Relationship Diagrams
   1. **Physical Design**

The physical design relates to the actual input and output processes of the system. This is laid down in terms of how data is input into a system, how it is verified/authenticated, how it is processed, and how it is displayed as output. Physical design, in this context, does not refer to the tangible physical design of an information system. To use an analogy, a personal computer's physical design involves input via a keyboard, processing within the CPU, and output via a monitor, printer, etc. It would not concern the actual layout of the tangible hardware, which for a PC would be a monitor, CPU, motherboard, hard drive, modems, video/graphics cards, USB slots, etc.

* 1. **Usecase Diagram** in designing an efficient and effective system it is important to consider usecase diagram. Usecase diagram is central to modeling the behavior of a system, subsystem or a class.

**Use case Diagram**

Login

Change status

Logout

Assign complaints

To other employee

Can check logs complaints

Make complaints

Can see the posts

ADMIN

IN

USER

Can make posts

Like or dislike posts

Figure No- 3.1:**Usecase Diagram**

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems. Use case diagrams model the functionality of a system using actors and use cases. Use cases are services or functions provided by the system to its users.

* 1. **Data Flow Diagram** is used to define the flow of the system and its resources such as information’s. DFDs are a way of expressing systems requirements in graphical manner. DFD represents one of the most ingenious tools used for structured analysis. It has the purpose of clarifying system requirements and identifying major transformations that will become programs in the system design. It is the major starting point in the design phase that functionalities decompose the requirement specification to the lowest level of detail. A DFD shows what kinds of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored.

1. **Level 0 (Context-Level)**

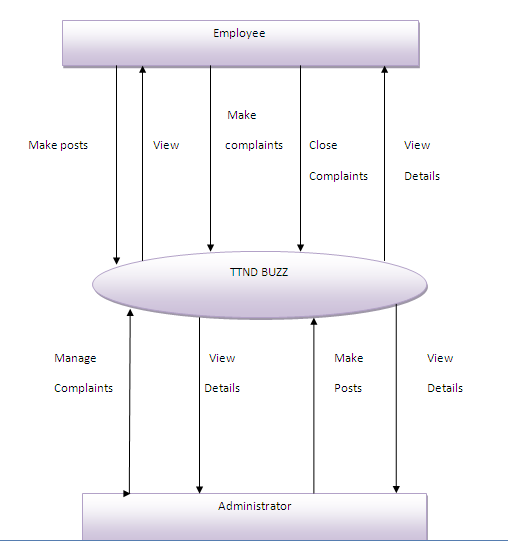
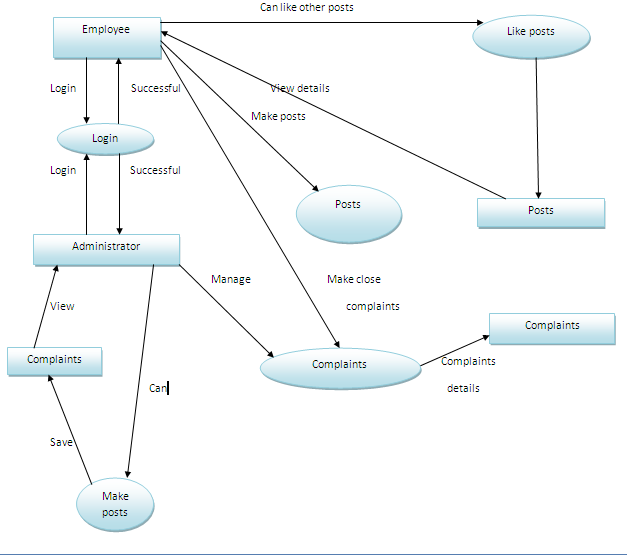


Figure No- 3.2: Zero Level Data Flow **Diagram**

b) **Level 1 DFD**

 Figure No- 3.3: One Level Data Flow **Diagram**

* 1. **ER Diagram**

An ERD is a logical representation of an organization’s data, and consists of three primary components. An entity-relationship model (ERM) is an abstract and conceptual representation of data. Entity-relationship modeling is a database modeling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion. Diagrams created by this process are called entity-relationship diagrams, ER diagrams, or ERDs. The primary purpose of the ERD is to represent data objects and their relationships.

1. Entity Relationship (ER) diagrams are drawn when designing a database system, After the systems specification, an ER diagram is drawn showing the conceptual design of the database, this diagram shows the type of information that is to be stored in the system and how these information associate with each other (e.g. one-to-one, one-to-many, etc.).
2. **Entity:**An entity may be defined as a thing which is recognized as being capable of an independent existence and which can be uniquely identified. By composite information,   
   **c) Attributes:** Attributes define the properties of a data object and take on one of three different characteristics.
3. **Relationship:** A relationship captures how two or more entities are related to one another.
4. **Cardinality:**The data model must be capable of representing the number of occurrences of objects in a given relationship. The cardinality of an object-relationship pair is
5. **One-to-One (1:1):** An occurrence of object ‘A’ can relate to one and only one occurrence of object ‘B’ and an occurrence of ‘B’ can relate to only one occurrence of ‘A’.
6. **Many-to-Many (M: N):**An occurrence of object ‘A’ can relate to one or more occurrences of ‘B’, while an occurrence of ‘B’ can relate to or more occurrences of ‘A’.
7. **One-to-Many (1: N):** One occurrence of object ‘A’ can relate to one or many occurrences of object ‘B’ but an occurrence of ‘B’ can relate to only one occurrence of ‘A’

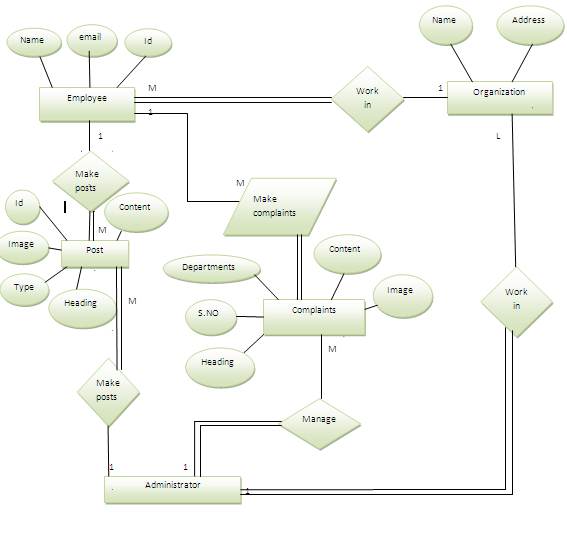


Figure No- 3.3: **ER Diagram**

* 1. **Flow Chart**

A flowchart is a type of diagram that represents an algorithm, workflow or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows. This diagrammatic representation illustrates a solution model to a given problem. Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields.

1. **Admin level**

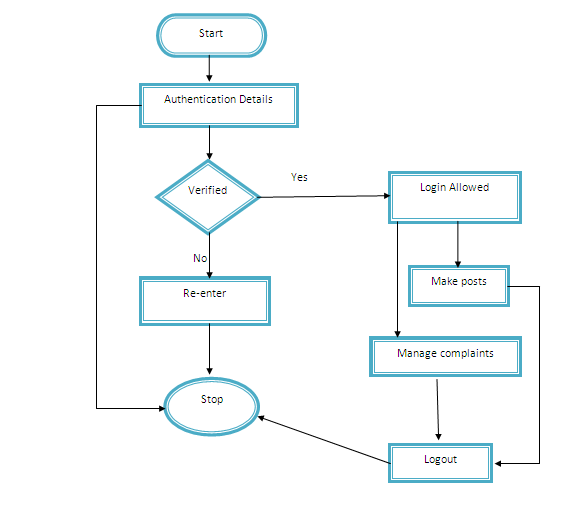


Figure No- 3.4 (a):**Flow Chart (Administrator)**

1. **Employee level**

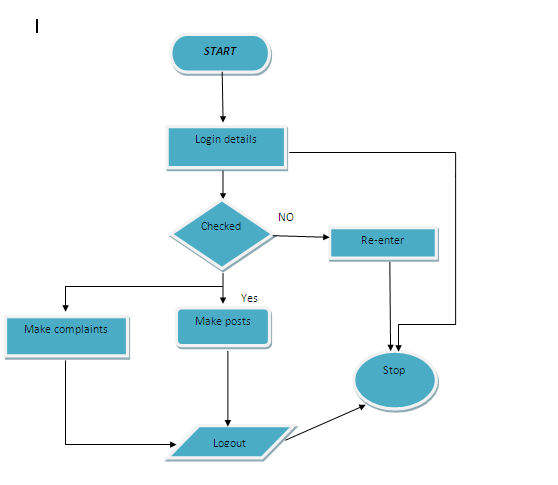


Figure No- 3.4 (b):**Flow Chart (Employee)**

1. **Interface Design**

**3.31 Output Design**

1. **Login screen**

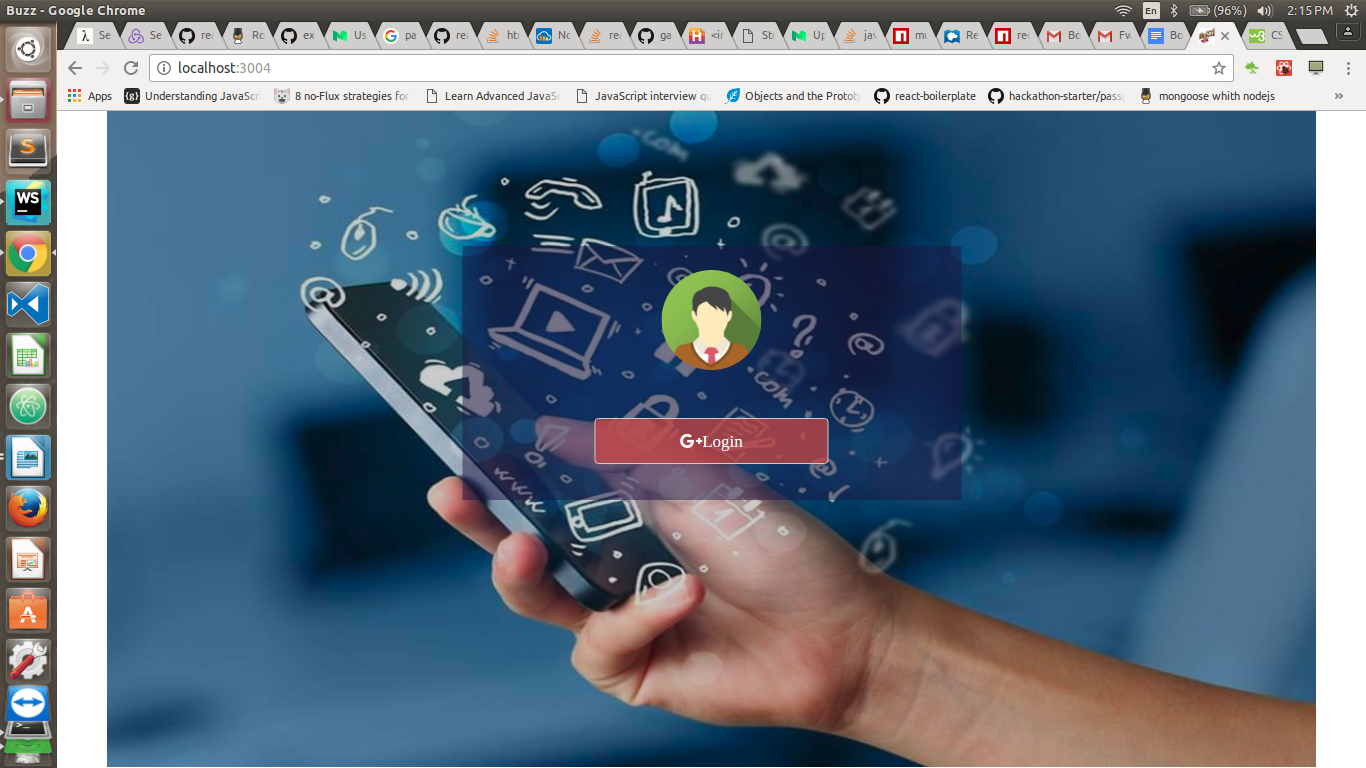
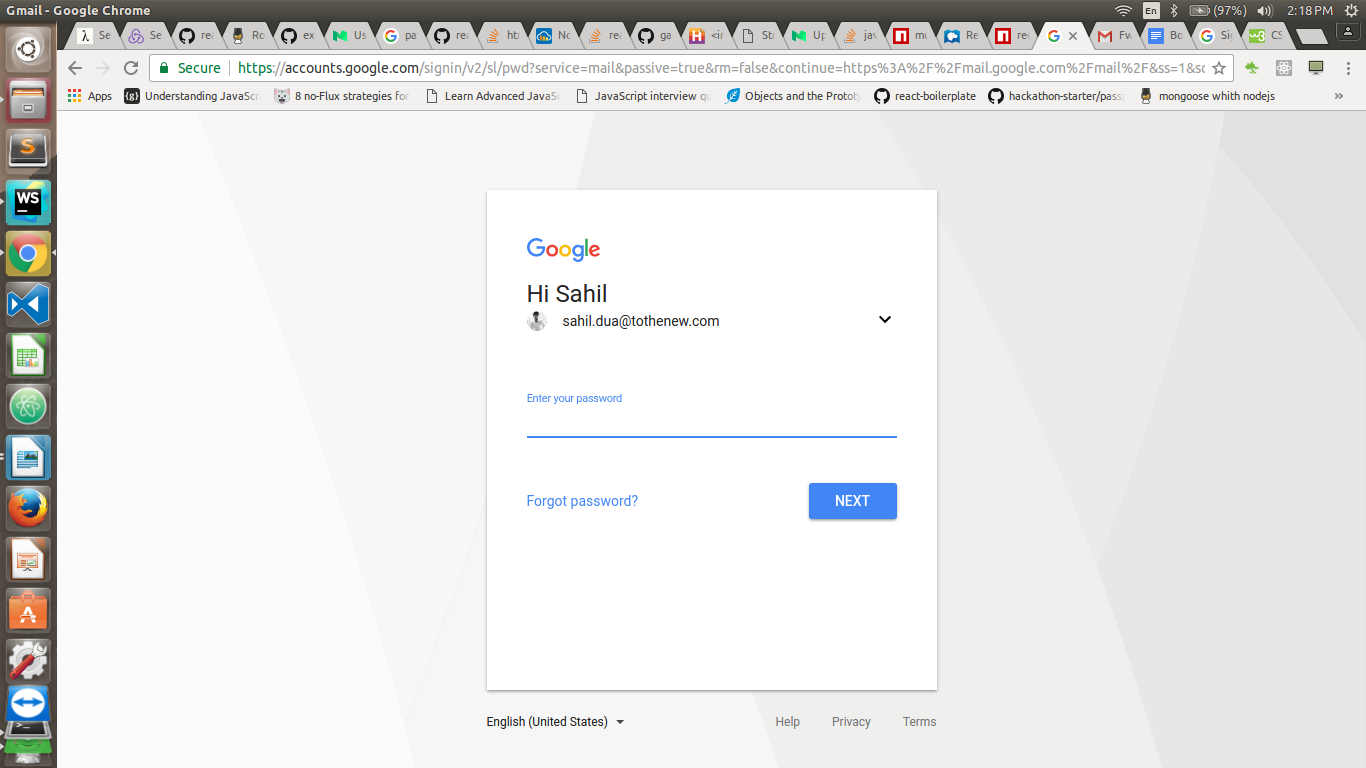
Figure No- 3.4(a): **Login Screen**

Figure No- 3.4(b): **Login Screen**

****

1. **Post Screen:**

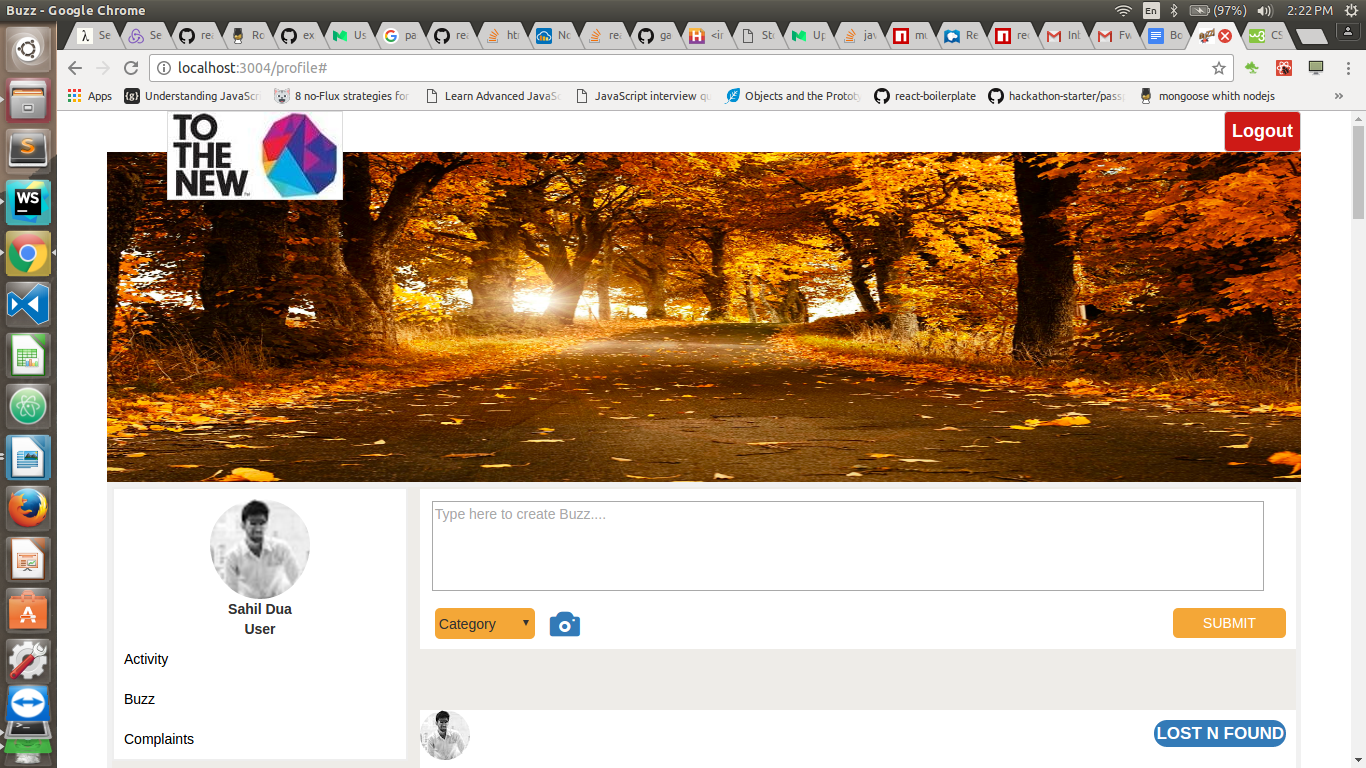


Figure No- 3.4(c): **Posts output**

1. **Lost and Found Filter Screen:**

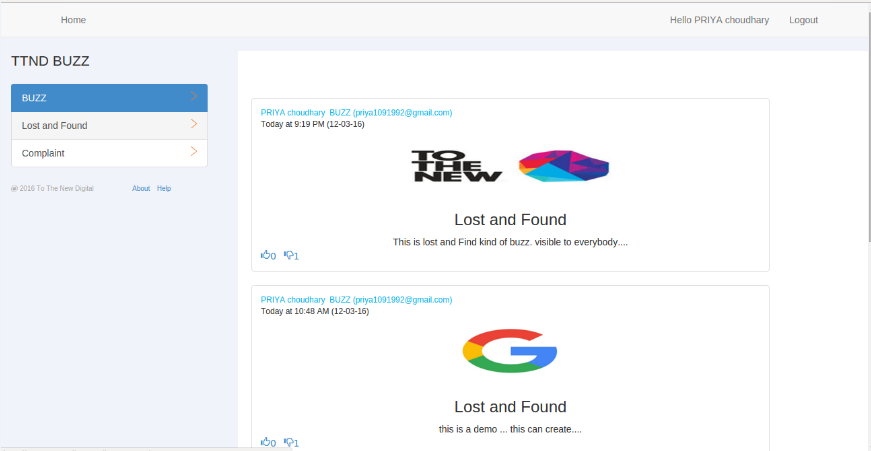


Figure No- 3.4(d): **All Lost and Found Posts output screen**

1. **Employee Complaint Screen**

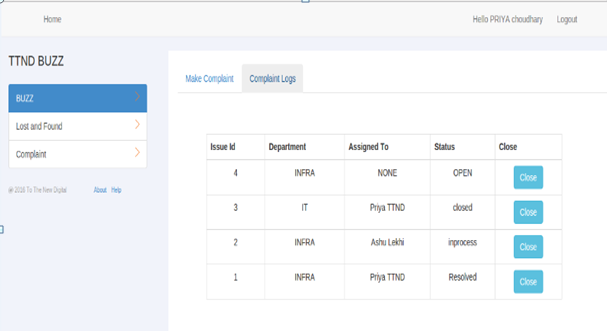


Figure No- 3.4(e): **Complaint lists output screen**

**3.32 Input Design**

1. **Buzz screen**

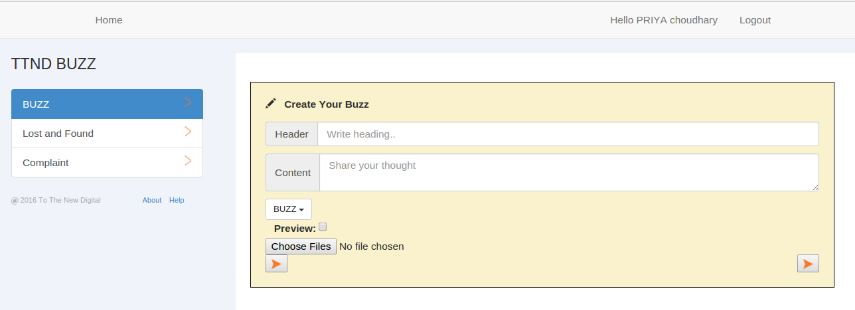


Figure No- 3.4(f): **Post screen**

1. **Complaint screen**

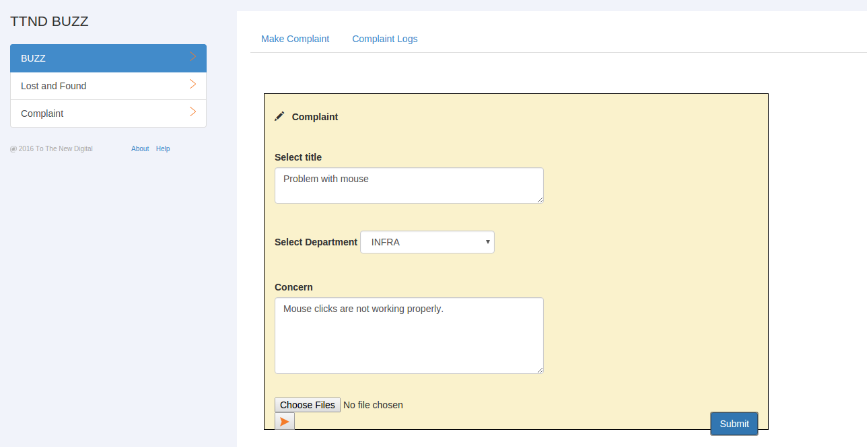


Figure No- 3.4(g): **Complaint screen**

1. **Administrator Complaint Manage screen**

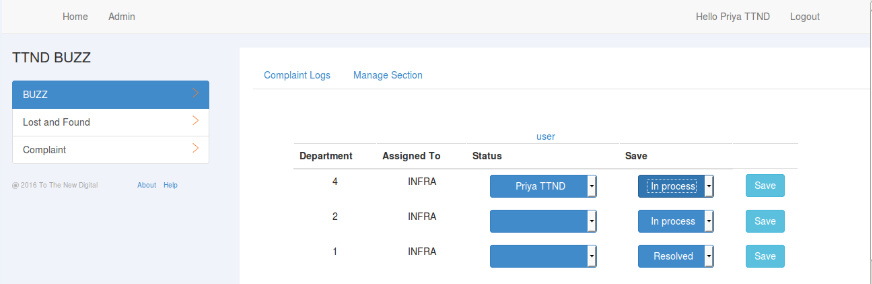


Figure No- 3.4(h): **Complaint manage screen**

1. **Employee Complaint Log screen**

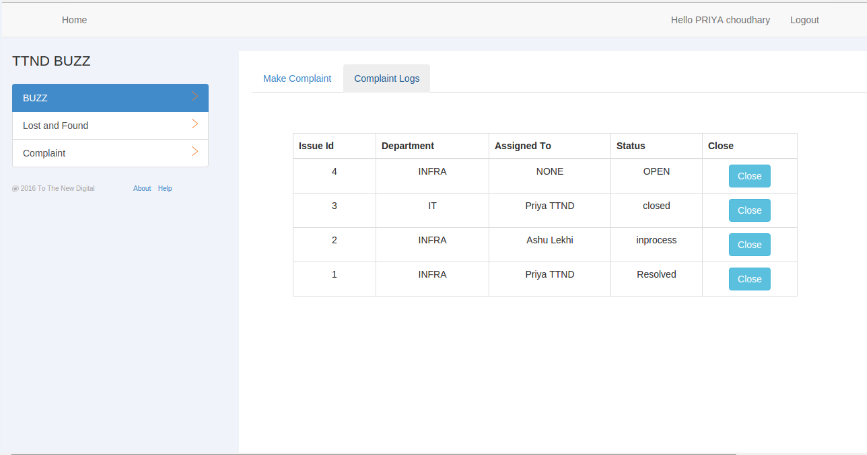


Figure No- 3.4(i): **Complaint screen**

**Database and File Design**

**3.41** Database design is used to define and specify the structure of objects used in the system. A wide array of design information must be developed during the database design. A database is the collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective of data base design is to make information access easy, quick, inexpensive, and flexible for the user.

**3.42 Database Files/Schema Design**

1. **Database File** This is main file that encompasses the entire database and that is saved to your hard-drive or floppy disk.
2. **Table** A table is a collection of data about a specific topic. There can be multiple tables in a database. In Mongodb tables are known as collections.
3. **Document**: Documents are the different categories within a Collection. Collections usually contain multiple Documents.
4. **Data Types**: Data types are the properties of each field. A field only has 1 data type. E.g. Fieldname: Student Last-name, Data type: Text

**3.43.1 User Collection:** This table is used to store the information of all the users (Employee and Administrator).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **No.** | | | **Field Name** | **Range of valid values for the field** | | **Remarks** | |
|  | 1 | | | ID | String | | This is the primary key field of the collection to give unique id to user | |
|  | 2 | | | Name | String. | | This field stores the name of the user. | |
|  | 3 | | | Email | String | | It stores the email of the user. | |
| 4 | | | Role | | | String | | It stores the role of a user. Whether employee or admin. |
|  |  |  | | | | | | |

**Table No – 3.5(a): Add user**

**3.43.2 Complaint Collection**: This table is used to store all the information of employee complaints .

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Field Name** | **Range of valid values for the field** | **Remarks** |
| 1 | ID | String | This is the primary key field of the collection. |
| 2 | Status | String | It stores the status of the complaint( Pending, In-process, Open, Closed). |
| 3 | Issue\_date | Date | It stores the date of the complaint. |
| 4 | Department | String | It stores the department to which the complaint is fired. |
| 5 | Issue\_id | Number | It is the unique id of all the complaints. |
| 6 | Title | String | It stores the title of the complaint. |
| 7 | Concern | String | It stores the complaint concern. |
| 8 | User | String | It stores the user name who makes the complaint. |
| 9 | Image\_url | String | It stores the image URL that is attached with the complaint. |
| 10 | Assigned\_to | String | It stores the name of the employee who handles the complaint. |
| 11 | Email | String | This is the email is of the employee who makes the complaint. |

**Table No – 3.5(b): Complaint collection**

**3.43.3 Post Collection:** This collection is used to store the posts data.

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Field Name** | **Range of valid values for the field** | **Remarks** |
| 1 | ID | String | This is the primary key field of the database as it is unique for all Posts. |
| 2 | Category | String | It stores the category of the post(Lost and found, Buzz). |
| 3 | Date\_Created | Date | It stores the date of the post. |
| 4 | Header | String | It stores the heading of the post. |
| 5 | Content | String | It stores the content of the post. |
| 6 | Media\_url | String | It stores the image url of the post |
| 7 | Name | String | It stores the name of the user who post the information. |
| 8 | Email | String | It stores the email of the user. |
| 9 | Likes | String | It stores the likes on a particular post. |
| 10 | Dislikes | String | It stores the dislikes on a particular post. |

**Table No – 3.5(c):Post Collection**

**Chapter-4**

**Systems Development**

1. **Purpose**
2. System Development is a conceptual model used in Project Management that describes the stages involved in an information system development project, from an initial feasibility study through maintenance of the complete application.
3. The different phases of system development activities viz., feasibility study, systems design, program development, procedures and forms development, acceptance testing, conversion, operation and maintenance and post-implementation review.
4. Systems development is the process of defining, designing, testing, and implementing a new software application or program. It could include the internal development of customized systems, the creation of database systems, or the acquisition of third party developed software.
5. Written standards and procedures must guide all information systems processing functions. The organization’s management must define and implement standards and adopt an appropriate system development life cycle methodology governing the process of developing, acquiring, implementing, and maintaining computerized information systems and related technology.
6. A general systems development or project management framework defines the scope and boundaries of managing projects, as well as the SDLC or project management methodology to be adopted and applied.
7. **Program Development**

Program development is the creation of formal model of the problem to be solved in the system analysis. The Front End is AngularJS , Back End is MongoDB and Server is Nodejs .

1. **AngularJS**
2. AngularJS is open source, completely free, and used by thousands of developers around the world.
3. It is licensed under the Apache license version 2.0.Build variants and multiple APK file generation.
4. Application written in AngularJS is cross-browser compliant. AngularJS automatically handles JavaScript AngularJS is a very powerful JavaScript Framework.
5. It is used in Single Page Application (SPA) projects.
6. It extends HTML DOM with additional attributes and makes it more responsive to user actions.
7. code suitable for each browser.
8. **NodeJS**
9. In [software development](https://en.wikipedia.org/wiki/Software_development), Node.js is an [open-source](https://en.wikipedia.org/wiki/Open-source_software), [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) [runtime environment](https://en.wikipedia.org/wiki/Runtime_system) for developing [server-side](https://en.wikipedia.org/wiki/Server-side) [Web applications](https://en.wikipedia.org/wiki/Web_application).
10. The runtime environment interprets JavaScript using [Google](https://en.wikipedia.org/wiki/Google)'s [V8](https://en.wikipedia.org/wiki/V8_(JavaScript_engine)) JavaScript engine.
11. Node.js has an [event-driven architecture](https://en.wikipedia.org/wiki/Event-driven_architecture) capable of [asynchronous I/O](https://en.wikipedia.org/wiki/Asynchronous_I/O). These design choices aim to optimize [throughput](https://en.wikipedia.org/wiki/Throughput) and [scalability](https://en.wikipedia.org/wiki/Scalability) in [Web applications](https://en.wikipedia.org/wiki/Web_application) with many input/output operations, as well as for [real-time Web](https://en.wikipedia.org/wiki/Real-time_Web) applications (e.g., [real-time communication](https://en.wikipedia.org/wiki/Real-time_communication) programs and [browser games](https://en.wikipedia.org/wiki/Browser_game)).

**4.23 MongoDB**

a) MongoDB (from *hu****mongo****us*) is a [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) [document-oriented database](https://en.wikipedia.org/wiki/Document-oriented_database).

b) Classified as a [NoSQL](https://en.wikipedia.org/wiki/NoSQL) database, MongoDB eschews the traditional table-based [relational database](https://en.wikipedia.org/wiki/Relational_database) structure in favor of [JSON](https://en.wikipedia.org/wiki/JSON)-like documents with dynamic [schemas](https://en.wikipedia.org/wiki/Database_schema) (MongoDB calls the format [BSON](https://en.wikipedia.org/wiki/BSON)), making the integration of data in certain types of applications easier and faster.

c) MongoDB is developed by [MongoDB Inc.](https://en.wikipedia.org/wiki/MongoDB_Inc.) and is published as [free and open-source software](https://en.wikipedia.org/wiki/Free_and_open_source_software) under a combination of the [GNU Affero General Public License](https://en.wikipedia.org/wiki/GNU_Affero_General_Public_License) and the [Apache License](https://en.wikipedia.org/wiki/Apache_License).

1. **Testing and Debugging**

**Software testing** is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include, but are not limited to the process of executing a program or application with the intent of finding software bugs (errors or other defects). Software testing can be stated as the process of validating and verifying that a computer program/application/product:

1. Testing is the process of running a system with the intention of finding errors.
2. Testing enhances the integrity of a system by detecting deviations in design.
3. Testing aims at detecting error-prone areas.
4. Testing also add value to the product by confirming to the user requirements.
5. **Debugging**

**Debugging** is a methodical process of finding and reducing the number of bugs, or defects, in a computer program or a piece of electronic hardware, thus making it behave as expected. Debugging tends to be harder when various subsystems are tightly coupled, as changes in one may cause bugs to emerge in another. The debug information that Remote Debug can provide includes the following:

1. Stack and function traces in error messages with:
2. Full parameter display for user defined functions
3. Function name, file name and line indications
4. Support for member functions
5. Memory allocation
6. Protection for infinite recursions
7. **Module Testing**

Module testing is the testing of complete code objects as produced by the compiler when built from source. Each test case is independent from the others. Substitutes such as method stubs, mock objects, fakes, and test harnesses can be used to assist testing a module in isolation. Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended. Benefits of Module Testing:-

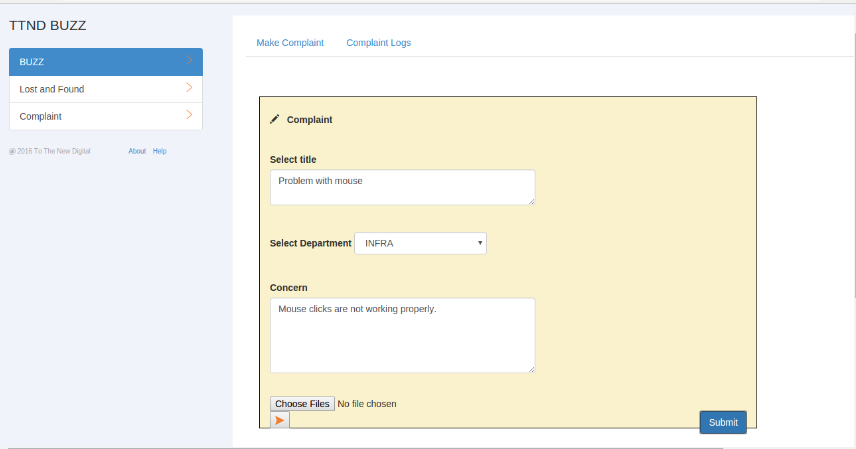
1. **Find Error Early**: - Unit tests find problems early in the development cycle.
2. **Facilitates Change**: - Unit testing allows the programmer to refractor code at a later date, and make sure the module still works correctly (e.g., in regression testing). The procedure is to write test cases for all functions and methods so that whenever a change causes a fault, it can be quickly identified.
3. **Simplifies Integration**: - Unit testing may reduce uncertainty in the units themselves and can be used in a bottom-up testing style approach. By testing the parts of a program first and then testing the sum of its parts, integration testing becomes much easier.
4. **Documentation**: - Unit testing provides a sort of living documentation of the system. Developers looking to learn what functionality is provided by a unit and how to use it can look at the unit tests to gain a basic understanding of the unit's interface (API).
5. **System Testing**

**System Testing** of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic. As a rule, system testing takes, as its input, all of the "integrated" software components that have passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called assemblages) or between any of the assemblages and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole. System testing is performed on the entire system in the context of a Functional Requirement Specification(s) (FRS) and/or a System Requirement Specification (SRS). System testing tests not only the design, but also the behavior and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specification. Types of tests to include in system testing. The following examples are different types of testing that should be considered during System testing:

1. **Graphical User Interface Testing**: - is the process of testing a product's graphical user interface to ensure it meets its written specifications. This is normally done through the use of a variety of test cases.
2. **Usability Testing: -** is a technique used in user-centered interaction design to evaluate a product by testing it on users. This can be seen as an irreplaceable usability practice, since it gives direct input on how real users use the system.
3. **Performance Testing: -** is in general testing performed to determine how a system performs in terms of responsiveness and stability under a particular workload. It can also serve to investigate measure, validate or verify other quality attributes of the system, such as scalability, reliability and resource usage.
4. **Compatibility Testing**: - part of software non-functional tests, is testing conducted on the application to evaluate the application's compatibility with the computing environment.
5. **Scalability Testing**: - Part of the battery of non-functional tests, is the testing of a software application for measuring its capability to scale up or scale out in terms of any of its non-functional capability.
6. **Regression Testing** is a type of software testing that seeks to uncover new software bugs, or regressions, in existing functional and non-functional areas of a system after changes such as enhancements, patches or configuration changes, have been made to them.
7. **Validation**

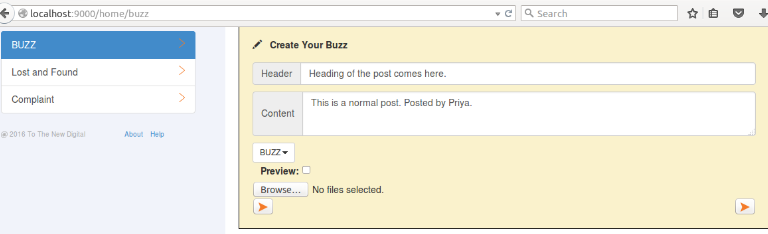
A **Validation rule** is a criterion or constraint used in the process of data validation, carried out after the data has been encoded onto an input medium and involves a data vet or validation program. This is distinct from formal verification, where the operation of a program is determined to be that which was intended, and that meets the purpose. The method is to check that data falls the appropriate parameters defined by the systems analyst. A judgment as to whether data is valid is made possible by the validation program, but it cannot ensure complete accuracy.

1. **Required Field Validation :** Field ‘Concern’ and ‘Department’ are necessary**.**



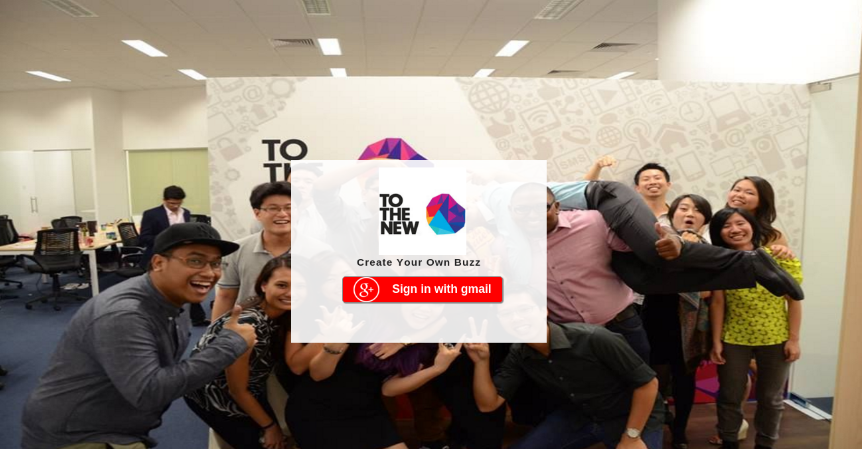
### Figure No- 4.1: **Required Field Validation**

1. **Required Field Validation:** Field ‘Header’ and ‘Content’ are necessary.



### Figure No- 4.2: **Required Field Validation**

1. **Validation on Login**: User can only login through Gmail account.



### Figure No- 4.3: **Validation on Login**

1. **Validation on Login**: User can only login through Mail Id of ‘To The New Digital’ mail.

### C:\Users\00819104414\Desktop\Capture.PNG

### Figure No- 4.4: **Validation on Login**

**Chapter-5**

**System Implementation**

1. **Introduction**

Implementation uses the design document to produce code. Demonstration that the program satisfies its specifications validates the code. Typically, sample runs of the program demonstrating the behavior for expected data values and boundary values are required. Small programs are written using the model, as programs get more complicated, testing and debugging alone may not be enough to produce reliable code. Instead, we have to write programs in a manner that will help insure that errors are caught or avoided.

1. **Acquisition**

Top down implementation begins with the user-invoked module and works toward the modules that do not call any other modules. The implementation may precede depth-first or breadth-first.

1. **Hardware Specification**

Hardware is the set of components associated with computer system. Hardware is the essential complement to software. It is the computer backbone which when “Web Application” has been developed using the following tools:-

|  |  |
| --- | --- |
| **HARDWARE** | **SPECIFICATIONS** |
| CPU | Intel DualCore or Higher |
| Speed | 1.2 GHz |
| Ram | 1 GB |
| Hard Disk | 4 GB |

**Table No**- **5.1: Hardware Specification**

1. **Software Specification**

Software is the set of programs, documents and procedures, routines associated with computer system. Software is the essential complement to hardware. It is the computer programs which when “Job Portal” has been developed using the following tools:-

|  |  |
| --- | --- |
| **SOFTWARE** | **SPECIFICATIONS** |
| Operating System | Android |
| Front End | AngularJS, Javascript |
| Back End | MongoDB |
| Server | NodeJS |

**Table No**- **5.2: Software Specification**

1. **Conversion**

Conversion is the activity that transforms the stored data in the existing information system into the format required for use in the new or changed information system. Other activities may also be done during conversion. Conversions can be very simple or very complex. The final activity within design is implementing the system, which involves installing and getting the users to use the new or changed information system. There are three types of conversion methods:-

1. **Direct**

In this type of conversion, input from the old system is transferred to the new system. When the transfer is completed, the organization switches over to using the new system. One disadvantage is that there is no phasing-in period to test out the new system Valuable accounting and cost information could be lost.

1. **Parallel**

In the parallel conversion method, the organization uses the old system in parallel with the new system for a period of time until it becomes comfortable with the new system. The organization then totally switches over to the new system and no longer uses the old one.

1. **Partial**

It can allow you to implement the system at in stages or modules. In this method implementation of each subsystem can done by using other three changeover methods. The risk of failure is limited up to the implementation stage. It is a less expensive than the parallel method. It is not possible if the system cannot be separated easily into logical modules or segments. If the system contains large number of the phases, phase operation cost is more than a pilot approach.

1. **Training Needs**

The activity of training personnel and operators is a necessary part of the implementation of the system. There are two types of training methods:-

1. **Vendor and In-Service Training: -** Often the best source of training on equipment is the vendor who supplies the equipment. Most vendors offer extensive educational programs as part of their services.
2. **In-House Training: -** The main advantage of in-house training is that instructions cab be tailored according to the requirements of the organization. Often the vendors negotiate fees and charges that are more economical if organization involves more personnel in the training program compared to the cost of travelling to the vendor’s location.
3. **Training Of Personnel And Operators**

A well designed system can succeed or fail because of the way they are operated and used. The quality of training received by personnel involved with the system in various capacities help or hinders and may even prevent the successful implementation of management information system. Those who are directly or indirectly related with the system development work must know in detail what their roles will be, how they can make efficient use of the system and what the system will not do for them. Both systems operators and other staff needs training.

1. **System Operators Training:-** Running of the system successfully depends on the personnel working in the computer centers. They are responsible for providing the necessary support. Their training must ensure that they are able to handle all possible operations, both routine and extra ordinary in nature.
2. **User Training: -** Users may be trained on the use of equipment, particularly in the case where, for example, a micro-computer is in use and the individual involved is both an operator and a user. In such cases, uses must be given training how to use the system also. Questions that may be trivial to the analyst, such as how to turn on the terminal, how to insert a diskette into a micro-computer ,or when is it safe to turn off equipment without danger of data loss, are significant to the new users who are not familiar with the computers.
3. **Documentation**

Documentation describes an information system and helps the users, managers, and IT staff who must interact with it. It includes program documentation, system documentation, operations documentation, and user documentation.

1. **Operation Documentation:-**Operations documentation contains all the information such as:-
2. Program, systems analyst, programmer, and system identification.
3. Scheduling information for printed output, such as report run frequency.
4. Input files and where they originate; and output files and destinations.
5. Special forms required, including online forms.
6. Error and informational messages to operators and restart procedures.
7. **User Documentation:-**User documentation, Written or other visual information consists of instructions and information to users who will interact with the system. User Operations documentation contains all the information such as:-
8. List of **minimum hardware and software** required to use the system.
9. How to **install** the system.
10. How to **start/stop** the system.
11. How to **use the features** of the system.
12. **Screenshots** showing the system in typical use.
13. Example of **inputs and outputs.**
14. Explanations of any **error messages** that might be shown.

**Chapter-6**

**Summary and Conclusions**

**Summary**

There are a large number of social sites like Face book, Twitter etc to share the thoughts or photos but this project is developed for a particular industry or company where only the employee of the organization can post on the site. There are two categories of posts: - Normal Buzz and Complaints. In the normal buzz (post) employee can write blog or can share images with other employee. Employee can use this application for any “lost and found” type post. In Complaints, category employee can send the complaint to the administrator and only the administrator can see the post. The administrator then sends the complaint to the particular department. The employee and the administrator can then keep the track of status of the complaint.

In the fast moving world, if people lack something, it is time. All are busy in their world. It will be welcomed if services are provided at their will.

The main objective is to create a TTND BUZZ by improving and promoting a single, consolidated Application presence for our major lines of Employee, easy-to-use application which helps us to:

1. Share blogs and images among other employee.
2. Make posts online.
3. Make complaints online.
4. User can posts lost and found category of post
5. Tracking status of the complaint online
6. Easily Maintained By Administrator.

vii) User can filter the blogs on the basis of their category.

**Design Work**

**Data Flow Diagram** is used to define the flow of the system and its resources such as information’s. DFDs are a way of expressing systems requirements in graphical manner. DFD represents one of the most ingenious tools used for structured analysis. It has the purpose of clarifying system requirements and identifying major transformations that will become programs in the system design. It is the major starting point in the design phase that functionalities decompose the requirement specification to the lowest level of detail. A DFD shows what kinds of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. A Data Flow Diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. DFDs can also be used for the visualization of data processing (structured design).DFD were useful to document the major data flows or to explore a new high-level design in terms of data flow.

**A physical DFD** shows how the system is actually implemented, either at the moment (Current Physical DFD), or how the designer intends it to be in the future (Required Physical DFD). Thus, a Physical DFD may be used to describe the set of data items that appear on each piece of paper that move around an office, and the fact that a particular set of pieces of paper are stored together in a filing cabinet. It is quite possible that a Physical DFD will include references to data that are duplicated, or redundant, and that the data stores, if implemented as a set of database tables, would constitute an un-normalized (or de-normalized) relational database. In contrast, a Logical DFD attempts to capture the data flow aspects of a system in a form that has neither redundancy nor duplication.

**System Developed**

There are two categories of users (stakeholder) who can use application

1. **Employer Module:** Required functionality for employer module.
2. Login
3. Post the blogs
4. View blogs posted by other employee
5. Like and dislike the posts.
6. Preview of the blog
7. Upload images
8. Post the Complaint
9. Preview of the complaint
10. Upload images in complaint section
11. View all complaints
12. Check complaint status
13. **Administrator** **Module:** Required Functionality for Administrator Module.
14. Login
15. Post the blogs
16. View blogs posted by other employee
17. Like and dislike the posts.
18. Preview of the blog
19. Upload images
20. Manage the Compliant of the employee
21. Assign the employee complaint to the particular user.
22. Administrator can change the status of the complaint.

.

**Testing**

Module testing is the testing of complete code objects as produced by the compiler when built from source. Each test case is independent from the others. Substitutes such as method stubs, mock objects, fakes, and test harnesses can be used to assist testing a module in isolation. Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended. Benefits of Module Testing are.

1. **Find Error Early**: - Unit tests find problems early in the development cycle.
2. **Facilitates change**: - Unit testing allows the programmer to refractor code at a later date, and make sure the module still works correctly (e.g., in regression testing). The procedure is to write test cases for all functions and methods so that whenever a change causes a fault, it can be quickly identified.
3. **Simplifies integration**: - Unit testing may reduce uncertainty in the units themselves and can be used in a bottom-up testing style approach. By testing the parts of a program first and then testing the sum of its parts, integration testing becomes much easier.
4. **Documentation**: - Unit testing provides a sort of living documentation of the system. Developers looking to learn what functionality is provided by a unit and how to use it can look at the unit tests to gain a basic understanding of the unit's interface (API).

**Conclusions**

The development of software includes so many people like user system developer, user of system and the management. Proper design builds upon this foundation to give a blue print, which is actually implemented by the developers. On realizing the importance of systematic documentation all the processes are implemented using a software engineering approach. Working in live environment enables one to appreciate the intricacies involved in the System Development Life Cycle (SDLC).I have gained a lot of practical knowledge from this project, which we think, shall make us stand in a good state in the future.

**Limitations**

1. The size of the database increases day-by-day, increasing the load on the database back up and data maintenance activity.
2. Employee of the company can be able to make comments on the Blogs or Posts.
3. Employee can chat with other logged in employee in group chat or personal chat..

**Scope for Future Development**

For future development, we will first consider all previously assumed options which are not yet developed. This we can make the website livelier in action, so that can be considered as popular online job portal website.

At this some future development has to be taken into account, such as.

1. Auto-Generated Mailing System informing both the Employee and the Administrator.
2. The errors occurred due to failure nature of the website has to be reduce as far possible to make the site & more user friendly in nature.
3. SMS-Alert direct phone.

**REFERENCES**

**Web PAGES**

1. W3schools, (1996), w3school in profile 2007, <http://www.w3schools.com/php>, Retrieved on January 2015.
2. W3schools, (1996), w3school in profile 2007, <http://www.w3schools.com/angular/>,

Retrieved on February 2016.

1. W3schools, (1996), w3school in profile 2007, <http://www.w3schools.com/node/>,

Retrieved on February 2016.

1. <https://en.wikipedia.org/wiki/MongoDB>, Retrieved on March 2016.
2. <https://nodejs.org/en/docs/>, Retrieved on March 2016.
3. <http://www.tutorialspoint.com/mongodb/mongodb_overview.htm>, Retrieved on February 2016.
4. <https://angularjs.org/>, Retrieved on February 2016.