COMPLAINTS MANAGEMENT SYSTEM

**ABSTRACT**

**ABSTRACT**

Online Complaint Management System provides an online way of solving the problems faced by the public by saving time and eradicate corruption. The objective of the complaints management system is to make complaints easier to coordinate, monitor, track and resolve, and to provide company with an effective tool to identify and target problem areas, monitor complaints handling performance and make business improvements. Online Complaint Management is a management technique for assessing, analyzing and responding to customer complaints. Complaints management software is used to record resolve and respond to customer complaints, requests as well as facilitate any other feedback. The main purpose of this project is to help the public in knowing their place details and getting their problems solved in online without going to the officer regularly until the problem is solved. By this system the public can save his time and eradicate corruption in government offices.In the proposed system the citizen need not go to the government office for getting his problem solved. He can get his problem solved by posting his problem in this proposed system and he can suggest a possible solution to the problems posted on the system. He can even get the information of the funds and other details of his place in detail through this system.

|  |  |  |
| --- | --- | --- |
| **CONTENTS** | | |
|  |  |  |
| 1 | INTRODUCTION  Introduction--------------------------------------- |  |
| 2 | SYSTEM STUDY  Problem statement-------------------------------  Existing system----------------------------------  Proposed system--------------------------------- |  |
| 3 | MODULE DESCRIPTION  Modules------------------------------------------ |  |
| 4 | FEASIBILITY STUDY  Operational Feasibility--------------------------  Technical Feasibility-----------------------------  Economical Feasibility-------------------------  Behavioral Feasibility--------------------------- |  |
| 5 | SYSTEM SPECIFICATION  Software Specification----------------------------  Hardware Specification--------------------------- |  |
| 6 | SYSTEM DESIGN  Input Design----------------------------------------  Output Design--------------------------------------  Data Flow Diagrams------------------------------  Database Design----------------------------------- |  |
| 7 | SYSTEM DEVELOPMENT  FrontEnd--------------------------------------------  BackEnd-------------------------------------------- |  |
| 8 | SYSTEM TESTING |  |
| 9 | SYSTEM IMPLEMENTATION  Software Implementation------------------------- |  |
| 10 | FUTURE ENHANCEMENT |  |
| 11 | CONCLUSION |  |
|  |  |  |
| 12 | REFERENCE |  |

**REQUIREMENT ANALYSIS AND SPECIFICATION**

**REQUIREMENT ANALYSIS**

Requirement analysis is significant and essential activity after elicitation .we analyses, refine and scrutinize the gathered requirement to make consistent and unambiguous requirements. This activity reviews all requirements and may provide a graphical view of the entire system. After the completion of the analysis, it is expected that the understandability of the project may improve significantly. Here, we may also use the interaction with the customer to clarify points of confusion and to understand which requirements are more important than others.

One effective way to find out what the customer wants is to construct a prototype, something that looks and preferably acts as part of the system they want. We can use their feedback to modify the prototype until the customer is satisfied continuously. Hence, the prototype helps the client to visualize the proposed system and increase the understanding of requirements. In requirement analysis the graphical view may help to find incorrect, inconsistent and missing requirements. Such models include the Data Flow diagram, Entity Relationship Diagram, etc.

After modeling the requirements, we will have a better understanding of the system behavior. The inconsistencies and ambiguities have been identified and corrected. Elicitation and analysis activities have provided better insight into the system. After the analyses requirements and the next step is to document these requirements in a prescribed format.

**Existing system**

In the existing system the citizens must go to the government office for any kind of help. For complaining about a problem there is a system called Prajavani in which the users can post their problems but cannot get the details of the problems and some other services. This system doesn’t have much popularity and is not user friendly.

**Proposed System**

Online Complaint Management System provides an online way of solving the problem faced by the public by saving time and eradicate corruption and the ability of providing many of the reports on the system and add to facilitate the process of submitting a complaint.

**MODULE DESCRIPTION**

1. Admin

2. Government officers

3. Users

**1. Admin**

➔Login

➔ Department registration

➔ Government officer registration

➔View complaints status

➔ Schedule appointments

**2. Government officers**

➔ Login

➔ View complaints

➔ Reply complaints

➔ View appointments

➔ Accept and reject appointments

➔View requests and reply

➔ View reviews

**3. Users/public**

➔ Register

➔Login

➔ Complaints register

➔View complaints status

➔View schedules

➔Request for appointments

➔view appointment status

➔Send requests

**REQUIREMENT SPECIFICATION**

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

A good SRS defines how the software system will interact with all internal modules ,hardware, communication with other programs and human user interactions with a wide range of real life scenarios. 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 starting writing test cases and making any plan for testing.

* **Functional Requirements**

Functional Requirements defines a function of a system or its component, Where a function is described as a specification of behavior between outputs and inputs. Functional Requirements may involve calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. It specifies a particular result of a system.

* **Nonfunctional Requirements**

A nonfunctional requirement defines the quality attribute of a software system. They judge the software system based on reliability, usability, security, portability and other nonfunctional standards that are critical to success of the software system. The process of specifying nonfunctional requirements requires the knowledge of the functionality of the system as well as the knowledge of the context within which the system will operate.

* **Security**
* **Reliable**
* **User friendly**
* **Platform independent**
* **Efficiency**
* **Performance**

**FEASIBILITY STUDY**

During the system analysis, a feasibility study of the proposed system was carried out to see whether it was beneficial or not. The proposed system must be evaluated from a technical feasibility, operational feasibility and economical feasibility.

* **Technical Feasibility**

Technical feasibility concentrates on the existing computer system and to what extent it can support the proposed system. We had undergone technical feasibility study taking into consideration whether the available resources such as computer systems and the peripherals are sufficient to implement the new software. The system requires smartphones, but it is already available .The software, tools selected to provide technical guarantee of accuracy, reliability, ease of access and data security. Thus, for both hardware and software requirements to be satisfied it is technically feasible.

* **Economic Feasibility**

The economic feasibility is the most important and frequently used method for evaluating the effectiveness of the proposed system. It is very essential because the main goal of the proposed system is to have economically better results along with increased efficiency. Cost benefit analysis is usually performed for this purpose. The innovation of the new system has influence on the economical side of the user. The new system can perform more additional features than the existing systems.

* **Operational Feasibility**

The system operation is the longest phase in the development life cycle of a system. So, Operational Feasibility should be given much importance. The system is user friendly. The proposed system offers greater friendliness, greater understandability and simplicity.

**SYSTEM SPECIFICATION**

Hardware and software requirements for the installation and smooth functioning of this product could be configured based on the requirements needed by the component of the operating environment that works as front-end system here we suggest minimum configuration for the both hardware and software components.Working off with this software is requirements concrete on system environments. It includes two phases.

* Hardware Requirements
* Software Requirements
* **SOFTWARE SPECIFICATION**

Operating System : Windows 10/XP OR Linux Coding Language : PYTHON

Framework ,:Django

Frontend :HTML,CSS,JS

Database :sqllite3

Browser : Mozilla Firefox/ Google Chrome /Internet Explorer IDE : Pycharm

Protocol : HTTP, SMTP.

* + **5.2 HARDWARE SPECIFICATION**

Processor : Intel core i3 or above

Clock Speed : 500 MHZ

System Bus : 32Bit

RAM : Minimum 2GB(recommended 8GB)

HDD : 20GB

Monitor : SVGA COLOR

Key Board : 108 Keys

Mouse : Any type of mouse

**SYSTEM DESIGN**

**SYSTEM DESIGN**

System design is the process of developing specifications for a candidate system that meet the criteria established in the system analysis. Major step in system design is the preparation of the input forms and the output reports in a form applicable to the user.

The main objective of the system design is to use the package easily by any computer operator. System Design is the creative act of invention, developing new inputs, a database, offline files, methods, procedures and output for processing business to meet an organization’s objective. System design builds information gathered during the system analysis.

The system design is the most creative and challenging phase. The first step is to determine how the output is produced and in what format. Samples of input and output are presented. Next, the input data and the master data are to be designed to meet the requirements of the proposed output. The operational phases are handled through program construction testing, including a list of programs needed to meet the system objective and complete documentation.

* 1. **INITIAL DESIGN**

Initial design of the complaints management system is all about designing user interface for operating and satisfying user requirements. For that purpose, we have investigated well- known technologies and identified HTML,CSS ,Bootstrap, javascript technologies etc.

We gathered requirements for this application in different ways and found that some technologies for coding and html,css and bootstrap for attractive styles.

**INPUT DESIGN**

Input design is the process of converting the user originated inputs to a computer format. The input design involves determining what the inputs are, how the data should be performed, how to validate data, how to minimize data entry and how to provide a multi user facility. The design for handling input specifies how data is accepted for computer processing. Input design is a part of overall system design that needs careful attention .

A system user interacting through a system must be able to tell the system whether to accept input, produce a report or end processing. The collection of input data is considered to be the most expensive part of the system design. Since the inputs have to be planned in such a manner so as to get the relevant information extreme care is taken to obtain the information. If the data going into the system is incorrect then processing and outputs will magnify this error. All input data are validated in the order and if any data violates any conditions, the user is warned by a message. If the data satisfies all the conditions, then it is transferred to the appropriate tables in the database.

We have to keep in mind the following things to design the system

* + - What data to input?
    - What medium to use?
    - The dialogue to guide users in providing input.
    - Methods for performing input validation and steps to follow when errors occur Input requirement gathering was one of the major trivial processes in web or android application development. The project involves text inputs.The inputs can be entered through keyboard and mouse. The text input is gathered by forms with text boxes.

**OUTPUT DESIGN**

Effective output design will improve the clarity and performance of output. Output design phase of the system is concerned with the convergence of information to the end user friendly manner. The output design should be efficient, intelligible so that system relationship with the end user is improved and thereby enhancing the process of decision making.

The Outputs from the COMPLAINT MANAGEMENT SYSTEM consist of different formats. Efficient and eligible output design should improve the system’s relationship with the user and help in decision making.

Outputs are the most important and direct source of information to the user and to the management. Output design generally deals with the results generated by the system. The output data is in the format of text.

**DATA FLOW DIAGRAM**

A Data Flow Diagram (DFD) is a graphical representation of ―flow of data through an information system, modelling its process aspects. Often they are preliminary steps used to create the overview of a system which can later be elaborated. DFD is also used for visualization of data processing (Structured design).

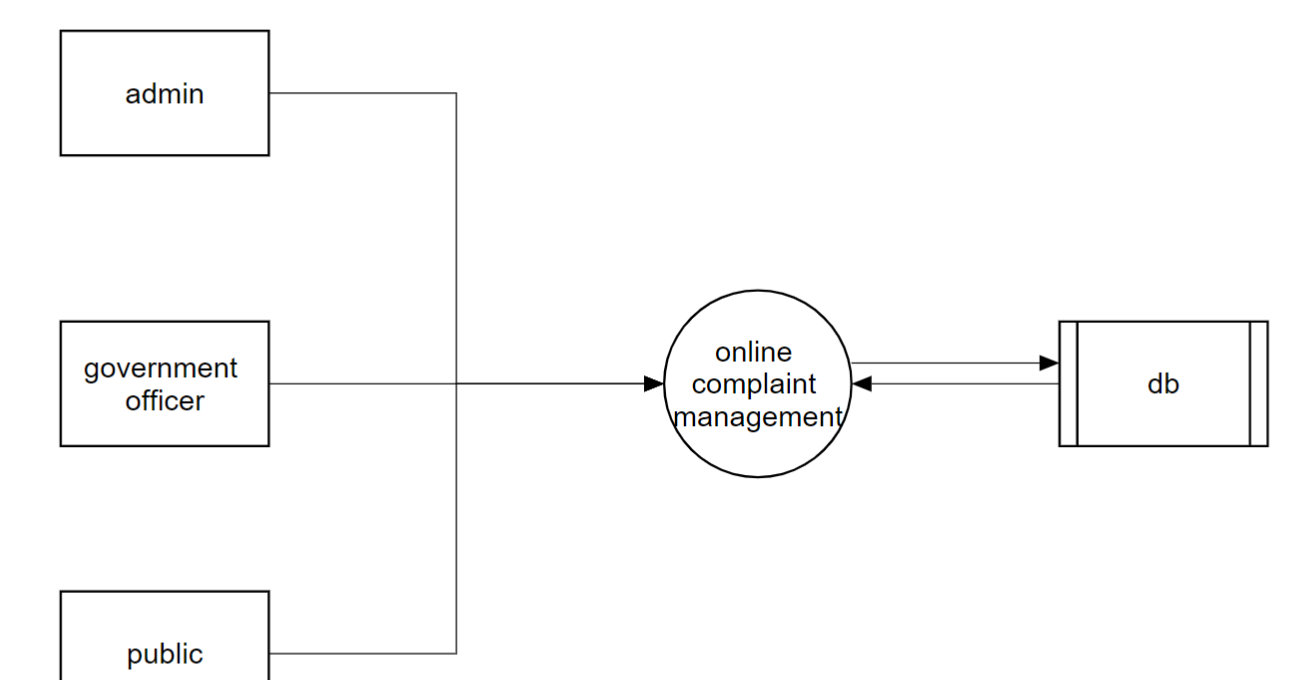
A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to and where the data will be stored. It does not show information about the timing of the process, or information about whether the process will operate in sequence or in parallel.

|  |  |  |
| --- | --- | --- |
| **Symbol** | **Name** | **Function** |
|  | Process | Performs some transformation of input data to yield output data. |
|  | Data Flow | Used to connect processes to each other, to sources or sinks; the arrow head indicates direction of dataflow. |
|  | Source or sink(External Entity) | A source of system inputs or sinks of system outputs. |
|  | Data store | A repository of data |

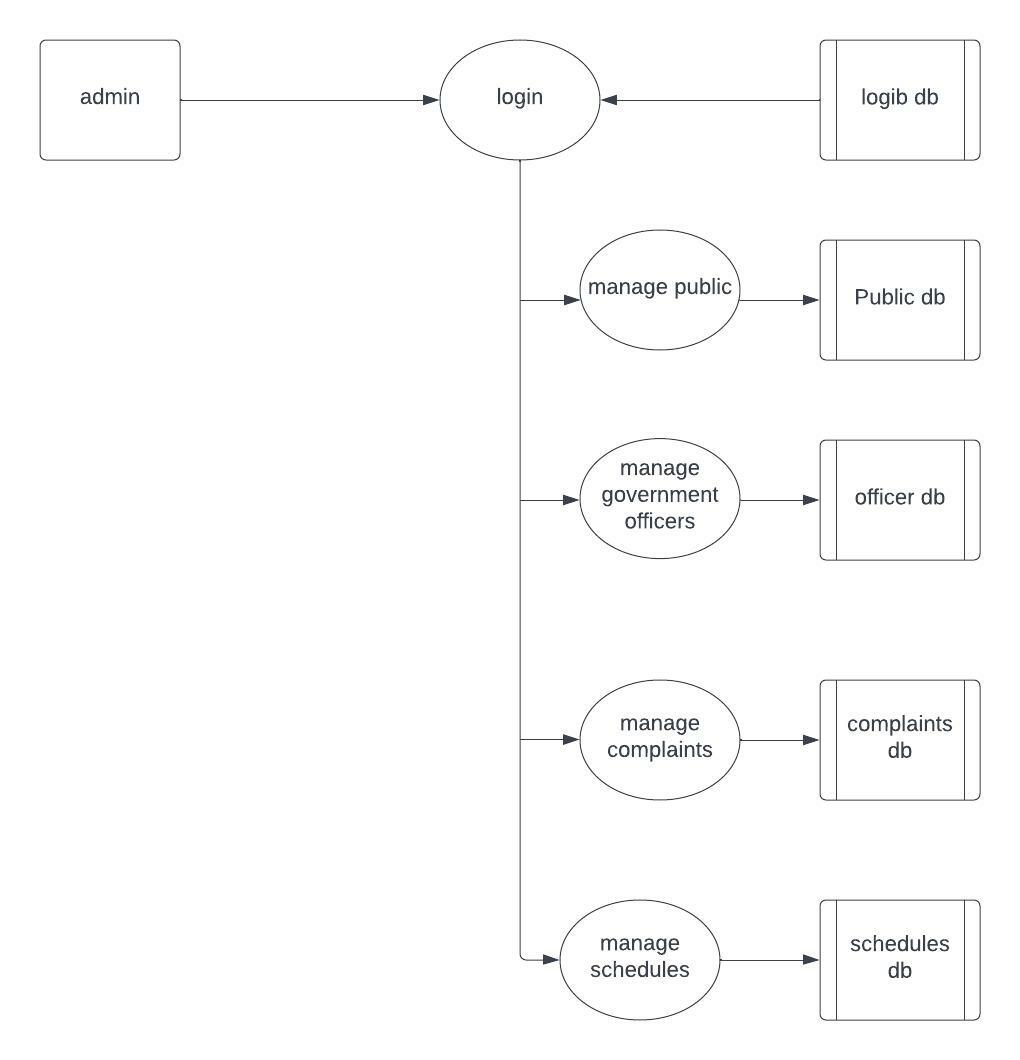
**Fig:** Above depicted are the major shapes used in DFD.

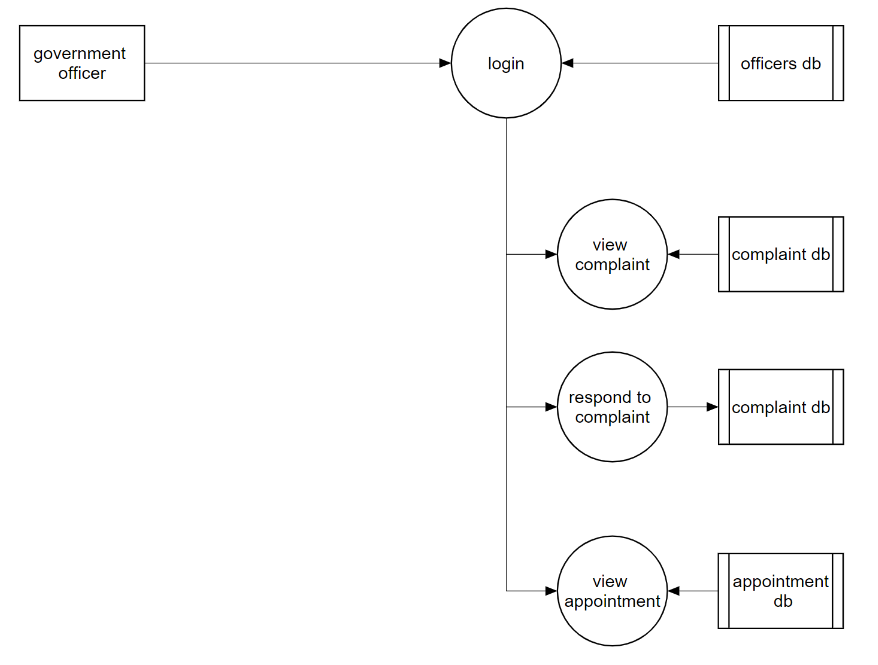
The DFD at the simplest level is referred to as the context analysis diagram. These are referred to as explaining its process in detail. Processes are numbered for easy identification and the data stores, source and destination of data are normally labelled in block letters. Each data flow is labelled for easy understanding.

**Level 0**

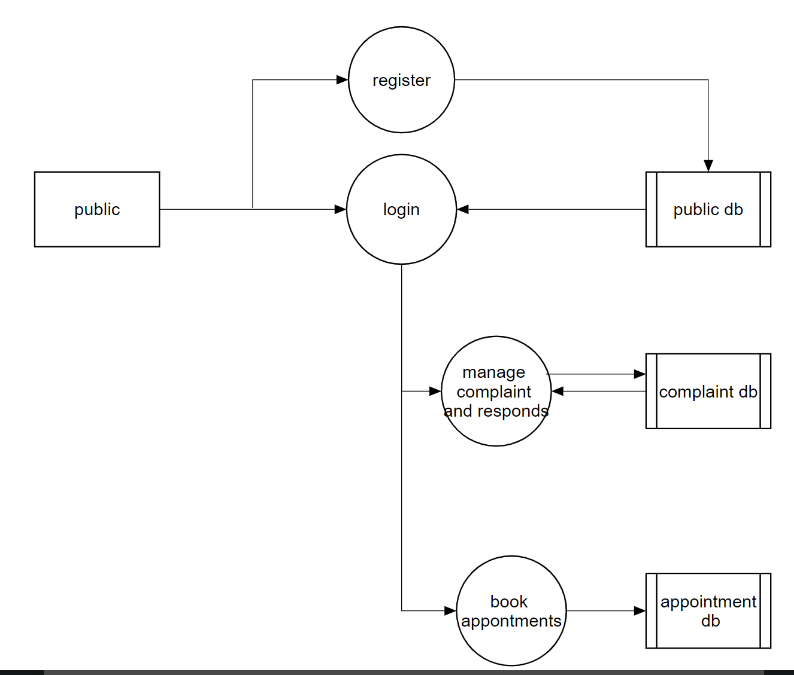
****

**Level 1**

****

****

**Level 2**

****

**Level 3**

**DATABASE DESIGN**

Database design manages large bodies of information. Database is the collection of related data. It provides safety of information. A database is a collection of interrelated data stored with minimum redundancy to save many users quickly and effectively. Database runs parallel without application design.

**Data Normalization**

The normalization simplifies the entries, removing redundancies from the system data and finally builds a data structure, which is both flexible and adaptable to the system. Normalization usually involves dividing a database into two or more tables and defining relationships between the tables. The objective is to isolate data so that additions, deletions, and modifications of a field can be made in just one table and then propagated through the rest of the database via the defined relationships.The different normal forms applied during the design of the database are:

* + - * **First Normal Form:**

A relationship is said to be in first normal form if and only if it satisfies the constraints that it contains atomic values.It states that the domain of an attribute must include only atomic (simple, indivisible) values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. It was defined to disallow multivalued attributes, composite attributes, and their combinations.First normalization used in all tables in the database.

* **Second Normal Form:**

A relationship is said to be in second normal form if it satisfies the 1NF condition for the primary key and every non primary key attribute of the relation should not depend on the primary key alone.Second normal form (2NF) is based on the concept of full functional dependency. A functional dependency X - Y is a fully functional dependency if removal of any attribute A from X means that the dependency does not hold any more.This normalization can be used in all the tables in the database because all the tables contain a primary key

* **Third Normal Form**

A relationship is said to be in third normal form if and only if it satisfies all the 2NF conditions and non-key attributes of the relation should not depend on other non-key attributes.Third normal form (3NF) is based on the concept of transitive dependency. A functional dependency X ~ Y in a relation schema R is a transitive dependency if there is a set of attributes Z that is neither a candidate key nor a subset of any key of R, and both X -Z and Z –Y hold.This normalization can be used in tables

1. **Table: Login**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Constraints** |
| Username | varchar(150) | Primary key |
| Password | varchar(128) |  |

**2. Table: Government officer**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Constraints** |
| Id | Varchar(100) | Primary key |
| Department | Varchar(20) |  |
| Contact no | Varchar(100) |  |
| email | Varchar(254) |  |
| address | Varchar(250) |  |
|  |  |  |

**3. Table: Public Registration**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Constraints** |
| Username | Varchar(150) | Primary key |
| Password | Varchar(150) |  |
| Address | Varchar(100) |  |
| Phone | Varchar(100) |  |
| E-mail | Varchar(254) |  |

**4. Table: Appointment**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Constraints** |
| Appointment Id | Varchar(11 | Primary key |
| Department | Varchar(20) |  |
| Start time | Date time |  |
| End time | Date time |  |

**5. Table: Complaint**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Constraints** |
| ComplaintId | Varchar(11) | Primary key |
| Description | Varchar(500) |  |
| Reply | Text |  |

**6. Table: Request**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Constraints** |
| RequestId | Varchar(11) | Primary key |
| Date | Date time |  |
| Subject | Varchar(20) |  |
| Document | Varchar(20) |  |
| Reply | Text |  |

**7. Table: department**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Constraints** |
| Name | Varchar(200) | Primary key |
| Place | Varchar(200) |  |
| Email | Varchar(254) |  |
| Contact number | Varchar(100) |  |

**SYSTEM DEVELOPMENT**

The goal of the development stage is to translate the design of the system produced during the design phase into code in a given programming language which can be executed by a computer and which performs the computations specified by the design.

**PYTHON DJANGO**

Python is an [interpreted](https://en.wikipedia.org/wiki/Interpreted_language), [high-level](https://en.wikipedia.org/wiki/High-level_programming_language) and [general-purpose programming language](https://en.wikipedia.org/wiki/General-purpose_programming_language). Created by [Guido van Rossum](https://en.wikipedia.org/wiki/Guido_van_Rossum) and first released in 1991, Python's design philosophy emphasizes [code readability](https://en.wikipedia.org/wiki/Code_readability) with its notable use of [significant whitespace](https://en.wikipedia.org/wiki/Off-side_rule). Its [language constructs](https://en.wikipedia.org/wiki/Language_construct) and [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) approach aim to help [programmers](https://en.wikipedia.org/wiki/Programmers) write clear, logical code for small and large-scale projects.

Python is [dynamically typed](https://en.wikipedia.org/wiki/Dynamic_programming_language) and [garbage-collected](https://en.wikipedia.org/wiki/Garbage_collection_(computer_science)). It supports multiple [programming paradigms](https://en.wikipedia.org/wiki/Programming_paradigms), including [structured](https://en.wikipedia.org/wiki/Structured_programming) (particularly, [procedural](https://en.wikipedia.org/wiki/Procedural_programming)), [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming), and [functional programming](https://en.wikipedia.org/wiki/Functional_programming). Python is often described as a "batteries included" language due to its comprehensive [standard library](https://en.wikipedia.org/wiki/Standard_library).

Python was created in the late 1980s as a successor to the [ABC language](https://en.wikipedia.org/wiki/ABC_(programming_language)). Python 2.0, released in 2000, introduced features like [list comprehensions](https://en.wikipedia.org/wiki/List_comprehension) and a garbage collection system with [reference counting](https://en.wikipedia.org/wiki/Reference_counting).

Python 3.0, released in 2008, was a major revision of the language that is not completely [backward-compatible](https://en.wikipedia.org/wiki/Backward_compatibility), and much Python 2 code does not run unmodified on Python 3.

The Python 2 language was officially discontinued in 2020 (first planned for 2015), and "Python 2.7.18 is the last Python 2.7 release and therefore the last Python 2 release."No more security patches or other improvements will be released for it. With Python 2's [end-of-life](https://en.wikipedia.org/wiki/End-of-life_(product)), only Python 3.6.xand later are supported.

Python [interpreters](https://en.wikipedia.org/wiki/Interpreter_(computing)) are available for many [operating systems](https://en.wikipedia.org/wiki/Operating_system). A global community of programmers develops and maintains [Python](https://en.wikipedia.org/wiki/CPython), a [free and open-source](https://en.wikipedia.org/wiki/Free_and_open-source_software)[[34]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-34) [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation). A non-profit organization, the [Python Software Foundation](https://en.wikipedia.org/wiki/Python_Software_Foundation), manages and directs resources for Python and Python development.

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It’s free and open source.

Django includes dozens of extras you can use to handle common Web development tasks. Django takes care of user authentication, content administration, site maps, RSS feeds, and many more tasks. Django takes security seriously and helps developers avoid many common security mistakes, such as SQL injection, cross-site scripting, cross-site request forgery and click jacking. Its user authentication system provides a secure way to manage user accounts and passwords. Django have ability to quickly and flexibly scale to meet the heaviest traffic demands.

`Django was created in the fall of 2003, when the web programmers at the Lawrence Journal*-*World newspaper, Adrian Holovaty and Simon Willison, began using Python to build applications. Jacob Kaplan-Moss was hired early in Django's development shortly before Simon Willison's internship ended. It was released publicly under a BSD license in July 2005. The framework was named after guitarist Django Reinhardt.In June 2008, it was announced that a newly formed Django Software Foundation (DSF) would maintain Django in the future.

**JAVASCRIPT**

JavaScript is an object-oriented scripting language used to enable programmatic access to objects with both the client application and other applications .It is primarily used in the form of client-side JavaScript, implemented as an integrated component of the web browser, allowing the development of enhanced user interface and dynamic website. JavaScript is a dialect of the ECMAScript that was influenced by many languages and was designed to look like Java, but to be easier for non-programmers to work with.

JavaScript, despite the name, is essentially unrelated to the Java programming language even though the two do have superficial similarities. Both languages use syntaxes influenced by that of C syntax, and JavaScript copies many Java names and naming conventions. The language’s name is the result of a co-marketing deal between Netscape and Sun, in exchange for Netscape bundling Sun's Java runtime with their then-dominant browser. The key design principles within JavaScript are inherited from the self and Scheme programming languages.

JavaScript is a trademark of Sun Microsystems. It was used under license for technology invented and implemented by Netscape communications and current entities such as the Mozilla Foundation.

**CASCADING STYLE SHEETS**

**Cascading Style Sheets** (**CSS**) is a [style sheet language](https://en.wikipedia.org/wiki/Style_sheet_language) used for describing the [presentation](https://en.wikipedia.org/wiki/Presentation_semantics) of a document written in a [markup language](https://en.wikipedia.org/wiki/Markup_language) like [HTML.](https://en.wikipedia.org/wiki/HTML) CSS is a cornerstone technology of the [World Wide Web,](https://en.wikipedia.org/wiki/World_Wide_Web) alongside HTML and [JavaScript](https://en.wikipedia.org/wiki/JavaScript).CSS is designed to enable the separation of presentation and content, including [layout](https://en.wikipedia.org/wiki/Page_layout), [colors,](https://en.wikipedia.org/wiki/Color) and [fonts.](https://en.wikipedia.org/wiki/Typeface) This separation can improve content [accessibility,](https://en.wikipedia.org/wiki/Accessibility) provide more flexibility and control in the specification of presentation characteristics, enable multiple [web pages](https://en.wikipedia.org/wiki/Web_page) to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content. Separation of formatting and content also makes it possible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or [screen reader](https://en.wikipedia.org/wiki/Screen_reader)), and on [Braille-based](https://en.wikipedia.org/wiki/Braille_display) tactile devices. CSS also has rules for alternate formatting if the content is accessed on a [mobile device.](https://en.wikipedia.org/wiki/Mobile_device)

The name *cascading* comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable. The CSS specifications are maintained by the [World Wide Web Consortium](https://en.wikipedia.org/wiki/World_Wide_Web_Consortium) (W3C). Internet media type ([MIME type](https://en.wikipedia.org/wiki/MIME_media_type)) text/css is registered for use with CSS by [RFC 2318](https://tools.ietf.org/html/rfc2318) (March 1998). The W3C operates a free [CSS validation service](https://en.wikipedia.org/wiki/W3C_Markup_Validation_Service#CSS_validation) for CSS documents. In addition to HTML, other markup languages support the use of CSS, including [XHTML,](https://en.wikipedia.org/wiki/XHTML) [plain XML,](https://en.wikipedia.org/wiki/Plain_Old_XML) [SVG](https://en.wikipedia.org/wiki/Scalable_Vector_Graphics), and [XUL.](https://en.wikipedia.org/wiki/XUL)

**BOOTSTRAP**

**Bootstrap** is a [free and open-source](https://en.wikipedia.org/wiki/Free_and_open-source_software) front-end [framework](https://en.wikipedia.org/wiki/Application_framework) for designing [websites](https://en.wikipedia.org/wiki/Website) and [web applications.](https://en.wikipedia.org/wiki/Web_application) It contains [HTML](https://en.wikipedia.org/wiki/HTML)- and [CSS](https://en.wikipedia.org/wiki/CSS)-based design templates for [typography,](https://en.wikipedia.org/wiki/Typography) forms, buttons, navigation and other interface components, as well as optional [JavaScript](https://en.wikipedia.org/wiki/JavaScript) extensions. Unlike many earlier web frameworks, it concerns itself with [front-end](https://en.wikipedia.org/wiki/Front-end_web_development) [development](https://en.wikipedia.org/wiki/Front-end_web_development) only. Bootstrap, originally named Twitter Blueprint, was developed by Mark Otto and Jacob Thornton at [Twitter](https://en.wikipedia.org/wiki/Twitter) as a framework to encourage consistency across internal tools. Before Bootstrap, various libraries were used for interface development, which led to inconsistencies and a high maintenance burden.After a few months of development by a small group, many developers at Twitter began to contribute to the project as a part of Hack Week, a [hackathon](https://en.wikipedia.org/wiki/Hackathon)-style week for the Twitter development team. It was renamed from Twitter Blueprint to Bootstrap, and released as an open source project on August 19, 2011.

**STRUCTURE AND FUNCTION**

Bootstrap is modular and consists of a series of [Less](https://en.wikipedia.org/wiki/Less_(stylesheet_language)) ([Sass](https://en.wikipedia.org/wiki/Sass_(stylesheet_language)) version 4 and onward) stylesheets that implement the various components of the toolkit. These stylesheets are generally compiled into a bundle and included in web pages, but individual components can be included or removed. Bootstrap provides a number of configuration variables that control things such as color and padding of various components. Since Bootstrap 2, the Bootstrap documentation has included a customization wizard which generates a customized version of Bootstrap based on the requested components and various settings. As of Bootstrap 4, [Sass](https://en.wikipedia.org/wiki/Sass_(stylesheet_language)) is used instead of [Less](https://en.wikipedia.org/wiki/Less_(stylesheet_language)) for the stylesheets. Each Bootstrap component consists of an HTML structure, CSS declarations, and in some cases accompanying JavaScript code. Grid system and responsive design comes standard with an 1170-pixel-wide [grid layout.](https://en.wikipedia.org/wiki/CSS_Grid_Layout) Alternatively, the developer can use a variable-width layout. For both cases, the toolkit has four variations to make use of different resolutions and types of devices: mobile phones, portrait and landscape, tablets and PCs with low and high resolution. Each variation adjusts the width of the columns.

**PHYCHARM IDE**

PyCharm is an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) used in [computer programming](https://en.wikipedia.org/wiki/Computer_programming), specifically for the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) language. It is developed by the [Czech](https://en.wikipedia.org/wiki/Czech_Republic) company [JetBrains](https://en.wikipedia.org/wiki/JetBrains).It provides code analysis, a graphical debugger, an integrated unit tester, integration with [version control systems](https://en.wikipedia.org/wiki/Revision_control) (VCSes), and supports web development with [Django](https://en.wikipedia.org/wiki/Django_(web_framework)) as well as [data science](https://en.wikipedia.org/wiki/Data_science) with [Anaconda](https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)).

PyCharm is [cross-platform](https://en.wikipedia.org/wiki/Cross-platform), with [Windows](https://en.wikipedia.org/wiki/Windows), [macOS](https://en.wikipedia.org/wiki/MacOS) and [Linux](https://en.wikipedia.org/wiki/Linux) versions. The Community Edition is released under the [Apache License](https://en.wikipedia.org/wiki/Apache_License),and there is also Professional Edition with extra features – released under a [proprietary license](https://en.wikipedia.org/wiki/Proprietary_software).

**7.2. BACK END:SQLite**

SQLite is an in-process library that implements a [self-contained](https://www.sqlite.org/selfcontained.html), [serverless](https://www.sqlite.org/serverless.html), [zero-configuration](https://www.sqlite.org/zeroconf.html), [transactional](https://www.sqlite.org/transactional.html) SQL database engine. The code for SQLite is in the [public domain](https://www.sqlite.org/copyright.html) and is thus free for use for any purpose, commercial or private. SQLite is the [most widely deployed](https://www.sqlite.org/mostdeployed.html) database in the world with more applications than we can count, including several [high-profile projects.](https://www.sqlite.org/famous.html)

SQLite is an embedded SQL database engine. Unlike most other SQL databases, SQLite does not have a separate server process. SQLite reads and writes directly to ordinary disk files. A complete SQL database with multiple tables, indices, triggers, and views, is contained in a single disk file. The database [file format](https://www.sqlite.org/fileformat2.html) is cross-platform - you can freely copy a database between 32-bit and 64-bit systems or between [big-endian](http://en.wikipedia.org/wiki/Endianness) and [little-endian](http://en.wikipedia.org/wiki/Endianness) architectures. These features make SQLite a popular choice as an [Application File Format](https://www.sqlite.org/appfileformat.html).

SQLite database files are a [recommended storage format](https://www.sqlite.org/locrsf.html) by the US Library of Congress. Think of SQLite not as a replacement for [Oracle](http://www.oracle.com/database/index.html) but as a replacement for [fopen()](http://man.he.net/man3/fopen)

SQLite is a compact library. With all features enabled, the [library size](https://www.sqlite.org/footprint.html) can be less than 600KiB, depending on the target platform and compiler optimization settings. (64-bit code is larger. And some compiler optimizations such as aggressive function inlining and loop unrolling can cause the object code to be much larger.) There is a tradeoff between memory usage and speed. SQLite generally runs faster the more memory you give it. Nevertheless, performance is usually quite good even in low-memory environments. Depending on how it is used, SQLite can be [faster than direct filesystem I/O](https://www.sqlite.org/fasterthanfs.html).

SQLite is [very carefully tested](https://www.sqlite.org/testing.html) prior to every release and has a reputation for being very reliable. Most of the SQLite source code is devoted purely to testing and verification. An automated test suite runs millions and millions of test cases involving hundreds of millions of individual SQL statements and achieves [100% branch test coverage](https://www.sqlite.org/testing.html#coverage). SQLite responds gracefully to memory allocation failures and disk I/O errors. Transactions are [ACID](http://en.wikipedia.org/wiki/ACID) even if interrupted by system crashes or power failures. All of this is verified by the automated tests using special test harnesses which simulate system failures. Of course, even with all this testing, there are still bugs. But unlike some similar projects (especially commercial competitors) SQLite is open and honest about all bugs and provides [bugs lists](http://www.sqlite.org/src/rptview?rn=1) and minute-by-minute [chronologies](http://www.sqlite.org/src/timeline) of code changes.

The SQLite code base is supported by an [international team](https://www.sqlite.org/crew.html) of developers who work on SQLite full-time. The developers continue to expand the capabilities of SQLite and enhance its reliability and performance while maintaining backwards compatibility with the [published interface spec](https://www.sqlite.org/c3ref/intro.html), [SQL syntax](https://www.sqlite.org/lang.html), and database [file format](https://www.sqlite.org/fileformat2.html). The source code is absolutely free to anybody who wants it, but [professional support](https://www.sqlite.org/prosupport.html) is also available.

The SQLite project was started on [2000-05-09](https://www.sqlite.org/src/timeline?c=2000-05-29+14:26:00). The future is always hard to predict, but the intent of the developers is to support SQLite through the year 2050. Design decisions are made with that objective in mind.

We the developers hope that you find SQLite useful and we entreat you to use it well: to make good and beautiful products that are fast, reliable, and simple to use. Seek forgiveness for yourself as you forgive others.

.

**SYSTEM TESTING**

**SYSTEM TESTING**

Testing is very important in determining the reliability and efficiency of software, and hence it is a very crucial stage in software development. Tests are conducted on the software to evaluate its performance at different levels.

Testing is vital to the success of the system. It is the penultimate step of software development. An elaborate testing of data is prepared and the system is using test data. While doing testing errors are noted and correction is made. The users are trained to operate the developed system. Both hardware and software are made to run the developed system successfully.

Testing is a process of executing a program with the intent of finding whether the software achieves the desired result. Tests are conducted to locate an undiscovered error. Different types of data are fed into the system and the end result is verified with the expected results. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. Testing is vital to the success of the system.

The various testing methods used for this application are:

* + - Unit testing
    - Integration testing
    - Validation testing
    - Output testing
  1. **Unit Testing**

Unit testing focuses verification efforts on the smallest unit of software design module. To check whether each module in this software works properly so that it gives desired outputs to the given inputs. All validations and conditions are tested in the module level in the unit test control parts are tested to ensure the information correctly flows into, and out of the program unit under test. Boundary conditions are tested to ensure that modules operate at the boundaries. All independent parts through the control structure ensure that all statements in a module have been executed at least once. We test all the activity of the MYAJAS application works properly.

* 1. **Integration Testing**

Data can be lost across an interface, one module has an adverse effect on the other sub-functions, when combined may not produce the desired functions. Integrated testing is the systematic way to uncover the errors within the interface. This testing is done by inputting necessary values and data in the sequential order. As data of one unit is needed for working of another. The need for an integrated system is to find the overall system performance.In MYAJAS application the data tested across the interface are found correctly and works fine.

* 1. **Validation Testing**

Validation test is defined with a simple definition that validation succeeds when the system functions in a manner that can be reasonably accepted by the customer. Validation is done to see whether the corresponding entries made in the tables are correct. Proper validations are done in case of insertion and updating of tables. If any such arises, then proper error messages or warning, if any, has to be displayed.

In MYAJAS application the validation testing is done to different fields especially in registration and signup fields. Where the validation done through php program.Main validation used is NULL value in the text boxes.And also done Mobile number , Email validation.

* 1. **Output Testing**

After performing validation testing, the next step is output testing of the proposed system. Since the system cannot be useful if it does not produce the required output. Asking the user about the format in which the system is required tests the output displayed or generated by the system under consideration. While testing, errors are again uncovered and corrected by using the above steps and correction are also noted for future use. The system has been verified and validated by running test data and live data. The output testing of MYAJAS application is done properly. The output is correct while testing.

**User Acceptance Testing**

User acceptance testing of the system is the key factor for the success of any system. As we have created any easy interface it is easy to use. Only tough portion is the information to be perfect and has only one kind of user: the administrator.

The system under consideration is tested for user acceptance by constantly keeping in touch with prospective systems at a time of the development and making changes whenever required. This is done with regard to the input screen design and output screen design.

**SYSTEMIMPLEMENTATION &**

**SYSTEM MAINTENANCE**

**SYSTEM IMPLEMENTATION**

The implementation includes all those activities that take place to convert from the old system to new. The manual data keeping in educational institutions that should be overcome from the proposed new system. A proper implementation is essential to provide a reliable system to meet the requirements of the customers. An improper implementation may affect the success of the android application.

There are several methods for handling the implementation and the consequent conversion from the old applications to the new application developed in this project.

The most secure methods for comparing the old system and the new system is to run the old and new system in parallel. In this approach, a person may operate the old existing application and the new application. This method offers high reliability and security.

**User Training**

The implementation of the proposed system includes the training of system operators. Training the system operators includes not only instructions in how to use the equipment, but also in how to diagnose malfunctions and in what steps to take when they occur. So proper training should be provided to the system operators. No training is complete without familiarizing users with simple system maintenance activities. Since the proposed system is developed in a GUI, training will be comparatively easier than systems developed in a non-GUI. There are different types of training. We can select off-site to give depth knowledge to the system operators.

Success of the system depends on the way in which it is operated and used. Therefore, the quality of training given to the operating person affects the successful implementation of the system. The training must ensure that the person can handle all the possible operations.

Training must also include data entry personnel. They must also be given training for the installation of new hardware, terminals, how to power the system, how to power it down, how to detect the malfunctions, how to solve the problems etc. the operators must also be provided with the knowledge of trouble shooting which involves the determination of the cause of the problem.

The proposed system requires trained personnel for operating the system. This will reduce the data entry errors considerably. It is preferable to provide the person with some kind of operating manuals that will explain all the details of the system.

For the purpose of training we have improved our user interface for a guiding style of use and we are providing an intuitive interface for users. Along with all the simplicity we are providing a help section for users of the application with a detailed description of how each module is working and feature wise specialties and benefits.

**System Maintenance**

Maintenance of the software is one of the major steps in the development of the computer system. Software, which is developed by the engineer, should undergo maintenance processes in a regular interval of time as time on new problems arises and it must be corrected accordingly. Maintenance and enhancement are a long-term process.

In this project, the maintenance is carried over by the staff. Since they are the key persons to develop this project they know clearly about the project and coding structure. So, they will change the coding whenever required. Regarding the project maintenance, the changes will occur then and there according to the conditions

Various types of maintenance that can be made are:

* **Corrective maintenance**: reactive modification (or repairs) of a software product performed after delivery to correct discovered problems. Included in this category is emergency maintenance, which is an unscheduled modification performed to temporarily keep a software product operational pending corrective maintenance.
* **Adaptive maintenance**: modification of a software product performed after delivery to keep a software product usable in a changed or changing environment. For example, the operating system might be upgraded and some changes to the software may be necessary.
* **Perfective maintenance**: modification of a software product after delivery to provide enhancements for users, improvement of program documentation, and recording to improve software performance, maintainability, or other software attributes.
* **Preventive Maintenance**: modification of a software product after delivery to detect and correct latent faults in the software product before them become operational faults.

The staff in the concern takes part in each and every level of the project. So they don’t need any training in software. During the development process they sat and entered each and every entry to test the project. They themselves used this as an opportunity to take training that is not needed for the users.

***SCOPE FOR FUTURE ENHANCEMENT***

**SCOPE FOR FUTURE ENHANCEMENT**

This **Complaint management system** is an **application,** has been designed and developed according to the current techniques and scopes of web designing and development tools. This system is very flexible so that the maintenance and further amendments based on the changing environment and requirements can be made easily with adding further information. Further enhancement is possible to update evaluating tools. This can be restructured as required.

**CONCLUSION**

.

The project named COMPLAINT MANAGEMENT SYSTEM developed as a web application

Provides an online way of solving problems faced by the public by saving time and and

eradicate corruption

The project is developed using Python, Django and Sqlite as back end. This language selection is based on the requirement specification of the user and analysis of the existing system, with flexibility for future enhancement. Since the system is developed in modules, future enhancement is very easy.

**BIBLIOGRAPHY**

**BIBLIOGRAPHY**

**Web References**

1. [http://www.stackoverflow.com](http://www.stackoverflow.com/) , *Stack Overflow*
2. [http://www.codeproject.com](http://www.codeproject.com/) *, CodeProject*
3. [http://w3schools.com](http://w3schools.com/) , *W3Schools*
4. [http://getbootstrap.com](http://getbootstrap.com/) , *Bootstrap 4*
5. <http://developer.android.com/guide/topics/ui/declaring-layout.html>, *Layout*
6. [http://developer.android.com/resources/samples/get.html,](http://developer.android.com/resources/samples/get.html) *Sample Source Code*