# ABSTRACT

The project named “HIDDEN AUDIO TRAKER” is a hacker system. The purpose of this project is to transferring media voice or sound format as image when we can’t able to transfer a file directly as media format because of its security constraints. This is a site that helps us to transfer files securely by implementing the technique called Steganography. Steganography is a process that involves hiding a message in an appropriate carrier. Encryption and Decryption techniques have been used to make the security system robust. The user can set a different password for every message he/she sends.

# LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| **Sl. no** | **Table Name** | **Page No.** |
| **1.** | **Log in** | **28** |
| **2.** | **Messages** | **28** |
| **3.** | **Userreg** | **28** |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Sl. No.** | **Figure Name** | **Page No.** |
| **4.2** | **Basic DFD symbols** | **19** |
| **4.2.1** | **Context Level DFD** | **20** |
| **4.2.2** | **First Level DFD for User** | **21** |
| **4.2.3** | **First Level DFD for User registration** | **21** |
| **s 4.2.4** | **First Level DFD for User Log in** | **22** |
| **4.2.5** | **Second Level DFD for Compose mail** | **22** |
| **4.2.6** | **Second Level DFD for Save Drafts** | **22** |
| **4.2.7** | **Second Level DFD for Change Password** | **23** |
| **4.2.8** | **Second Level DFD for Sent items** | **23** |
| **6** | **Elements of Usecase Diagram** | **24** |
| **6.1** | **Usecase Diagram for User** | **25** |

# STUDY PHASE

## INTRODUCTION

The project named “HIDDEN AUDIO TRACKER” is a hacker system. The purpose of this project is to transfer media voice or sound format as an image in a situation where we are unable to transfer a file directly as media format or sound format because of the security constraints. This is a site that helps us to transfer files securely by implementing the technique called Steganography. Steganography is a process that involves hiding a message in an appropriate carrier. Encryption and Decryption techniques have been used to make the security system robust. The user can set different passwords for every message the user sends.

It is a hacker system that will help in the transferring media voice or sound format as image when we are unable to transfer a file directly as media format because of its security constrains. Nowadays the secured transferring of files is the major problem. To avoid this problem, we implement many technologies. One of the technologies is used for secured file transferring is steganography. Steganography is a means of obscuring data where secret messages are hidden inside computer files such as images, sound files, videos and even executable files so that, no one except the sender and the receiver will suspect the existence of the stealth information in it. In this project we can hide Audio in an image file.

## OBJECTIVES

* Studying the existing system to find out its problem.
* Finding the solution to the problem found in the existing system.
* Designing and building a web application of the system

## TECHNOLOGIES METERIALS AND METHODS

### DATABASE TOOLS

**MySQL**:

MySQL is a fast, easy to use relational database. It is currently the most popular opensource database. It is very commonly used in conjunction with PHP scripts to create powerful and dynamic server-side applications. MySQL is used for many small and big businesses. It is developed, marketed and supported by MySQL AB, a Swedish company. It is written in C and C++.

### PROGRAMMING TOOLS

**PYTHON:**

Python is a dynamic, high level, free open source and interpreted programming language. It supports object-oriented programming as well as procedural oriented programming. In Python, we don’t need to declare the type of variable because it is a dynamic typed language. For example, x=10, here x can be anything such as String, int etc.

## Features in Python

There are many features in Python, some of which are discussed below –

### Easy to code:

Python is high level programming language. Python is very easy to learn language as compared to other language like c, c#, java script, java etc. It is very easy to code in python language and anybody can learn python basic in few hours or days. It is also developer-friendly language.

### Free and Open Source:

Python language is freely available at official website and you can download it. Since, it is open-source, this means that source code is also available to the public. So you can download it as, use it as well as share it.

### Object-Oriented Language:

One of the key features of python is Object-Oriented programming. Python supports object-oriented language and concepts of classes, objects encapsulation etc.

### GUI Programming Support:

Graphical Users interfaces can be made using a module such as PyQt5, PyQt4, wxPython or Tk in python.PyQt5 is the most popular option for creating graphical apps with Python.

### High-Level Language:

Python is a high-level language. When we write programs in python, we do not need to remember the system architecture, nor do we need to manage the memory.

### Extensible feature:

Python is an Extensible language. We can write some python code into c or c++ language and also we can compile that code in c/c++ language.

### Python is Portable language:

Python language is also a portable language. For example, if we have python code for windows and if we want to run this code on other platform such as Linux, Unix and Mac then we do not need to change it, we can run this code on any platform.

### Python is Integrated language:

Python is also an Integrated language because we can easily integrated python with other language like c, c++ etc.

### Interpreted Language:

Python is an Interpreted Language. Because python code is executed line by line at a time. Like other language c, c++, java etc. there is no need to compile

python code this makes it easier to debug our code. The source code of python is converted into an immediate form called bytecode.

### Large Standard Library:

Python has a large standard library which provides rich set of module and functions so you do not have to write your own code for every single thing. There are many libraries present in python for such as regular expressions, unit- testing, web browsers etc.

### Dynamically Typed Language

Python is dynamically-typed language. That means the type (for example- int, double, long etc.) for a variable is decided at run time not in advance. Because of this feature we don’t need to specify the type of variable.

### Django

Django is a [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) based [free and open-source](https://en.wikipedia.org/wiki/Free_and_open-source_software) [web framework,](https://en.wikipedia.org/wiki/Web_framework) which follows the model-template-view (MTV) [architectural pattern.](https://en.wikipedia.org/wiki/Architectural_pattern_(computer_science)) Django's primary goal is to ease the creation of complex, database driven websites. The framework emphasizes [reusability](https://en.wikipedia.org/wiki/Reusability) and pluggability of components, less code, low coupling, rapid development, and the principle of [don't repeat yourself](https://en.wikipedia.org/wiki/Don%27t_repeat_yourself). Python is used throughout, even for settings files and data models. Django also provides an optional administrative [create, read, update and delete](https://en.wikipedia.org/wiki/Create%2C_read%2C_update_and_delete) interface that is generated dynamically through [introspection](https://en.wikipedia.org/wiki/Type_introspection) and configured via admin models.

Despite having its own nomenclature, such as naming the callable objects generating the [HTTP](https://en.wikipedia.org/wiki/HTTP) responses views, the core Django framework can be seen as an [MVC](https://en.wikipedia.org/wiki/Model-view-controller) architecture. It consists of an [object-relational mapper](https://en.wikipedia.org/wiki/Object-relational_mapping) (ORM) that mediates between [data models](https://en.wikipedia.org/wiki/Data_modeling) (defined as Python classes) and a [relational database](https://en.wikipedia.org/wiki/Relational_database) (Model), a system for processing HTTP requests with a [web templating system](https://en.wikipedia.org/wiki/Web_template_system) (View), and a [regular-expression](https://en.wikipedia.org/wiki/Regular_expression)-based [URL](https://en.wikipedia.org/wiki/Uniform_Resource_Locator) dispatcher (Controller).

Also included in the core framework are:

* A lightweight and standalone web server for development and testing.
* A form serialization and validation system that can translate between [HTML](https://en.wikipedia.org/wiki/HTML) forms and values suitable for storage in the database.
* A template system that utilizes the concept of [inheritance](https://en.wikipedia.org/wiki/Inheritance_(object-oriented_programming)) borrowed from object- oriented programming.
* A [caching](https://en.wikipedia.org/wiki/Web_cache) framework that can use any of several cache methods
* Support for [middleware](https://en.wikipedia.org/wiki/Middleware) classes that can intervene at various stages of request processing and carry out custom functions
* An internal dispatcher system that allows components of an application to communicate events to each other via pre-defined signal.
* An [internationalization](https://en.wikipedia.org/wiki/Internationalization_and_localization) system, including translations of Django's own components into a variety of languages.
* A serialization system that can produce and read xml and/or JSON representations of Django model instances.
* A system for extending the capabilities of the template engine
* An interface to Python's built-in [unit test](https://en.wikipedia.org/wiki/Unit_test) framework
* Django REST framework is a powerful and flexible toolkit for building Web APIs

### Java Script

Java script is a scripting language that can be used to create client-side scripts and server side scripts. Client-side scripts are executed in the browser while server- side scripts are executed on a server. That is JavaScript is an object-based scripting language for developing client based and server-based internet applications. We can insert JavaScript statements directly into an HTML page. When the page is displayed in the browser, the JavaScript statements are interpreted and executed by the browser. JavaScript statements can recognize and respond to user events such as mouse clicks or system generated events and so on. So you can change the content and position of the elements on the page dynamically, in response to user interaction.

When the client requests an HTML page that includes a client-side Script, the server forwards the full content of the HTML document- the JavaScript statements and the HTML content. When the browser receives the document, it executes the HTML and JavaScript statements without any interaction with the server

while both client-side JavaScript and server-side JavaScript have the same core language; each also has additional features relevant to the environment. That is, client-side JavaScript includes predefined objects that can be used only in the browser. Server-side JavaScript contain predefined objects that can be used in server-side application

### Hyper Text Mark Up Language

An HTML file is a text file containing small markup tags. These tags tell the web browser how to display the page. An HTML file must have an htm or html file extension. An HTML file can be created by using a simple text editor. HTML documents are text files made up of HTML elements e.g.: <html>, <body>. HTML elements are defined using HTML tags.

HTML tags are used to markup HTML elements. The two characters surround HTML tags <and>. The surrounding characters are called angle brackets. HTML tags normally come in pairs like <b> and </b>. The first tag in a pair is the start tag: the second tag is the end tag. The text between the start and the end tag is the element content. HTML tags are not case sensitive; <b> means the same as <B>.

## FEASIBILITY ANALYSIS

A feasibility study is an analysis that takes all of a project's relevant factors into account— including economic, technical, legal, and scheduling considerations—to ascertain the likelihood of completing the project successfully. Project managers use feasibility studies to discern the pros and cons of undertaking a project before they invest a lot of time and money into it.

Feasibility studies also can provide a company's management with crucial information that could prevent the company from entering blindly into risky businesses.

## Understanding Feasibility Studies

A feasibility study is simply an assessment of the practicality of a proposed plan or project. As the name implies, these studies ask: Is this project feasible? Do we have the people, tools, technology, and resources necessary for this project to

succeed? Will the project get us the return on investment (ROI) that we need and expect?

The goals of feasibility studies are as follows:

* To understand thoroughly all aspects of a project, concept, or plan
* To become aware of any potential problems that could occur while implementing The project.qa2
* To determine if, after considering all significant factors, the project is viable—that is, worth undertaking.

## The Importance of Feasibility Studies

Feasibility studies are important to business development. They can allow a business to address where and how it will operate. They can also identify potential obstacles that may impede its operations and recognize the amount of funding it will need to get the business up and running. Feasibility studies aim for marketing strategies that could help convince investors or banks that investing in a particular project or business is a wise choice.

## TECHNICAL FEASIBILITY

Technical feasibility concerns whether a project can meet its performance objectives. We have already seen some examples of this:

* Modelling the queues for the enrolment process checked that the proposed system would cope with its expected load.
* Developing an expert system for programme advice checked that it would be feasible for candidates to make wise enrolment choices over the internet.

The primary aim of a technical feasibility study is to remove uncertainty. Since the present value of a proposed project depends on the probability of its success, it is important to quantify the risk before committing money to it.

For example, some uncertainty still surrounds the automatic timetabling of examinations: Will a computer system be able to fit the examination sessions into the available period, as the existing manual system has proved that it can?

A key problem here is that the input data for the algorithm won’t be available until after enrolments are complete. If the timetabling application proves a failure, time and money will have been wasted, the enrolment staff will feel let down, and the reputation and morale of the project team will suffer. It is crucial that automatic timetabling succeeds. Unfortunately, even after the enrolment period, the data will only contain information for new candidates, and even a successful experiment won’t guarantee similar success when a set of data for all candidates become available.

The information that is needed for input to the timetabling process is a list of subject clashes : two subjects clash if at least one candidate is enrolled in both of them, which implies that their examinations mustn’t be held at the same time. To assume the worst

case, that every subject clashes with every other, would lead to a timetable that only allowed one subject per session, which is clearly unrealistic.

A more realistic test could be made by using the fact that subjects can only clash with subjects that are common to the same study programme and the same semester. Each subject might be assumed to clash with all Offered subjects that share any programme with it. This scenario can be tested as soon as the Components table has been populated, preceding the enrolment period. If the timetabling program works well with this set of data, it is certain to succeed in practice.

But what if this test proves unsatisfactory? If the programme proves inadequate, it is well to know as soon as possible. With luck, the subject clashes that were used to schedule the previous semester’s examination timetable might still be available. If they are, and a test using these data proves that the timetabling algorithm is adequate, all is well. All these considerations suggest a strategy for reducing the uncertainty about the project’s success before too much effort is wasted.

Likewise, in any project, technical feasibility should be assessed before making any serious financial commitment. Unfortunately, in the eagerness to deliver a product on schedule, such experiments may seem like a waste of time — but if omitted, a project runs the risk of being an expensive failure.

### ECONOMIC FEASIBILITY

Economic feasibility is a kind of cost-benefit analysis of the examined project, which assesses whether it is possible to implement it. This term means the assessment and analysis of a project's potential to support the decision-making process by objectively and rationally identifying its strengths, weaknesses, opportunities and risks associated with it, the resources that will be needed to implement the project, and an assessment of its chances of success. It consists of market analysis, economic analysis, technical and strategic analysis.

### OPERATIONAL FEASIBILITY

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

The operational feasibility assessment focuses on the degree to which the proposed development project fits in with the existing business environment and objectives with regard to development schedule, delivery date, corporate culture and existing business processes.

To ensure success, desired operational outcomes must be imparted during design and development. These include such design-dependent parameters as reliability, maintainability, supportability, usability, reducibility, disposability, sustainability, affordability and others. These parameters are required to be considered at the early stages of design if desired operational behaviour are to be realised . A system design and development requires appropriate and timely application of engineering and management efforts to meet the previously mentioned parameters. A system may serve its intended purpose most effectively when its technical and operating characteristics are engineered into the design. Therefore, operational feasibility is a critical aspect of systems engineering that needs to be an integral part of the early design phase.

## SYSTEM ANALYSIS

The third phase of software development is system analysis. Analysis involved a detailed study of the current system, leading to specifications of a new system. Analysis is a detailed study of various operations performed by a system and their relationships within and outside the system. During analysis, data are collected on the available files, decision points and transactions handled by the present system. Interviews, on-site observation and questionnaire are the tools used for system analysis. Using the following steps it becomes easy to draw the exact boundary of the new system under consideration:

* + - * Keeping in view the problems and new requirements
      * Workout the pros and cons including new areas of the system
      * All procedures, requirements must be analysed and documented in the form of detailed data flow diagrams (DFDs), data dictionary, logical data structures and miniature specifications. System Analysis also includes sub-dividing of complex process involving the entire system, identification of data store and manual processes.

The main points to be discussed in system analysis are:

* + - * Specification of what the new system is to accomplish based on the user requirements.
      * Functional hierarchy showing the functions to be performed by the new system and their relationship with each other.
      * Function network which are similar to function hierarchy but they highlight the functions which are common to more than one procedure.
      * List of attributes of the entities - these are the data items which need to be held about each entity (record)

### EXISTING SYSTEM

Now a days it is difficult to transfer files securely over internet. Commonly messages are send directly as text. Communication in well known forms greatly increases the risk of information being leaked in transit. Hacker can easily access the

confidential messages. So we cannot ensure that our messages are delivered at correct destination without any modification. Existing systems does not enhance individual privacy and also users want to type their messages. This process is very time consuming since in this globalized world no one has time to waste.

### LIMITATIONS

The existing system has some limitations that said as follows:

* In existing system we have to type the message.
* The basic and major drawbacks in the existing system are the speed of retrieval of data.
* The present system is time consuming.
* There are plenty of chances of duplicity of data and information.
* Updating is very tedious job.
* Large number of records is needed.
* Lack of security.

### PROPOSED SYSTEM

Hidden Audio Tracker is a hacker system that is it will help transferring media voice or sound format as image when we can’t able to transfer a file directly as media format because of its security constraints. This is a site that helps us to transfer files securely by implementing the technique called Steganography. Steganography is a process that involves hiding a message in an appropriate carrier for example, an image or an audio file. The carrier can then be sent to a receiver without anyone else knowing that it contains a hidden message. Avoiding communication in well-known forms reduces the risk of information being leaked in tr ansit. Another form of Steganography, called watermarking is used primarily for identification and entails embedding a unique piece of information within a medium without noticeably altering the medium. Steganography can also enhance individual privacy. This is effective only if the hidden communication is not detected. This proposed system is to provide a good, efficient method for hiding the data from hackers and sent to the destination in a safe manner. This proposed system will not change the size of the file even after encoding and also suitable for any type of audio file format. Encryption and Decryption techniques have been used to make the security system robust.

The project is being well made to tackle the security issues and user interface makes user to handle well the system and provides god communication between user and the system.In this project we can hide audio in an image file. Here audio file is converted to text and covered with an image. A user can view all images send by others in his inbox. Here the user can upload an audio and picture files, extract or split the image file to both image and audio. The user is also able to download file. A user can access this site across anywhere in the world.

### ADVANTAGES

* The proposed system increases the individual privacy.
* It improves the security.
* It is not essential for the users to type the message one can speak whch is uploaded as message.
* This proposed system saves time.
* User-friendly interfaces : The system provides friendly graphical user interfaces with efficient help and tips, which makes addition of records easier. Several validation routines are also provided which ensure error free addition of records. Moreover large quantity of data can be easily inserted with less time consumption.
* Consistency: The major problem that can be overcome is the problem of consistency. It ensures that changes made are available to all.

### CHALLENGES

* Availability of computer hardware and systems to use.
* Challenge of impersonation.
* Challenge of state continuity
* Level of IT literacy
* References

## SYSTEM REQUIREMENT SPECIFICATION

**SYSTEM MODULES**

It is a hacker system that is it will help transferring media voice or sound format as image when we can’t able to transfer a file directly as media format because of its security constrains. Now a day the secured transferring of files is the major problem, to avoid this problem we implement many technologies. Here one of these technologies is used for secured file transferring that is stegenography. Steganography is a means of obscuring data where secret messages are hidden inside computer files such as images, sound files,videos and even executable files so that ,no one except the sender and the receiver will suspect the existence of the stealth information in it. In this project we can hide Audio in an image file.

### Module description

Impose Module

 User can upload carrier image and audio file

 System Impose audio in image

 Send this to specified user

Extract Module

 Receiving data

 Extract Audio

 Download

### Impose Module:

In this module we upload both an audio and picture file. The system fetch the audio to the selected Picture file and when the function got finished, sound carried image file spontaneously save to the Server.

By using the given URL, a user can easily transfer the Audio fetched image file to any other user.

### Extract Module:

A user can view all images send by others in his inbox. Here the user can extract or split the Image file to both Image and Audio. The User is also able to download audio file, after the mentioned process got finished.

# DESIGN PHASE

## INTRODUCTION TO DATA FLOW DIAGRAM

A Data Flow Diagram is a network that describes the flow of data and processes that change, or transform, data throughout the system. This network is constructed by use a set of symbols that do not imply a physical implementation. It is a graphical tool for structured analysis of the system requirements. DFD models a system by using external entities from which data flows to a process, which transforms the data and creates, output-data-flows which go to other processes or external entities or files.

Data in files may also flow to processes as inputs.

There are various symbols used in a DFD. Bubbles represent the processes. Named arrows indicate the data flow. External entities are represented by rectangles. Entities supplying data are known as sources and those that consume data are called sinks. Data are stored in a data store by a process in the system. Each component in a DFD is labelled with a descriptive name. Process names are further identified with a number.

The Data Flow Diagram shows the logical flow of a system and defines the boundaries of the system. For a candidate system, it describes the

Input (source), outputs (destination), database (files) and procedures (data Flow), all in a format that meet the user’s requirements.

The main merit of DFD is that it can provide an overview of system requirements, what data a system would process, what transformations of data are done, what files are used, and where the results flow.

**Rules for constructing a Data Flow Diagram**

1. Arrows should not cross each other
2. Squares, circles and files must bear names.
3. Decomposed data flow squares and circles can have same time.
4. Choose meaningful names for data flow
5. Draw all data flows around the outside of the diagram.
   1. **DATA F LOW DIAGRAM**

Each component in a DFD is labelled with a descriptive name. Process name are further identified with number. Context level DFD is draw first. Then the process is decomposed into several elementary levels and is represented in the order of importance. A DFD describes what data flow (logical) rather than how they are processed, so it does not depend on hardware, software, and data structure or file organization. A DFD methodology is quite effective; especially when the required design is clear and the analyst need a notation language for communication.

A DFD methodology is quite effective; especially when the required design is clear and the analyst need a notation language for communication.

**Basic Data Flow Diagram Symbols**

A data flow is a route, which enables packets of data to travel from one point to another. Data may flow from a source to a process and from data store or process. An arrow line depicts the flow, with arrow head pointing in the direction of the flow.

Circles stands for process that converts data in to information. A process represents transformation where incoming data flows are changed into outgoing data flows.

A data store is a repository of data that is to be stored for use by a one or more process may be as simple as buffer or queue or sophisticated as relational database. They should have clear names. If a process merely uses the content of store and does not alter it, the

arrowhead goes only from the store to the process. If a process alters the details in the store then a double headed arrow is used.

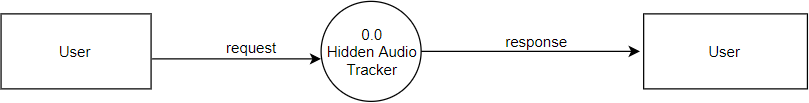
A source or sink is a person or part of an organization, which enters or receives information from the system, but is considered to be outside the contest of data flow model.

**: response**

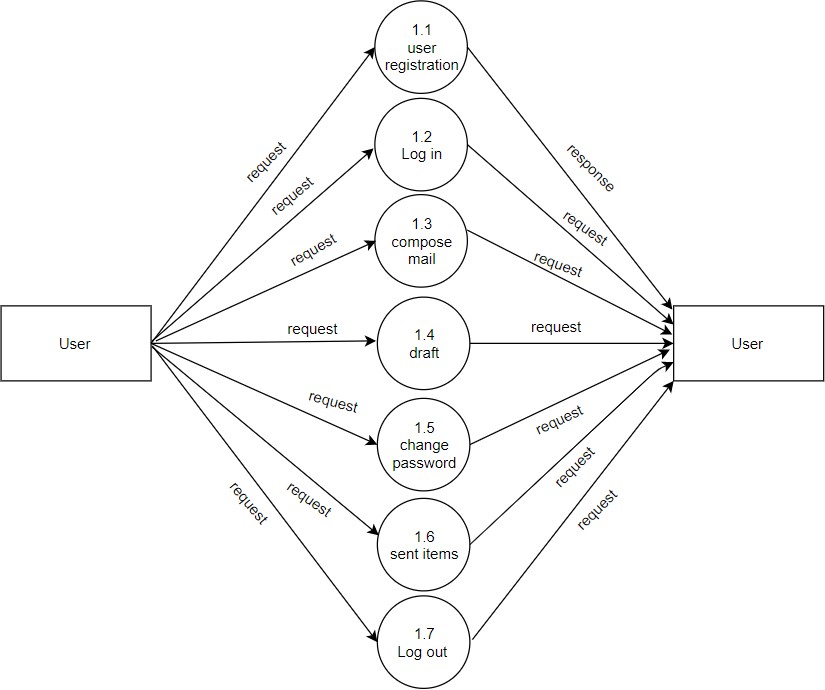
**:**

**: request**

### CONTEXT LEVEL DFD



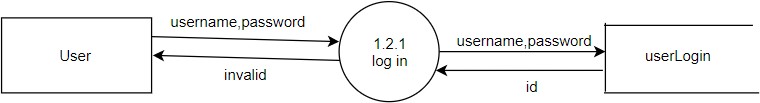
* + 1. **FIRST LEVEL DFD FOR USER**



### FIRST LEVEL DFD FOR USER REGISTRATION



* + 1. **FIRST LEVEL DFD FOR USER LOG IN**



### FIRST LEVEL DFD FOR COMPOSE MAIL



* + 1. **FIRST LEVEL DFD FOR SAVING DRAFTS**



### FIRST LEVEL DFD FOR CHANGE PASSWORD



* + 1. **FIRST LEVEL DFD FOR SENT ITEMS**



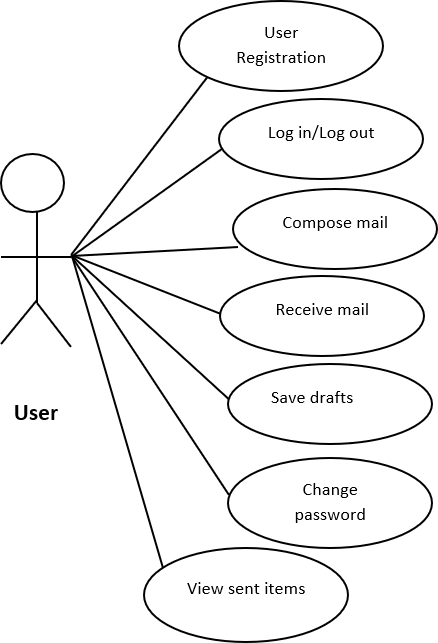
## USECASE DIAGRAM

A Use case diagram is a dynamic or behaviour diagram in UML. Use case diagram model the functionality of a system using actors and use cases. Use cases are set of actions, services, and functions that the system needs to perform. In this context, a “system” is something being developed or operated, such as web site. The “actors” are people or entities operating under defined roles within the system.

* System

.

* Use case
* Actor
* Realtionships
  + 1. **Usecase diagram for user**



## DATABASE DESIGN

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make information access easy, quick, inexpensive and flexible for the users. The general theme behind a database is to integrate all information. Database design is recognized as a standard of management information system and is available virtually for every computer system. In database design several specific objectives are considered:

* Ease of learning and use
* Controlled redundancy
* Data independence
* Accuracy and integrity
* Recovery from failure

A database is an integrated collection of data and provides centralized access to the data. Usually the centralized data managing the software is called RDBMS. The main significant difference between RDBMS and other DBMS is the separation of data as seen by the program and data has in direct access to stores device. This is the difference between logical and physical data.

## NORMALIZATION

Designing a database is complete task and the normalization theory is a useful aid in the design process. The process of normalization is concerned with transformation of conceptual schema into computer representation form. There will be need for most databases to grow by adding new attributes and new relations. The data will be used in new ways. Tuples will be added and deleted. Information stored may undergo updating also. New association may also be added. In such situations the performance of a database is entirely depend upon its design. A bad database design may lead to certain undesirable things like:

* + - 1. Repetition of information
      2. Inability to represent certain information
      3. Loss of information

To minimize these anomalies, Normalization may be used. If the database is in a normalized form, the data can be growing without, in most cases, forcing the rewriting application programs. This is important because of the excessive and growing cost of maintaining an organization’s application programs and its data from the disrupting effects of database growth. As the quality of application programs increases, the cost of maintaining the without normalization will rise to prohibitive levels. A normalized database can also encompass many related activities of an organization thereby minimizing the need for rewriting the applications of programs.

Thus, normalization helps one attain a good database design and there by ensures continued efficiency of database. Normalization theory is built around the concept of normal forms. A relation is said to be in normal form if it satisfies a certain specified set of constraints. For example, a relation is said to be in first normal form (1NF) if it satisfies the constraint that it contains atomic values only. Thus every normalized relation is in 1NF.Numerous normal forms have been defined. Code defined the first three normal forms.

All normalized relations are in 1NF, some 1NF relations are also in 2NF and some 2NF relations are also in 3NF.2NF relations are more desirable than 1Nf and 3NF are more desirable than 2NF. That is, the database designer should prefer 3NF than 1NF or 2NF.Normalization procedure states that a relation that is in some given normal form can be converted into a set of relations in a more desirable form. I can define this procedure as the successive reduction of a given collection of relations to some more desirable form. This procedure is reversible. That is, it is always possible to take the output from the procedure and convert them back into input. In this process, no information is lost. So it is also called “no loss decomposition”.

### First Normal Form

A relation is in first normal form (1NF) if and all its attributes are based on single domain. The objective of normalizing a table is to remove its repeating groups and ensure that all entries of the resulting table have at most single value.

### Second Normal Form

A table is said to be second Normal Form (2NF), when it is in 1NF and every attribute in the record is functionally dependent upon the whole key, and not just a part of the key.

### Third Normal Form

A table is in third Normal Form (3NF), when it is in 2NF and every non-key attribute is functionally dependent on just the primary key.

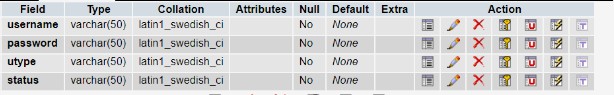
## TABLE STRUCTURE

Table is a collection of complete details about a particular subject. These data are saved in rows and Columns. The data of each Row are different units. Hence, rows are called RECORDS and Columns of each row are called FIELDS.

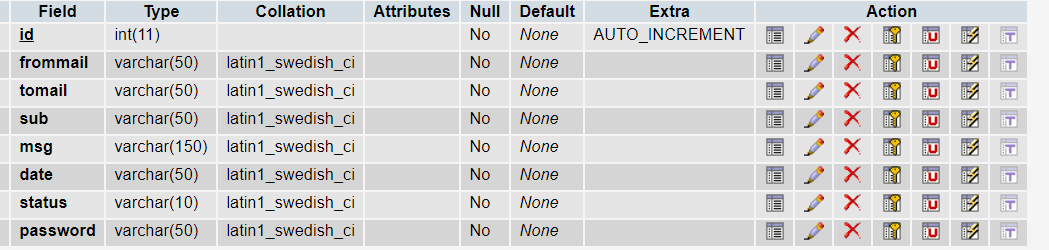
Data is stored in tables, which is available in the backend. The items and data, which are entered in the input, form id directly stored in this table using linking of database. I can link more than one table to input forms. I can collect the details from the different tables to display on the output. There are mainly 3 tables in our project. They are, 1. login

* 1. messages
  2. userreg

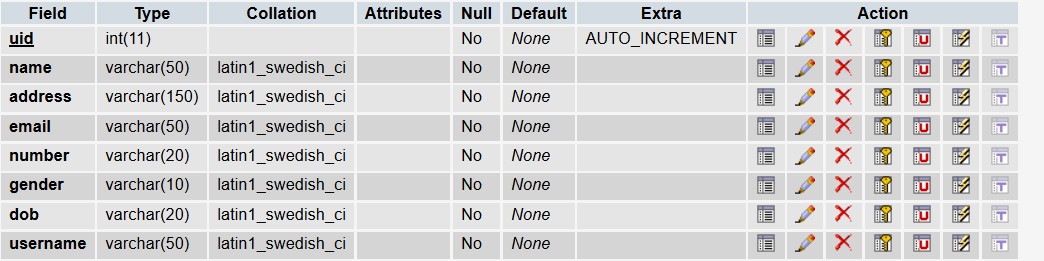
1. Log in



1. Messages



1. Userreg



## SYSTEM DESIGN

The system design is the most creative and challenging phase of system development life cycle. It is an approach for the creation of proposed system, in which the logic and details structure of the proposed system is designed, which will help the system coding. The most creative and challenging phase of the system development process is design phase it is a solution, how to approach to the creation of the proposed system. Design is the first step in the development of the engineered product is initiated only after a clear exposition of expected product is available. System Design is vital for efficient database management. It provides the understanding of procedural details necessary for implementing the system. A number of sub- systems is to be identified which constitute the whole system

### SYSTEM ARCHITECTURE

A system architecture or systems architecture is the conceptual model that defines the structure, behaviour, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures of the system, System architecture can comprise system components, the externally visible properties of those components, the relationships (e.g. the behaviour) between them. It can provide a plan from which products can be procured, and systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture; collectively these are called architecture description languages (ADLs).

The system architecture can best be thought of as a set of representations of an existing (or to be created) system. It is used to convey the informational content of the elements comprising a system, the relationships among those elements, and the rules governing those relationships. The architectural components and set of relationships between these components that architecture describes may consist of hardware, software, documentation, facilities, manual procedures, or roles played by organizations or people.

### MODULE DESIGN

Modular programming is a software design technique that emphasizes separating the functionality of a program into independent, interchangeable modules, such that each contains everything necessary to execute only one aspect of the desired functionality. Conceptually, modules represent a separation of concerns, and improve maintainability by enforcing logical boundaries between components. Two types of modular programming techniques. Top down and Bottom up Approaches

### TOP DOWN APPROACH

Top down programming means I start at the top and work towards down. In this I first consider the entire program and it is then sub divided in to less complex, smaller and easily manageable form until a stage is reach when further break down will serve no useful purpose. That is, to cut all the program in to a number of independent task and then cut task in to smaller sub task and so on. Until they are small enough to code easily. These task and sub task form the basic functions in the program. In top down programming the programmers write the main () first. That means all the major functions it will need. Later the programmers look at the requirements of each of these .

### BOTTOM UP APPROACH

Bottom up programming is exactly opposite top down programming. In this I start at bottom work towards top. That is the programmers write the lower level function first and then using this lower level functions they write the next higher level and so on.

In the basic elements are specified first, these elements are link together to form large programs and this is repeated until the complete program is formed. Advantages

* Reusability of code Disadvantages
* Less easier to implement compared to top down

## DEVELOPMENT PHASE

* 1. **SYSTEM ENVIRONMENT**

### SOFTWARE SPECIFICATION

* + - * Operating system : Windows XP/windows7/windows 8/Windows 10
      * Front end : DJANGO
      * Back end : MYSQL

### HARDWARE SPECIFICATION

* + - * Microprocessor : Pentium-4
      * RAM : 128 MB
      * Speed : 1.6 GHz
      * Hard Disk : 40 GB
      * Floppy Drive : 1.44 MB
      * Monitor : 15” Color
      * Key Board : 114 keys
      * CD Drive : 52
      * Mouse : Logitech Scrolling Mouse

## CODING

Coding is a list of step-by-step instructions that get computers to do what you want them to do. This step is also called programming phase. The performance of software design starts by using program code with appropriate programming language and developing error free executable programs in efficient manner. Coding is undertaken once the design phase is complete and the design documents have been successfully reviewed .Computer Coding is term used for writing codes & executing it for getting desired output. In this phase, every module identified and specified in the design document is independently coded and unit tested.

* The input to the coding phase is the design document.
* During the coding phase, various modules identified in the design document are coded according to the respective module specifications. In this phase, each module identified and specified in the design document is independently coded and unit tested.
* A coding standard gives a regular form to the codes written by different engineers.
* It provides sound understanding of the code.
* It encourages good programming practice.

**Urls.py**

from django.contrib import admin from django.urls import path from hatapp import views

urlpatterns = [

path('admin/', admin.site.urls), path('',views.guest\_home,name="Ghome"), path('reg/',views.guest\_reg,name="Greg"), path('login/',views.guest\_login,name="Glog"), path('uhome/',views.user\_home,name="Uhome"), path('uprofile/',views.user\_profile,name="Uprof"), path('compose88/',views.user\_compose,name="Ucompose"), path('compose/',views.edit,name='exeternal'), path('inbox/',views.user\_inbox,name='inboxda'), path('inbox1/',views.inbox1,name='inbox1da'), path('draft/',views.draft,name='inbox1da'), path('profileedit/',views.profileedit,name='profileedit'),

]

**Views.py**

from django.shortcuts import render,HttpResponseRedirect import pymysql

#import pyaudio import wave

import speech\_recognition as spreg import datetime

from datetime import date

from django.core.files.storage import FileSystemStorage #database connection db=pymysql.connect("localhost","root","","hat\_image") c=db.cursor()

# Create your views here. def guest\_home(request):

return render(request,'guest/guest\_home.html') def guest\_reg(request):

da=str(date.today()) print(da)

data=""

if request.POST: name=request.POST.get("name") address=request.POST.get("address") email=request.POST.get("email") number=request.POST.get("number") gender=request.POST.get("gender") dob=request.POST.get("dob") uname=request.POST.get("uname") password=request.POST.get("password") cpass=request.POST.get("cpassword") request.session["uname"]=uname if(cpass==password):

utype="user" status="pending"

c.execute(“insert into userreg (name,address,email,gender,dob,username) values ('"+ name +"','"+ address +"','"+ email +"','"+ number +"','"+ gender +"','"+ dob

+"','"+ uname +"')")

db.commit()

c.execute("insert into login values('"+ uname +"','"+ password +"','"+ utype

+"','"+ status +"')")

db.commit()

data="Successfully completed registartion" else:

data="Password donot match"

return render(request,'guest/guest\_reg.html',{"msg":data,"da":da}) def guest\_login(request):

data=""

if request.POST: uname=request.POST.get("uname") passw=request.POST.get("password")

c.execute("select \* from login where username='"+ uname +"' and password='"+ passw +"'")

log=c.fetchone() if log :

request.session["uname"]=log[0] if log[2]=='user':

return HttpResponseRedirect('/uhome') # elif log[2]=='admin':

# return HttpResponseRedirect('/ahome') else:

return HttpResponseRedirect('/login')

else:

return HttpResponseRedirect('/login')

return render(request,'guest/guest\_login.html',{"data":data}) def user\_home(request):

return render(request,'user/user\_home.html') def user\_profile(request):

name=request.session["uname"]

c.execute("select \* from userreg where username='"+ name +"'")

udata=c.fetchall() if request.POST: for i in udata: id=i[0]

request.session['id']=id

return HttpResponseRedirect('/profileedit/')

return render(request,'user/user\_profile.html',{"udata":udata}) def profileedit(request):

if request.session['id']:

c.execute("select \* from userreg where uid='"+ str(request.session['id']) +"'") udata=c.fetchall()

if request.POST: name=request.POST.get("name") address=request.POST.get("address") email=request.POST.get("email") number=request.POST.get("number") gender=request.POST.get("gender") dob=request.POST.get("dob") uname=request.POST.get("uname")

c.execute("update userreg set name='"+ name +"',address='"+ address

+"',email='"+ email +"',number='"+ number +"',gender='"+ gender +"',dob='"+ dob

+"',username='"+ uname +"' where uid='" + str(request.session["id"]) + "'") db.commit()

return HttpResponseRedirect('/uprofile/')

return render(request,'user/profileedit.html',{"udata":udata}) def user\_compose(request):

if 'rec' in request.POST: #record and get audio print("hi")

CHUNK = 1024

FORMAT = pyaudio.paInt16

CHANNELS = 2

RATE = 44100

RECORD\_SECONDS = 5

WAVE\_OUTPUT\_FILENAME = "sample\_audio.wav" p = pyaudio.PyAudio()

stream = p.open(format=FORMAT, channels=CHANNELS, rate=RATE,

input=True, frames\_per\_buffer=CHUNK)

print("\* recording") frames = []

for i in range(0, int(RATE / CHUNK \* RECORD\_SECONDS)): data = stream.read(CHUNK)

frames.append(data) print("\* done recording") stream.stop\_stream() stream.close() p.terminate()

wf = wave.open(WAVE\_OUTPUT\_FILENAME, 'wb') wf.setnchannels(CHANNELS) wf.setsampwidth(p.get\_sample\_size(FORMAT)) wf.setframerate(RATE) wf.writeframes(b''.join(frames))

wf.close()

sound\_file = 'sample\_audio.wav' recog = spreg.Recognizer()

with spreg.AudioFile(sound\_file) as source:

speech = recog.record(source) #use record instead of listning try:

text = recog.recognize\_google(speech) print('The file contains: ' + text)

except spreg.UnknownValueError: print('Unable to recognize the audio')

except spreg.RequestError as e:

print("Request error from Google Speech Recognition service; {}".format(e)) if 'send' in request.POST:

to=request.POST.get("to") com=request.POST.get("compose") fil=request.POST.get("file") pas=request.POST.get("pas") #file=requset.Files.get("file")

c.execute("insert into compose(from,to,messages,file,pas) values('"+ request.session["uname"] +"',,'"+ to +"','"+ com +"','"+ fil +"','"+ pas +"')")

db.commit()

if 'draft' in request.POST: s=''

return render(request,'user/user\_compose.html') def edit(request):

if 'sent' in request.POST and request.FILES.get("file"): frommail=request.session["uname"] data=request.POST.get("msg") to=request.POST.get("to") sub=request.POST.get("sub") pasw=request.POST.get("pass") myfile=request.FILES.get("file") fs=FileSystemStorage() filename=fs.save(myfile.name , myfile) uploaded\_file\_url = fs.url(filename) date=datetime.datetime.now()

status="sent"

c.execute("insert into

messages(frommail,tomail,sub,msg,date,status,password,path) values('"+ frommail

+"','"+ to +"','"+ sub +"','"+ data +"','"+ str(date) +"','"+ status +"','"+ pasw +"','"+ uploaded\_file\_url +"')")

db.commit()

if 'save' in request.POST and request.FILES.get("file"): frommail=request.session["uname"] data=request.POST.get("msg") to=request.POST.get("to") sub=request.POST.get("sub") pasw=request.POST.get("pass") date=datetime.datetime.now() myfile=request.FILES.get("file") fs=FileSystemStorage() filename=fs.save(myfile.name , myfile) uploaded\_file\_url = fs.url(filename)

status="save"

c.execute("insert into

messages(frommail,tomail,sub,msg,date,status,password,path) values('"+ frommail

+"','"+ to +"','"+ sub +"','"+ data +"','"+ str(date) +"','"+ status +"','"+ pasw +"','"+ uploaded\_file\_url +"')")

db.commit()

return render(request,'user/edit.html') def user\_inbox(request):

sent='sent' name=request.session["uname"]

c.execute("select \* from messages where tomail='"+ request.session["uname"] +"' and status='"+ sent +"' order by date desc ")

print("select \* from messages where tomail='"+ request.session["uname"] +"' and status='"+ sent +"' order by date desc ")

co=c.fetchall() if request.GET:

msgidd=request.GET.get("msgid")

if msgidd: request.session["msgid"]=msgidd

return HttpResponseRedirect("/inbox1/")

return render(request,'user/user\_inbox.html',{"udata":co}) def inbox1(request):

co=""

c.execute("select path from messages where id='"+ request.session["msgid"] +"'") path=c.fetchone()

path=path[0]

if request.POST: pass1=request.POST.get("pass")

c.execute("select \* from messages where id='"+ request.session["msgid"] +"' and password='"+ pass1 +"' order by date desc")

co=c.fetchall()

return render(request,'user/inbox1.html',{"udata":co,"path":path}) def draft(request):

save='save'

c.execute("select \* from messages where status='"+ save +"' and frommail='"+ request.session["uname"] +"'")

co=c.fetchall() if request.GET:

msgidd=request.GET.get("msgid") if msgidd:

request.session["msgid"]=msgidd

return HttpResponseRedirect("/inbox1/")

return render(request,'user/user\_draft.html',{"udata":co})

## TESTING AND IMPLEMENTATION

* 1. **TESTING**

It is the process of evaluating a system or its components with the intent to find that whether it satisfies the related requirements. In computer hardware and software development, testing is used at key checkpoints in it’s done with executing the software overall process to identify whether objectives are being met. The process or method of finding errors in a software application or program so that the application functions according to the end user's requirement is called software testing. Software testing is used to assess the feature of a software item. Testing process ensures the quality of the product. Testing is the process that should be done during the development of software. In other words software testing is a verification and validation process.

## BLACK BOX TESTING

Black box testing is also called functional testing. It is a software testing method and is used to test the software without knowing the internal structure of code or program. Internal system design is not considered in this type of testing. This type of testing is mainly focus on the software requirements and specifications. In the proposed work black box testing is used for the following requirements:

* + - * Login
      * New User Registration
      * Edit Details

## Advantages of Black Box Testing are:

* + - * The designer and the tester are independent of each other because testing is unbalanced.
      * The tester does not need knowledge of any specific programming languages like java ,c++...
      * Test is done from the point of view of the user.
      * Code access not required Disadvantages of Black Box Testing are:
      * The test can be redundant if the software designer has already run a test case
      * Test cases are difficult to design, without having clear functional specifications
      * It is difficult to identify all possible inputs in limited testing time. So writing test cases are difficult to design.

## WHITE BOX TESTING

White box testing is also known as structural testing and glass box testing. This testing is related to the knowledge of the internal logic of an application’s code. It is also called open box testing. Internal software and code working must be known for this type of testing. Tests are related to the coverage of code statements, branches, paths, conditions. In proposed work white box testing is used for the following requirements:

* + - * Programming code
      * Database Accessing

## Advantages of White box testing are:

* + - * The testing of the software no need to wait for the GUI
      * It helps in optimizing the code
      * Beneficent side-effects
      * Provide stability and usability of the test cases.
      * It helps in removing the extra lines of code, which can bring in hidden defects.
      * Is done with executing the software

## Disadvantages of white box testing are:

* + - * Expensive
      * It takes more time for the tester to develop the test cases. • Test cases are a waste if changes in the implementation code are done frequently.

## UNIT TESTING

Unit is the smallest testable part of software. Unit testing is used to validate that individual units of source code are working properly. In object-oriented programming, the smallest unit is a method and it contain a base/super class, abstract class or derived/child class but in procedural programming language a unit may be an individual program, function, procedure, etc., while The main advantage of the unit testing is used to improve the quality of code and save the tester's time and effort.

## SYSTEM TESTING

System testing is the testing and is used to ensure that by putting the software in different environments it still works. It is done with executing the software system testing the application is working correctly from the point of view of a user. The main purpose of this system testing is to evaluate the system’s compliance with the specified requirements. Whole system is tested as per the requirements. Black-box type testing that is related to overall requirements specifications, covers all combined parts of a system.

## USER ACCEPTANCE TESTING

Acceptance testing is to ensure that the delivered product meets the expectations of the user. It is belongs to the class of black box testing. The goal of acceptance testing is to verify that the software is done with executing the software ready and can be used by the end-users to perform those functions and tasks for the software development. This type of testing is done to verify if system meets the customer specified requirements. User or customers do this testing to determine whether to accept application.

## VALIDATION TESTING

Validation is the process of evaluating the final software product that correctly identifies the customer expectations and requirements. Validation Testing checks that the product correctly meets the customer needs. This testing can also be defined as to demonstrate that the product fulfils its intended use when deployed on suitable environment. This testing is done with executing the software.

## TEST CASES

Test Case Id : TC1

Tests Used : Black Box Testing/White Box Testing Correct Data : First Name :Neenamol

Last Name :Joseph

Email [:neena10@gmail.com](mailto:neena10@gmail.com)

Password : neena

Pincode 685619

Contact number 9961166836

Function : New user registration

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sl  No: | Step | Test Data | Expected  Result | Actual  Result | Status |
| 1 | Enter correct registration Details And  submit | Registration Information | Registered Successfully | Registered Successfully | Success |
| 2 | Enter registration Details And press back button | Registration Information | Cancel all details and staying on the same  registration page | Cancel all  details and staying on the same registration  page | Success |
| 3 | Enter email without @  symbol | Email:anugmail.com | Invalid email id | Invalid email id | Success |
| 4 | Enter pincode with less  than or  greater than 6 characters | Pincode:68561 | Invalid pincode | Invalid pincode | Success |
| 5 | Enter contact number with less  than or greater than 10  characters | Contact number:99611668361 | Invalid contact number | Invalid contact number | Success |
| 4 | Password and confirm password mismatch. click Submit  button | Password and confirm password mismatch.click Submit button / cancel button | Password mismatch | Password mismatch | Success |

Test Case1

**Test Case Id :** TC2

**Tests Used :** Black Box Testing/White Box Testing

**Correct Data :** Username: [anijamyladiyil@gmail.com](mailto:anijamyladiyil@gmail.com)

Password: 12345678

**Function :** Registered user Login

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sl No: | Step | Test Data | Expected Result | Actual Result | Status |
| 1 | Enter Username only then click sign in button | Username: [anijamyladyil@gmail.com](mailto:anijamyladyil@gmail.com) | Please Enter Username And Password Correctly | Please Enter Username And Password Correctly | Success |
| 2 | Enter incorrect Username or Password | Username: [anupriya@gmail.com](mailto:anupriya@gmail.com) Password:1234 | Wrong Username or password | Wrong Username or password | Success |
| 3 | Enter Username and Password correctly | Username: [anijamyladiyil@gmail.com](mailto:anijamyladiyil@gmail.com) Password:123 | Valid username and password,go to home page. | Valid username and password,go to home page. | Success |
| 4 | Enter Username and Password correctly | Username:admin Password: admin | Valid username and password,go to main page of Admin | Valid username and password,go to main page of Admin | Success |

Test Case 2

**Test Case No** :TC3

**Tests Used** :**White**box **Testing/Blackbox Testing Function** :Edit Details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sl No: | Step | Test Data | Expected Result | Actual Result | Status |
| 1 | Edit user details and click update/cancel button | User details Management | Message “Updated Successfully” | Message “Updated Successfully” | Success |

Test Case 3

**Test Case No** :TC4

### Tests Used :Whitebox Testing/Blackbox Testing Function :verify user

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Si No: | Step | Test Data | Expected Result | Actual Result | Status |
| 1 | Check user details and click Go button | User registeration details | Add New user to database | Add New user to database | Success |
| 2 | Check user details and click Cancel button | User registeration details | Delete User | Delete User | Success |

Test Case 4

## SCREEN LAYOUTS

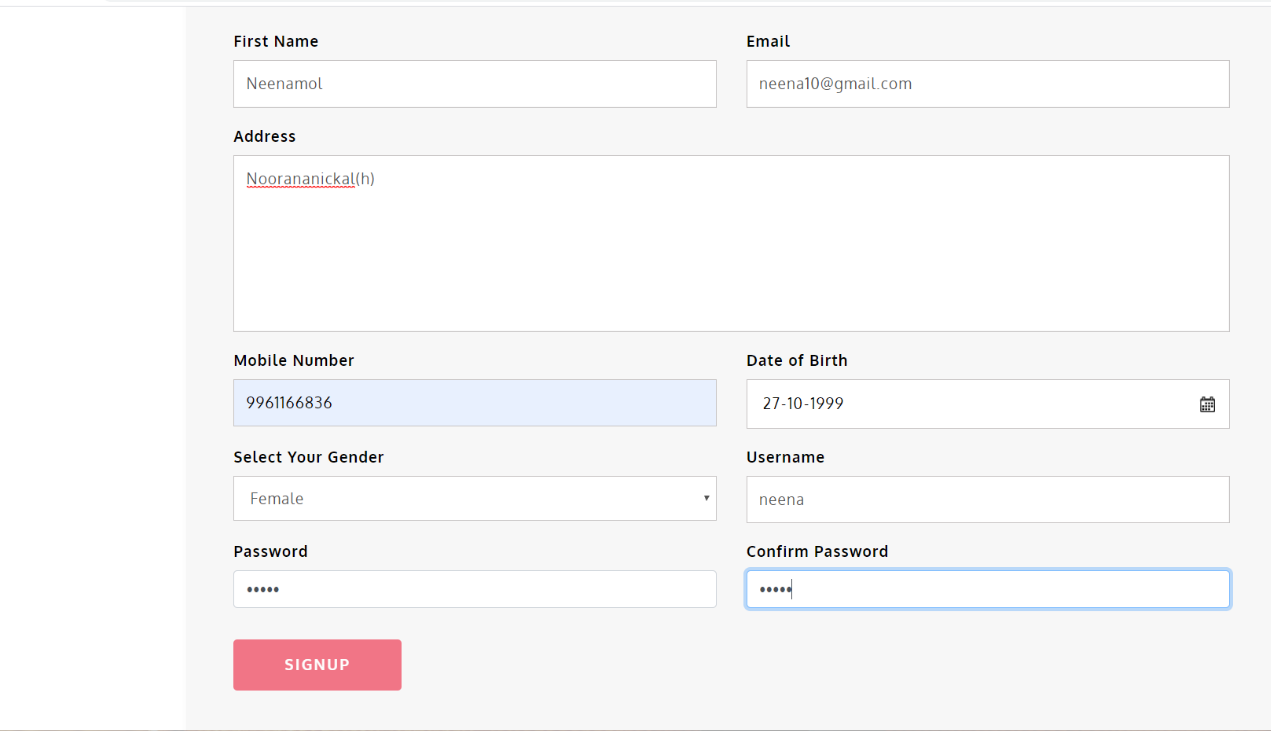
* 1. **FORM DESIGN**

### HOME PAGE

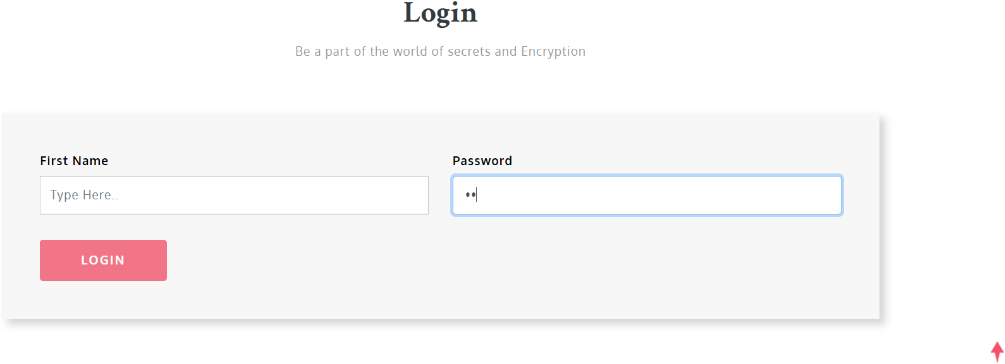


* + 1. **REGISTRATION PAGE**

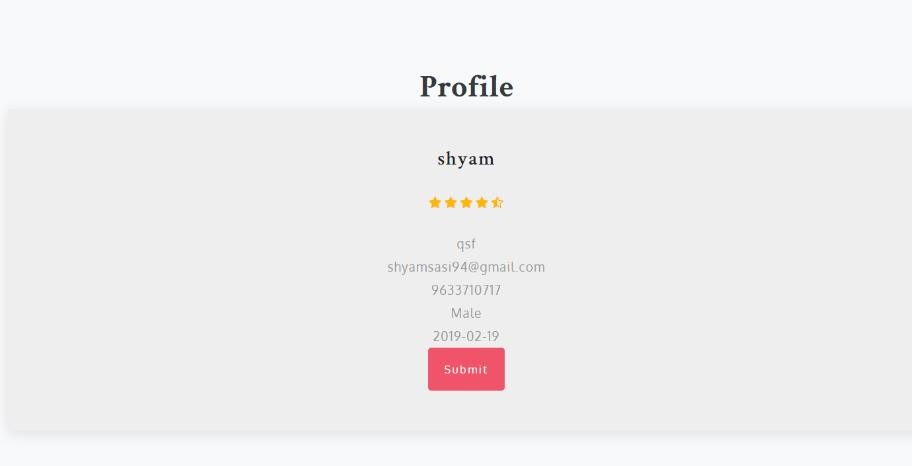
Registration page



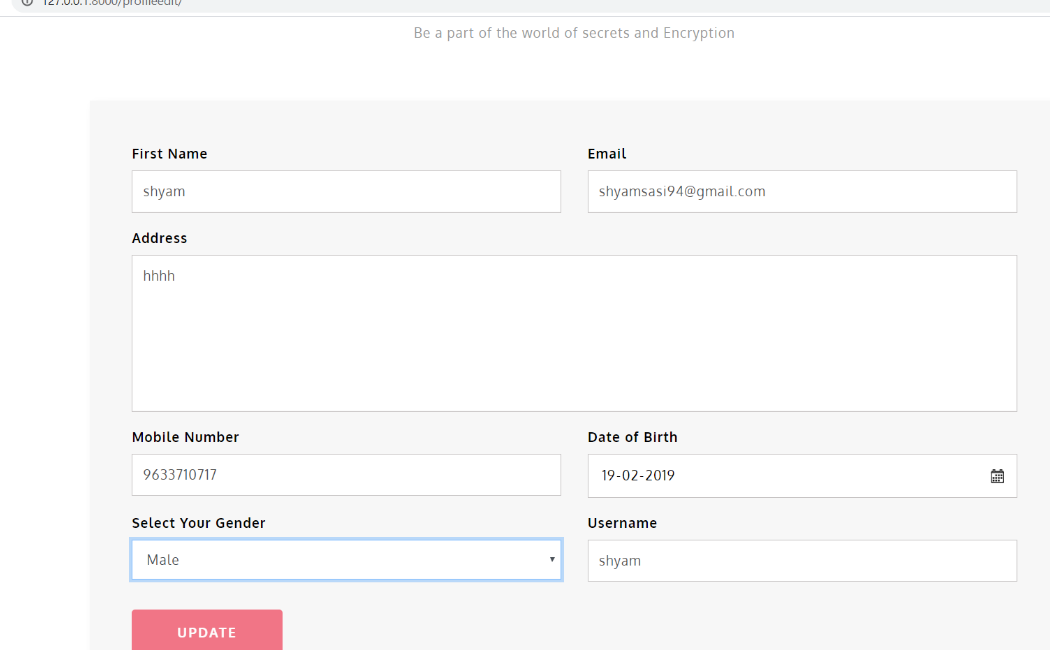
### LOGIN PAGE



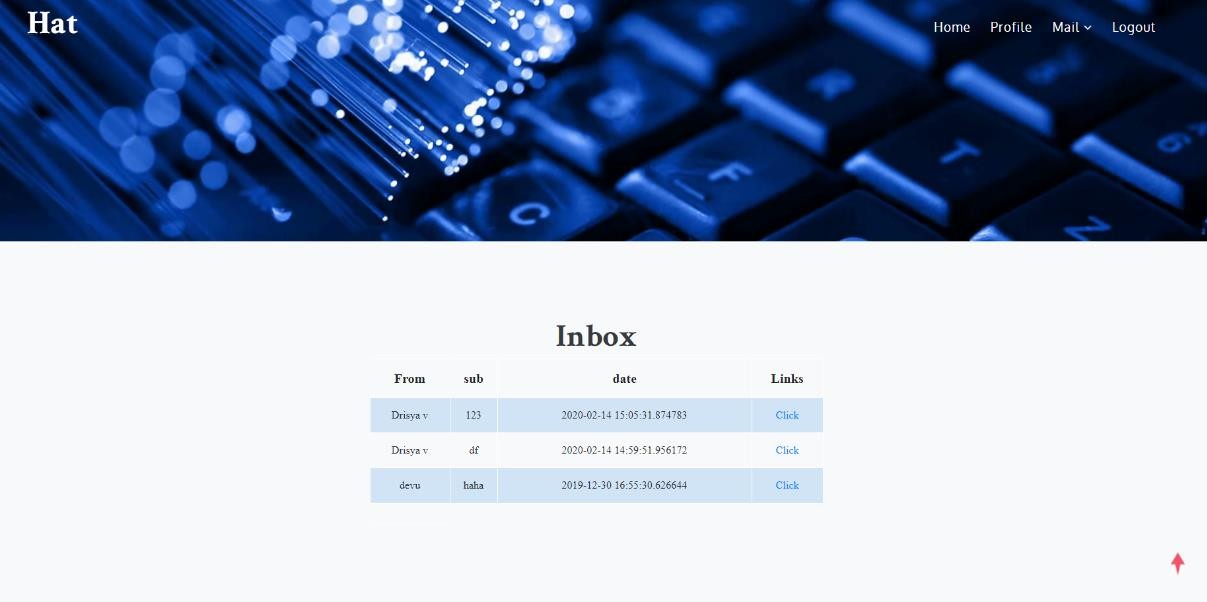
* + 1. **VIEWING PROFILE**



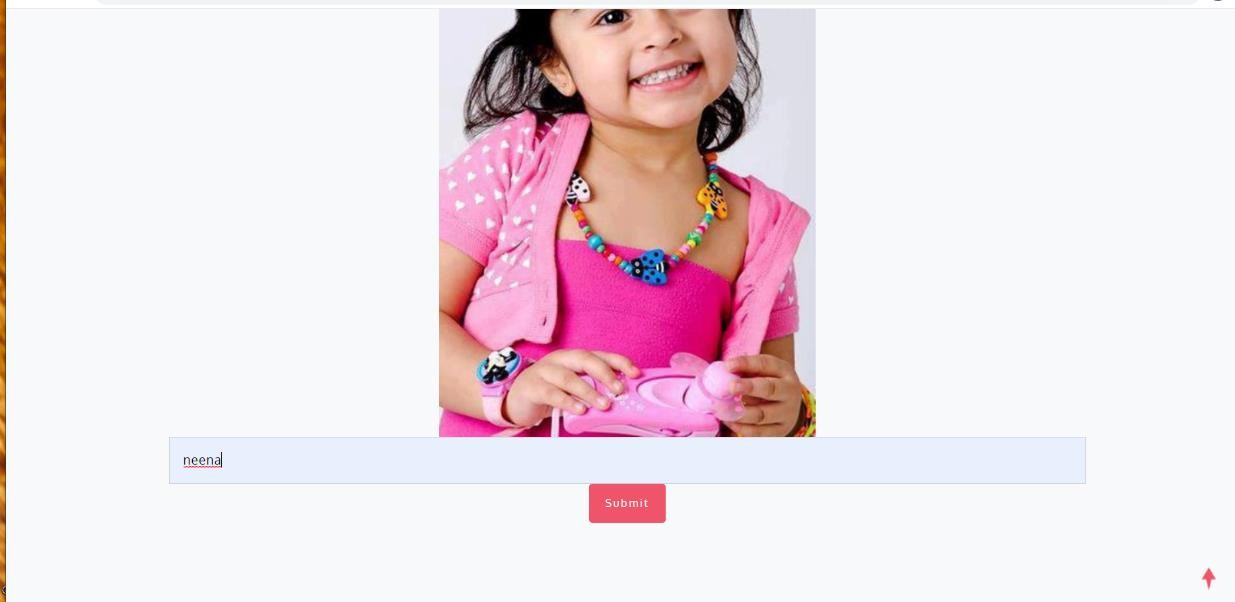
### UPDATING PROFILE



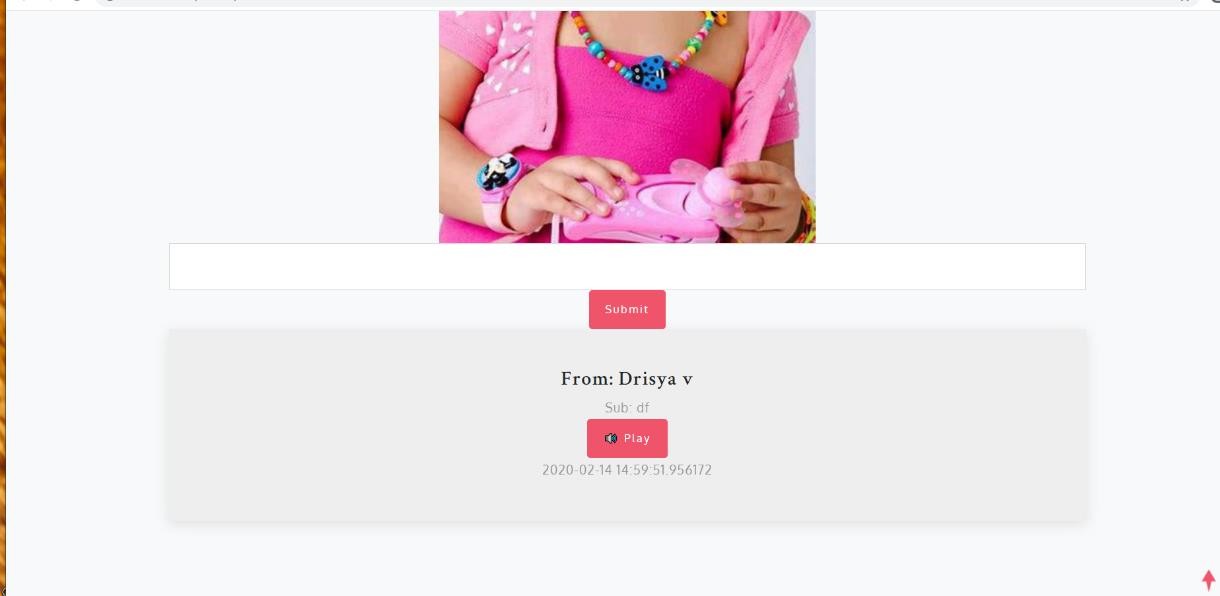
* + 1. **VIEWING INBOX**



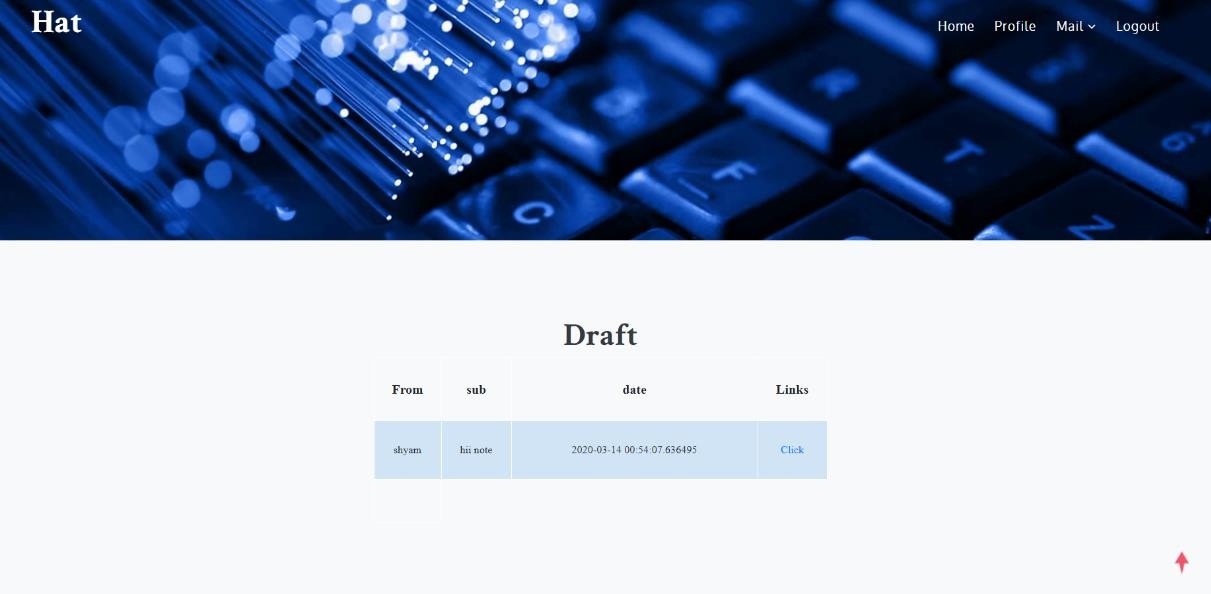
### ENTERING PASSWORD FOR ENCRYPTED AUDIO IN THE IMAGE



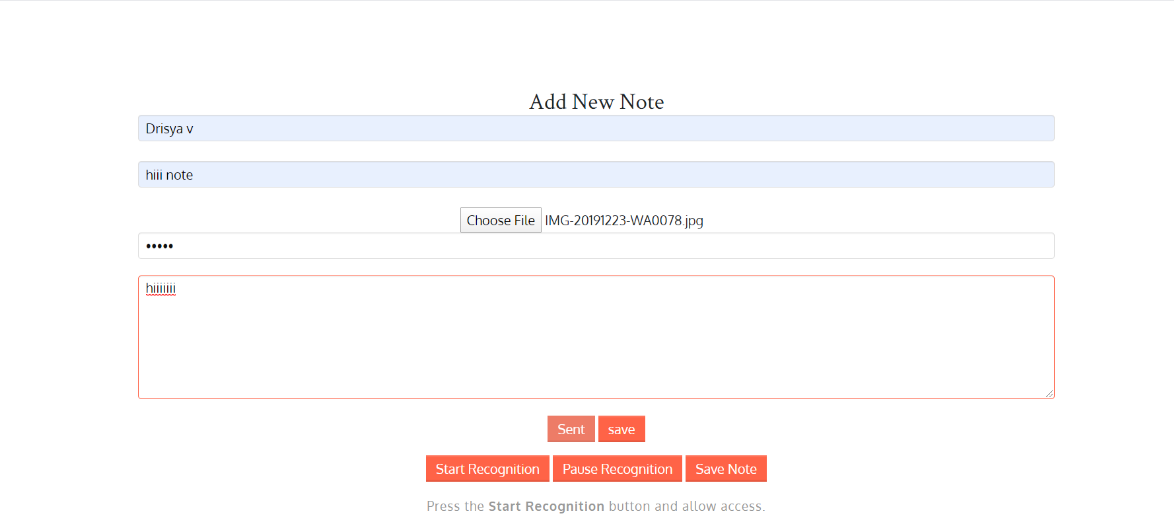
* + 1. **DECRYPTED AUDIO**



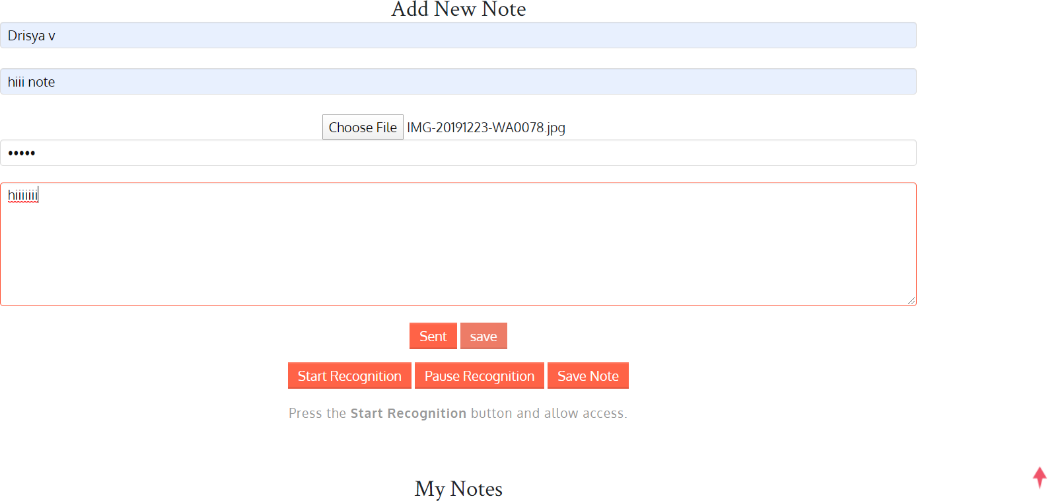
### DRAFTS



* + 1. **COMPOSE MAIL**



* + 1. **SAVING DRAFTS**



# CONCLUSION AND FUTURE WORK

## CONCLUSION

The project “HIDDEN AUDIO TRACKER” is developed to improve the privacy and security of sending and reciving files. it has been developed in .NET. Keeping in mind the specification of the system, all the specification of the proposed system has developed and tested with real data and everything worked successfully.

It is an attempt to overcome the insecurity and lack of individual privacy while sending a person or highly confidential file.it is less time consuming.

## ADVANTAGE OF PROJECT

* + - It provide better and efficient service
    - Greater accuracy
    - Increased reliability

## LIMITATIONS OF PROJECT

Hidden Audio Tracker is only useful for the systems which have same hacker system.The password is sends in manual. But it can be also hack. So these are the great limitations of Hidden Audio Tracker.

## FUTURE WORK

Now Hidden Audio Tracker implemented only on hacker systems. Later it can implement on any type of communication mechanisms and can also increase the security of the system.

## BIBLIOGRAPHY

* + - Pankaj Jalote, ― An Integrated approach to Software Engineering
    - Roger S Pressman – Software Engineering
    - Abraham Silberschatz, ― Database System Concepts
    - Ramez Elmasri , Shamkant &B.Navathe― Introduction to Data Base Management system.
    - David Beazley,Brian K. Jones – Python Cookbook

### WEBLIOGRAPHY

* + - <http://www.wikipedia.com/>
    - <http://www.stackoverflow.com/>
    - <http://w3schools.com/>
    - <http://www.tutorialpoint.com/>
    - <http://www.w3layouts.com/>