With this system it is tried to achieve making a better communication between police and community. While people can make easy reporting to officers, in reverse police can benefit from reported missing people, which reduces their work instead of doing it manually by themselves.

INTRODUCTION

Project Background

The project title is "Online Crime Database Management System ". Main features of this software are reporting online crimes, complaints, and missing people.

ANALYSIS

System Analysis

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. System analysis is an important phase of any system development process. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

Existing System

In the existing system only a crime can be reported, also it has more workload for the authorized person, but in the case of Proposed System, the user can register the site and send the first information report or complaint about a particular situation or person.

Drawbacks of Existing System

More man power.

Time consuming.

Consumes large volume of paper work.

Needs manual calculations.

No direct role for the higher officials.

To avoid all these limitations and make the working more accurately the system needs to be computerized in a better way.

Proposed System

The aim of proposed system is to develop a system of improved facilities. The proposed system can overcome all the limitations of the existing system. The existing system has several disadvantages and many more difficulties to work well. The proposed system tries to eliminate or reduce these difficulties up to some extent. The proposed system will help the user to reduce the workload and mental conflict.

Expected Advantages of Proposed System

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. It has got following features;

Minimize manual data entry.

Minimum time needed for the various processing.

Greater efficiency.

Better service.

User friendliness and interactive.

Minimum time required.

Feasibility Study

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provide the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. The following are its features:

Technical Feasibility

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

Technical issues raised during the investigation are:

Does the existing technology sufficient for the suggested one?

Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project is developed within latest technology. Through the technology may become obsolete after some period of time, due to the fact that never version of same software supports older versions, the system may still be used. So there are minimal constraints involved with this project. The system has been developed using Java the project is technically feasible for development.

Economical Feasibility

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

The costs conduct a full system investigation.

The cost of the hardware and software.

The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development.

BEHAVIORAL FEASIBILITY

This includes the following questions:

Is there sufficient support for the users?

Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioural aspects are considered carefully and conclude that the project is behaviourally feasible.

DESIGN

Introduction

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term "design" is defined as "the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization". It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

Input Design

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

What data should be given as input?

How the data should be arranged or coded?

Methods for preparing input validations and steps to follow when error occur.

Output Design

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

Select methods for presenting information.

Create document, report, or other formats that contain information produced by the system.

Database Design

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.

The database design is a two level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.

In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

Data Integrity

Data independence

Normalization is the process of decomposing the attributes in an application, which results in a set of tables with very simple structure. The purpose of normalization is to make tables as simple as possible. Normalization is carried out in this system for the following reasons.

To structure the data so that there is no repetition of data , this helps in saving.

To permit simple retrieval of data in response to query and report request.

To simplify the maintenance of the data through updates, insertions, deletions.

To reduce the need to restructure or reorganize data which new application requirements arise.

ERD

C:UserskerimAppDataLocalMicrosoftWindowsTemporary Internet FilesContent.Worderd\_Page\_1.jpg

Entity Relationship Diagram

Software Environments

Java Server Pages (JSP)

Java Server Pages technology is the Java platform technology for building applications containing dynamic Web content such as HTML, and XML. The Java Server Pages technology enables the authoring of Web pages that create dynamic content easily but with maximum power and flexibility.

The Java Server Pages technology offers a number of advantages:

Write Once, Run Anywhere properties

The Java Server Pages technology is platform independent, both in its dynamic Web pages, its Web servers, and its underlying server components. You can author JSP pages on any platform, run them on any Web server or Web enabled application server, and access them from any Web browser. You can also build the server components on any platform and run them on any server.

Reuse of components and tag libraries

The Java Server Pages technology emphasizes the use of reusable components such as: JavaBeans components, Enterprise JavaBeans components and tag libraries. These components can be used in interactive tools for component development and page composition. This saves considerable development time while giving the cross platform power and flexibility of the Java programming language and other scripting languages.

Separation of dynamic and static content

The Java Server Pages technology enables the separation of static content from dynamic content that is inserted into the static template. This greatly simplifies the creation of content. This separation is supported by beans specifically designed for the interaction with server-side objects.

Support for scripting and actions

The Java Server Pages technology supports scripting elements as well as actions. Actions permit the encapsulation of useful functionality in a convenient form that can also be manipulated by tools; scripts provide a mechanism to glue together this functionality in a per-page manner.

JSP architecture

JSPs are built on top of SUN's servlet technology. JSPs are essential an HTML page with special JSP tags embedded. These JSP tags can contain Java code. The JSP file extension is .jsp rather than .htm or .html. The JSP engine parses the .jsp and creates a Java servlet source file. It then compiles the source file into a class file; this is done the first time and this why the JSP is probably slower the first time it is accessed. Any time after this the special compiled servlet is executed and is therefore returns faster.

Java Script

JavaScript is a programming language that allows scripting of events, objects, and actions to create Internet applications. A website development environment that will allow the creation of Interactive Web Pages. The coding techniques capable of accepting a client's requests and processing these requests.

The web site development environment should also provide the facility for 'validating' user input. With JavaScript, forms are a consideration in nearly every page you design. Capturing user requests is traditionally done via a 'form'. So the web site needs to have facilities to create forms. Text fields and text areas can dynamically change in response to user responses.

TOMCAT

Tomcat is a servlet container and Java Server Pages implementation it may be used stand alone ,or in conjunction with several popular web servers .Tomcat is a security update release. This release closes a whole that potentially allowed access to resources protected by a in web.xml. Installing and Running Tomcat requires a Java Runtime Environment (JRE).Conformant to JRE including any Java2 platform system. If one wishes to develop applications you will need a java compiler , such as the one included in a java development kit environment including JDKs conformant with Java2.

CODING

MYSQL Database

In our database there are 8 tables;

Tables of Crime\_DBMS

Tables and Screen Shots

-- -----------------------------------------------------

-- Table `Crime\_DBMS`.`user`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `Crime\_DBMS`.`user` (

`user\_id` INT NOT NULL AUTO\_INCREMENT ,

`frst\_name` VARCHAR(45) NOT NULL ,

`surname` VARCHAR(45) NOT NULL ,

`dob` DATE NOT NULL ,

`address` VARCHAR(45) NOT NULL ,

`post\_code` VARCHAR(7) NOT NULL ,

`email` VARCHAR(45) NOT NULL ,

`phone` VARCHAR(20) NULL ,

PRIMARY KEY (`user\_id`) ,

UNIQUE INDEX `user\_id\_UNIQUE` (`user\_id` ASC) )

ENGINE = InnoDB;

"User"Table

-- -----------------------------------------------------

-- Table `Crime\_DBMS`.`admin`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `Crime\_DBMS`.`admin` (

`admin\_id` INT NOT NULL AUTO\_INCREMENT ,

`frst\_name` VARCHAR(45) NOT NULL ,

`surname` VARCHAR(45) NOT NULL ,

`bage\_num` VARCHAR(10) NOT NULL ,

`area` VARCHAR(45) NOT NULL ,

PRIMARY KEY (`admin\_id`) ,

UNIQUE INDEX `admin\_id\_UNIQUE` (`admin\_id` ASC) )

ENGINE = InnoDB;

"Admin" Table

-- -----------------------------------------------------

-- Table `Crime\_DBMS`.`fir\_report`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `Crime\_DBMS`.`fir\_report`(

`fir\_id` INT NOT NULL AUTO\_INCREMENT ,

`description` VARCHAR(100) NOT NULL ,

`details\_of\_suspect` VARCHAR(50) NOT NULL ,

`location` VARCHAR(45) NOT NULL ,

`date` DATETIME NOT NULL ,

`user\_user\_id` INT NOT NULL ,

`admin\_admin\_id` INT NOT NULL ,

PRIMARY KEY (`fir\_id`) ,

UNIQUE INDEX `fir\_id\_UNIQUE` (`fir\_id` ASC) ,

INDEX `fk\_fir\_report\_user1` (`user\_user\_id` ASC) ,

INDEX `fk\_fir\_report\_admin1` (`admin\_admin\_id` ASC) ,

CONSTRAINT `fk\_fir\_report\_user1`

FOREIGN KEY (`user\_user\_id` )

REFERENCES `Crime\_DBMS`.`user` (`user\_id` )

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_fir\_report\_admin1`

FOREIGN KEY (`admin\_admin\_id` )

REFERENCES `Crime\_DBMS`.`admin` (`admin\_id` )

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

"Fir\_Report" Table

-- -----------------------------------------------------

-- Table `Crime\_DBMS`.`complaint\_report`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `Crime\_DBMS`.`complaint\_report` (

`cmplnt\_id` INT NOT NULL AUTO\_INCREMENT ,

`description` VARCHAR(100) NOT NULL ,

`address` VARCHAR(45) NOT NULL ,

`date` DATETIME NOT NULL ,

`user\_user\_id` INT NOT NULL ,

`admin\_admin\_id` INT NOT NULL ,

PRIMARY KEY (`cmplnt\_id`) ,

UNIQUE INDEX `cmplnt\_id\_UNIQUE` (`cmplnt\_id` ASC) ,

INDEX `fk\_complaint\_report\_user` (`user\_user\_id` ASC) ,

INDEX `fk\_complaint\_report\_admin1` (`admin\_admin\_id` ASC) ,

CONSTRAINT `fk\_complaint\_report\_user`

FOREIGN KEY (`user\_user\_id` )

REFERENCES `Crime\_DBMS`.`user` (`user\_id` )

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_complaint\_report\_admin1`

FOREIGN KEY (`admin\_admin\_id` )

REFERENCES `Crime\_DBMS`.`admin` (`admin\_id` )

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

"Complaint\_Report" Table

-- -----------------------------------------------------

-- Table `Crime\_DBMS`.`missing\_report`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `Crime\_DBMS`.`missing\_report` (

`missing\_id` INT NOT NULL AUTO\_INCREMENT ,

`mis\_per\_fname` VARCHAR(45) NOT NULL ,

`mis\_per\_surname` VARCHAR(45) NOT NULL ,

`sex` VARCHAR(4) NOT NULL ,

`age` INT NOT NULL ,

`height` VARCHAR(45) NULL ,

`mis\_per\_address` VARCHAR(45) NOT NULL ,

`last\_seen\_place` VARCHAR(45) NOT NULL ,

`last\_seen\_date` DATETIME NOT NULL ,

`user\_user\_id` INT NOT NULL ,

`admin\_admin\_id` INT NOT NULL ,

PRIMARY KEY (`missing\_id`) ,

UNIQUE INDEX `missing\_id\_UNIQUE`(`missing\_id` ASC),

INDEX `fk\_missing\_report\_user1`(`user\_user\_id` ASC),

INDEX `fk\_missing\_report\_admin1` (`admin\_admin\_id` ASC) ,

CONSTRAINT `fk\_missing\_report\_user1`

FOREIGN KEY (`user\_user\_id` )

REFERENCES `Crime\_DBMS`.`user` (`user\_id` )

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_missing\_report\_admin1`

FOREIGN KEY (`admin\_admin\_id` )

REFERENCES `Crime\_DBMS`.`admin` (`admin\_id` )

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

"Missing\_Report" Table

-- -----------------------------------------------------

-- Table `Crime\_DBMS`.`most\_wanted`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `Crime\_DBMS`.`most\_wanted` (

`mstwntd\_id` INT NOT NULL AUTO\_INCREMENT ,

`frst\_name` VARCHAR(45) NOT NULL ,

`surname` VARCHAR(45) NOT NULL ,

`age` INT NOT NULL ,

`nationality` VARCHAR(45) NOT NULL ,

`user\_user\_id` INT NOT NULL ,

`admin\_admin\_id` INT NOT NULL ,

PRIMARY KEY (`mstwntd\_id`) ,

UNIQUE INDEX `mstwntd\_id\_UNIQUE` (`mstwntd\_id` ASC) ,

INDEX `fk\_most\_wanted\_user1` (`user\_user\_id` ASC) ,

INDEX `fk\_most\_wanted\_admin1` (`admin\_admin\_id` ASC) ,

CONSTRAINT `fk\_most\_wanted\_user1`

FOREIGN KEY (`user\_user\_id` )

REFERENCES `Crime\_DBMS`.`user` (`user\_id` )

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_most\_wanted\_admin1`

FOREIGN KEY (`admin\_admin\_id` )

REFERENCES `Crime\_DBMS`.`admin` (`admin\_id` )

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `Crime\_DBMS`.`prisoner`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `Crime\_DBMS`.`prisoner` (

`prisoner\_id` INT NOT NULL AUTO\_INCREMENT ,

`frst\_name` VARCHAR(45) NOT NULL ,

`surname` VARCHAR(45) NOT NULL ,

`age` INT NOT NULL ,

`address` VARCHAR(45) NOT NULL ,

`date\_entr` DATE NOT NULL ,

`date\_rls` DATE NOT NULL ,

`user\_user\_id` INT NOT NULL ,

`admin\_admin\_id` INT NOT NULL ,

PRIMARY KEY (`prisoner\_id`) ,

UNIQUE INDEX `prisoner\_id\_UNIQUE`(`prisoner\_id` ASC),

INDEX `fk\_prisoner\_user1` (`user\_user\_id` ASC) ,

INDEX `fk\_prisoner\_admin1` (`admin\_admin\_id` ASC) ,

CONSTRAINT `fk\_prisoner\_user1`

FOREIGN KEY (`user\_user\_id` )

REFERENCES `Crime\_DBMS`.`user` (`user\_id` )

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_prisoner\_admin1`

FOREIGN KEY (`admin\_admin\_id` )

REFERENCES `Crime\_DBMS`.`admin` (`admin\_id` )

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

"Prisoner" Table

-- -----------------------------------------------------

-- Table `Crime\_DBMS`.`crime\_type`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `Crime\_DBMS`.`crime\_type` (

`crime\_type\_id` INT NOT NULL ,

`crime\_name` VARCHAR(45) NOT NULL ,

`prisoner\_prisoner\_id` INT NOT NULL ,

`most\_wanted\_mstwntd\_id` INT NOT NULL ,

PRIMARY KEY (`crime\_type\_id`, `prisoner\_prisoner\_id`, `most\_wanted\_mstwntd\_id`) ,

INDEX `fk\_crime\_type\_prisoner1` (`prisoner\_prisoner\_id` ASC) ,

INDEX `fk\_crime\_type\_most\_wanted1` (`most\_wanted\_mstwntd\_id` ASC) ,

CONSTRAINT `fk\_crime\_type\_prisoner1`

FOREIGN KEY (`prisoner\_prisoner\_id` )

REFERENCES `Crime\_DBMS`.`prisoner` (`prisoner\_id` )

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_crime\_type\_most\_wanted1`

FOREIGN KEY (`most\_wanted\_mstwntd\_id` )

REFERENCES `Crime\_DBMS`.`most\_wanted` (`mstwntd\_id` )

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

"Crime\_Type" Table

Screen Shot 1 of

Data Inserted

Screen Shot 2 of

Data Inserted

JAVA Script

Fir.js

Screen Shot of Fir.js

Complaint.js

Screen Shot of Complaint.js

Missing.js

Screen Shot of Missing.js

Complaintaction.js

<%

try {

String description = request.getParameter("description"); //requests value for variable

String address = request.getParameter("address"); //requests value for variable

String date = request.getParameter("date"); //requests value for variable

String user\_user\_id = request.getParameter("user\_user\_id"); //requests value for variable

Class.forName("com.mysql.jdbc.Driver").newInstance(); // load jdbs driver "com.mysql.jdbc.Driver"

Connection con = DriverManager.getConnection /\*Create a connection by using getConnection() method that takes parameters

("jdbc:mysql://localhost:3306/crime\_dbms", "root", "akg6112099"); of string type connection url,user name a password to connect to database\*/

Statement st = con.createStatement(); //creating statement

st.executeUpdate ("INSERT INTO complaint\_report /\*statement to insert data into database, values for (description, address,

(description, address, date, user\_user\_id) VALUES('"+description+"', date,user\_id)\*/

'"+address+"', "+date+", "+user\_user\_id+")");

out.println("Complaint submitted,take action soon,Thank You!!!"); //print

} catch (Exception e) {

out.println(e); //screen message executed

}

%>

Screen Shot of

Inserting Data

Screen Shot after

Data Inserted

JSP

Viewmostwanted.jsp

<

%try {Class.forName("com.mysql.jdbc.Driver").newInstance();

Connection con1 = DriverManager.getConnection("Jdbc:mysql://localhost/crime\_dbms", "root", "akg6112099");

Statement st = con1.createStatement();

ResultSet rs = st.executeQuery("select \* from most\_wanted");

String frst\_name = "", surname = "", age = "", nationality = "";%>

<%while (rs.next()) {

frst\_name = rs.getString("frst\_name");

surname = rs.getString("surname");

age = rs.getString("age");

nationality = rs.getString("nationality");

out.println("");

out.println("" + frst\_name + "");

out.println("" + surname + "");

out.println("" + age + "");

out.println("" + nationality + "");

out.println("");}

%>

Screen Shot of Viewmostwanted.jsp

Viewmissingperson.jsp

<%

try {Class.forName("com.mysql.jdbc.Driver").newInstance();

Connection con1 = DriverManager.getConnection("Jdbc:mysql://localhost/crime\_dbms", "root", "akg6112099");

Statement st = con1.createStatement();

ResultSet rs = st.executeQuery("select \* from missing\_report");

String mis\_per\_fname = "", mis\_per\_surname = "", sex = "", age = "", height = "",

mis\_per\_address = "", last\_seen\_place = "", last\_seen\_date = "";%>

<%while (rs.next()) {

mis\_per\_fname = rs.getString("mis\_per\_fname");

mis\_per\_surname = rs.getString("mis\_per\_surname");

sex = rs.getString("sex");

age = rs.getString("age");

height = rs.getString("height");

mis\_per\_address = rs.getString("mis\_per\_address");

last\_seen\_place = rs.getString("last\_seen\_place");

last\_seen\_date = rs.getString("last\_seen\_date");

out.println("");

out.println("" + mis\_per\_fname + "");

out.println("" + mis\_per\_surname + "");

out.println("" + sex + "");

out.println("" + age + "");

out.println("" + height + "");

out.println("" + mis\_per\_address + ""