1. INTRODUCTION

1.1 PROJECT INTRODUCTION

Lockdown Pass System is a web-based technology that will manage the records of pass which is issue by administrative and also help to provide online Lockdown Pass to people who need to travel compulsory. Lockdown Pass System is an automatic system which delivers data processing in very high speed in systematic manner.

The E-pass system provides choice to the Admin to pick out a class (Which is necessities and as directed by the state government) to assign a specific a selected style of e-pass turning into there to particular class. Admin has rights to add/remove a specific class. Admin portal has associate choice to add details of Admin himself and additionally provides all the access management to the admin. The admin will generate and issue a pass to a specific person. this is often associate access based mostly internet application developed mistreatment Php, MySQL, and Java Script, when the victorious generation of the e-pass the pass holders will verify their travel simply coming into the pass variety within the given input field. The pass will be verified globally from anyplace with the assistance of the pass Id. This helps the user and also the authorities to validate the pass with convenience.

PASS GENERATION AND MANAGEMENT SYSTEM throughout Lockdown could be a web-based technology that may manage the records of pass that is issue by body and additionally facilitate to supply on-line curfew e-pass to folks that have to be compelled to travel mandatory. Curfew Pass Management System is associate automatic system that delivers processing in terribly high speed in systematic manner. In Lockdown e-Pass we have a tendency to use PHP and MySQL info. this is often the project that keeps records of the pass that is issue by body.

2. LITERATURE SURVEY

2.1 EXISTING SYSTEM

This project has been developed to manage the entire working of the Lockdown Pass System administrative. This software simplifies and replaces all the manual effort and the paper works done by the administrative to a completely electronically environment. Hence both the user and the administrative are at their ease.

2.2 PROPOSED SYSTEM

The Lockdown Pass System has many powerful features and is certainly more than a "simple" diagramming tool. With its support of MDA (Model Driven Architecture), it is more aimed at people using UML in an intensive way and with some code generations objectives than for simply drawing diagrams to document requirements. Modern world is computer world where the things have to be done promptly that requires optimal resources and optimal methods. Due to this inevitable requirement, computerization of each and every sector in the main stream is must, so that it can be held itself in the race. Few eye-catching features of this project are its simplicity, accuracy, and its user-friendly interface.

2.3 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are:

2.3.1 Economic Feasibility

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

2.3.2 Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes for the implementing this system.

2.3.3 Operational Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

2.4 SOFTWARE SPECIFICATION

HTML

HTML or **Hypertext Markup Language** is the standard markup language used to create web pages.

HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets (like `<html>`). HTML tags most commonly come in pairs like `<h1>` and `</h1>`, although some tags represent empty elements and so are unpaired, for example `<img>`. The first tag in a pair is the *start tag*, and the second tag is the *end tag* (they are also called opening tags and closing tags). Though not always necessary, it is best practice to append a slash to tags which are not paired with a closing tag.

The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language rather than a programming language.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means

to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as JavaScript which affect the behavior of HTML web pages.

CASCADING STYLE SHEETS (CSS)

It is a style sheet language used for describing the look and formatting of a document written in a markup language. While most often used to style web pages and interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. CSS is a cornerstone specification of the web and almost all web pages use CSS style sheets to describe their presentation.

CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colors, and fonts.[1] This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content .

CSS can also allow the same markup page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or screen reader) and on Braille-based, tactile devices. It can also be used to allow the web page to display differently depending on the screen size or device on which it is being viewed. While the author of a document typically links that document to a CSS file, readers can use a different style sheet, perhaps one on their own computer, to override the one the author has specified. However, if the author or the reader did not link the document to a specific style sheet the default style of the browser will be applied.

MySQL

MySQL is developed, distributed, and supported by Oracle Corporation. MySQL is a database system used on the web it runs on a server. MySQL is ideal for both small and large applications. It is very fast, reliable, and easy to use. It supports standard SQL. MySQL can be compiled on a number of platforms.

The data in MySQL is stored in tables. A table is a collection of related data, and it consists.

JAVASCRIPT

JavaScript is the scripting language of the Web. All modern HTML pages are using JavaScript. A scripting language is a lightweight programming language. JavaScript code can be inserted into any HTML page, and it can be executed by all types of web browsers. JavaScript is easy to learn.

PHP

WHAT IS PHP?

- PHP is an acronym for "PHP Hypertext Preprocessor"
- PHP is a widely-used, open-source scripting language
- PHP scripts are executed on the server
- PHP costs nothing, it is free to download and use

WHAT IS PHP FILE?

- PHP files can contain text, HTML, CSS, JavaScript, and PHP code
- PHP code are executed on the server, and the result is returned to the browser as plain HTML
- PHP files have extension ".php"

WHAT CAN PHP DO?

- PHP can generate dynamic page content
- PHP can create, open, read, write, delete, and close files on the server
- PHP can collect form data
- PHP can send and receive cookies
- PHP can add, delete, modify data in your database
- PHP can restrict users to access some pages on your website
- PHP can encrypt data
- With PHP you are not limited to output HTML. You can output images, PDF files, and even Flash movies. You can also output any text, such as XHTML and XML.

WHY PHP?

- PHP runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
- PHP is compatible with almost all servers used today (Apache, IIS, etc.)
- PHP supports a wide range of databases
- PHP is free. Download it from the official PHP resource: www.php.net.

3. SOFTWARE REQUIREMENTS SPESIFICATION

3.1 INTRODUCTION

To be used efficiently, all computer software needs certain hardware components or the other software resources to be present on a computer. These pre-requisites are known as(computer) system requirements and are often used as a guideline as opposed to an absolute rule. Most software defines two sets of system requirements: minimum and recommended. With increasing demand for higher processing power and resources in newer versions of software, system requirements tend to increase over time. Industry analysts suggest that this trend plays a bigger part in driving upgrades to existing computer systems than technological advancements.

3.2 HARDWARE REQUIREMENTS

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatibility and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects of hardware requirements.

HARDWARE REQUIREMENTS FOR PRESENT PROJECT

PROCESSOR : Intel dual Core, i3

RAM : 1 GB

HARD DISK : 80 GB

3.3 SOFTWARE REQUIREMENTS

Software Requirements deal with defining software resource requirements and pre-requisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or pre-requisites are generally not included in the software installation package and need to be installed separately before the software is installed.

SOFTWARE REQUIREMENTS FOR PRESENT PROJECT

OPERATING SYSTEM	:	Windows 7/ XP/8
FRONT END	:	HTML, CSS, JavaScript query
SERVER-SIDE SCRIPT	:	PHP
DATABASE	:	MySQL

3.4 MODULES

Lockdown e-Pass has one module i.e. admin.

3.4.1 Admin module:

- Dashboard: during these sections, admin will concisely read the entire range of class and the way several passes are generated in at some point, yesterdays and last seven's days.
- Category: during this section, admin will manage class (add/update).
- Add Pass: during this section, admin add pass.
- Manage Pass: during this section, admin will update pass and take print of that pass.
- Reports: during this section admin will generate pass reports between 2 dates.
- Search: during this section, admin will search a selected travel by pass range. Admin may also update his profile, amendment the arcanum and recover the arcanum.
- Brief info concerning homepage User will visit home page and search pass details by coming into pass range. Admin can also change his/her own password.

4. SYSTEM DESIGN

4.1 INTRODUCTION TO UML

The unified modeling language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmatic rules. A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagrams.

4.1.1 UML Design

The Unified Modeling Language (UML) is a standard language for specifying, visualizing, constructing, and documenting the software system and its components. It is a graphical language, which provides a vocabulary and set of semantics and rules. The UML focuses on the conceptual and physical representation of the system. It captures the decisions and understandings about systems that must be constructed. It is used to understand, design, configure, maintain, and control information about the systems.

The UML is a language for

- Visualizing
- Specifying
- Constructing
- Documenting

Visualizing

Through UML we see or visualize an existing system and ultimately, we visualize how the system is going to be after implementation. Unless we think, we cannot implement. UML helps to visualize, how the components of the system communicate and interact with each other.

Specifying

Specifying means building, models that are precise, unambiguous and complete UML addresses the specification of all the important analysis design, implementation decisions that must be made in developing and deploying a software system.

Constructing

UML models can be directly connected to a variety of programming language through mapping a model from UML to a programming language like JAVA or C++ or VB. Forward Engineering and Reverse Engineering is possible through UML.

Documenting

The Deliverables of a project apart from coding are some Artifacts, which are critical in controlling, measuring and communicating about a system during its developing requirements, architecture, desire, source code, project plans, tests, prototypes releasers, etc.

4.1.2 UML Approach

UML Diagram

A diagram is the graphical presentation of a set of elements, most often rendered as a connected graph of vertices and arcs. you draw diagram to visualize a system from different perspective, so a diagram is a projection into a system. For all but most trivial systems, a diagram represents an elided view of the elements that make up a system. The same element may appear in all diagrams, only a few diagrams, or in no diagrams at all. In theory, a diagram may contain any combination of things and relationships. In practice, however, a small number of common combinations arise, which are consistent with the five most useful views that comprise the architecture of a software-intensive system. For this reason, the UML includes nine such diagrams

- Class diagram
- Object diagram
- Use case diagram
- Sequence diagram
- Collaboration diagram
- State chart diagram

- Activity diagram
- Component diagram
- Deployment diagram

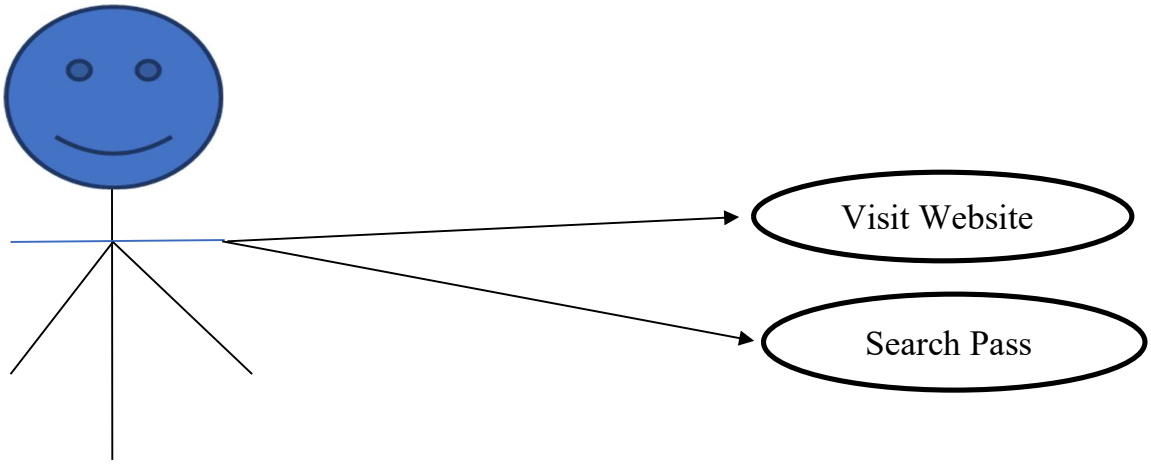
USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is type of behavioral diagram defined by and created from a use-case analysis. its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

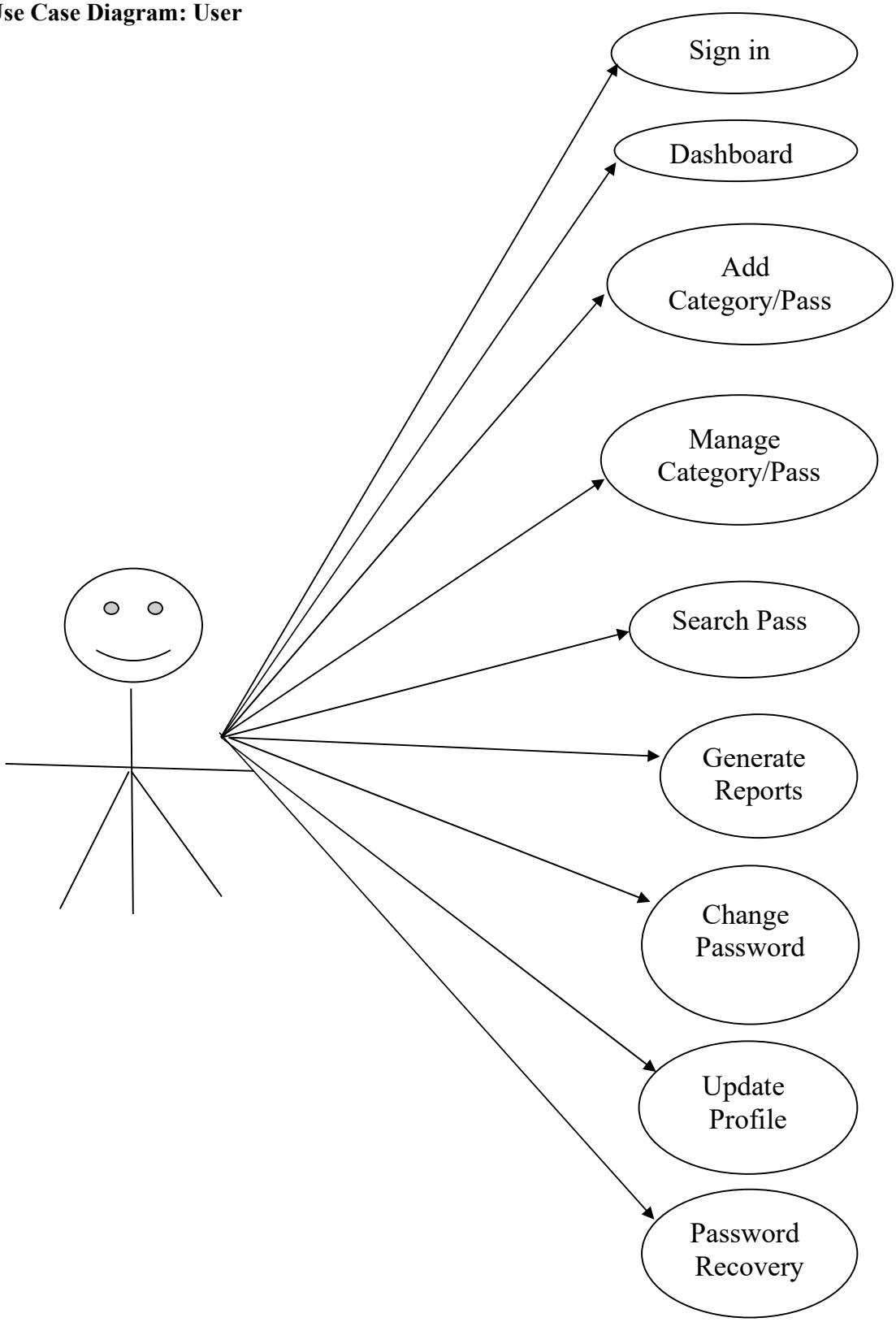
Use case diagrams are formally included in two modeling languages defined by the OMG: the unified modeling language (UML) and the systems modeling language(sysML)

Use case diagram of our project

Use Case Diagram: User

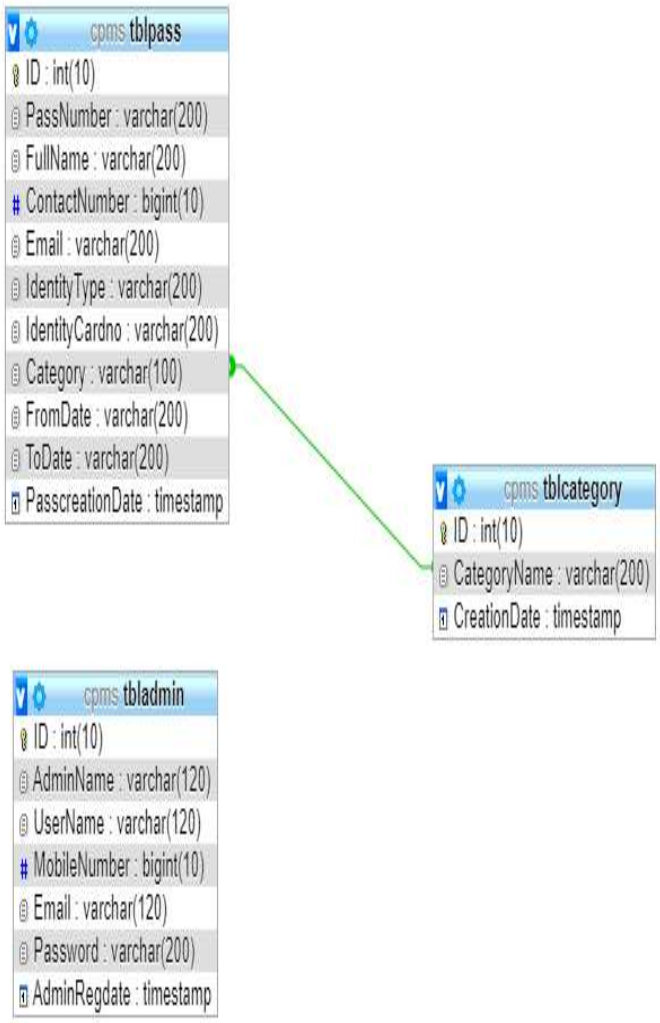


Use Case Diagram: User



4.2 Class Diagram

A Class is a category or group of things that has similar attributes and common behavior. A Rectangle is the icon that represents the class it is divided into three areas. The upper most area contains the name, the middle; area contains the attributes and the lowest areas show the operations. Class diagrams provides the representation that developers work from. Class diagrams help on the analysis side, too.

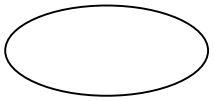
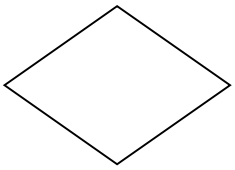


4.3 ER Diagram

ENTITY-RELATIONSHIP Diagrams

E-R (Entity-Relationship) Diagram is used to represents the relationship between entities in the table.

The symbols used in E-R diagrams are:

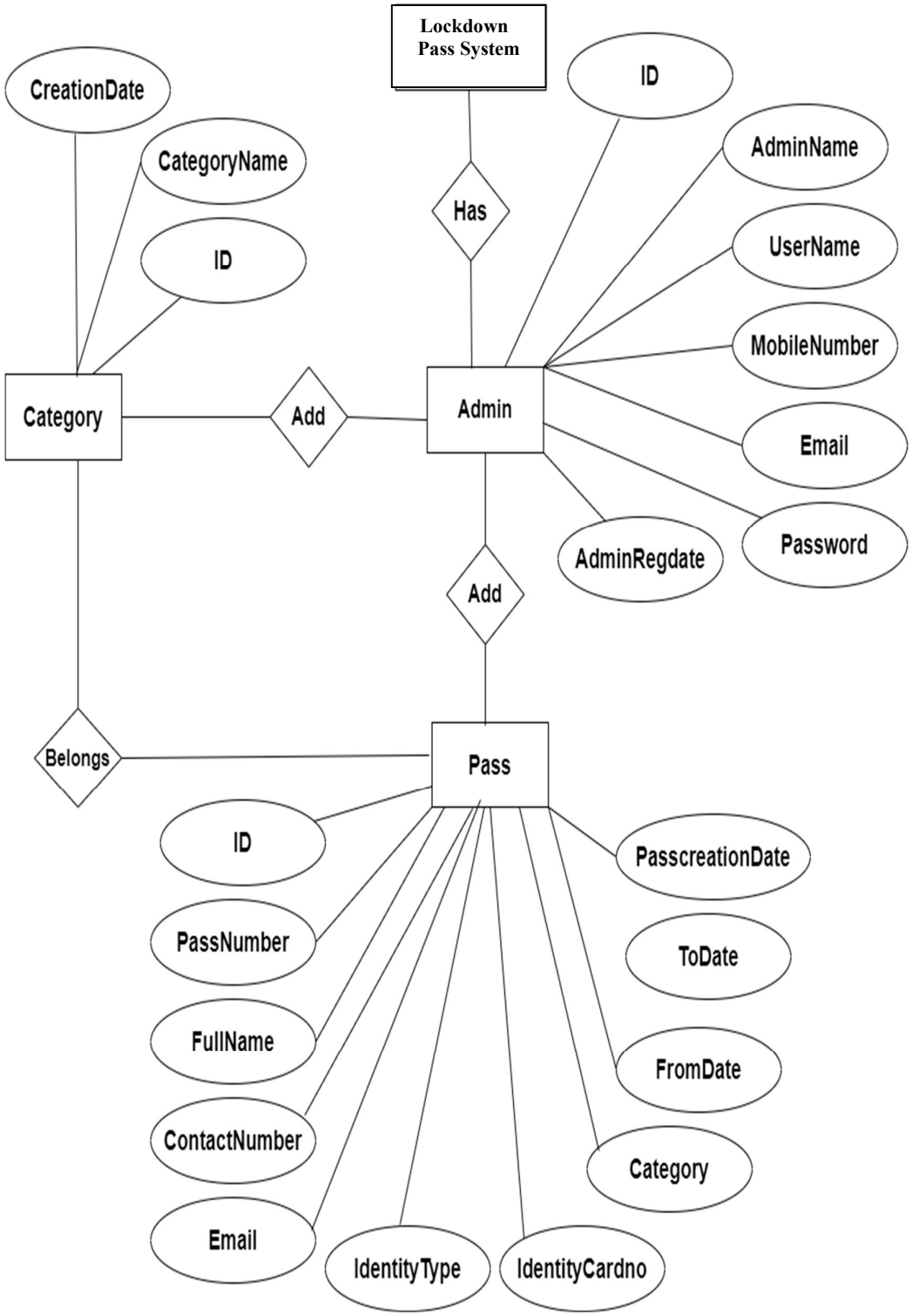
SYMBOL	PURPOSE
	Represents Entity sets.
	Represent attributes.
	Represent Relationship Sets.
	Line represents flow

Structured analysis is a set of tools and techniques that the analyst.

To develop a new kind of a system:

The traditional approach focuses on the cost benefit and feasibility analysis, Project management, and hardware and software selection a personal consideration.

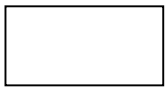
ER Diagram


Lockdown Pass System
ID
AdminName
UserName
MobileNumber
Email
Password
AdminRegdate
Has
Admin
Add
Category
CreationDate
CategoryName
ID
Add
Belongs
Pass
ID
PassNumber
FullName
ContactNumber
Email
IdentityType
IdentityCardno
PasscreationDate
ToDate
FromDate
Category

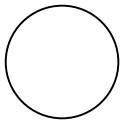
4.5 DFD DIAGRAM

A DFD does not show a sequence of steps. A DFD only shows what the different process in a system is and what data flows between them.

The following are some DFD symbols used in the project

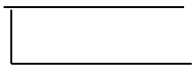


External entities



Process: A transaction of information that resides within the bounds of the system to be module.

Dataflows



DATASTORE: A repository of data that is to be stored for use by one or more processes, may be as simple as buffer of queue or as a relational database.

RULES FOR DFD:

- Fix the scope of the system by means of context diagrams.
- Organize the DFD so that the main sequence of the actions reads left to right and top to bottom.
- Identify all inputs and outputs.
- Identify and label each process internal to the system with rounded circles.
- A process is required for all the data transformation and transfers. Therefore, never connect a data store to a data source or the destinations or another data store with just a data flow arrow.
- Do not indicate hardware and ignore control information.

- Make sure the names of the processes accurately convey everything the process is done.
- There must not be unnamed process.
- Indicate external sources and destinations of the data, with squares.
- Number each occurrence of repeated external entities.
- Identify all data flows for each process step, except simple Record retrievals.
- Label data flow on each arrow.
- Use details flow on each arrow.
- Use the details flow arrow to indicate data movements.
- There can't be unnamed data flow.
- A data flow can't connect two external entities.

LEVELS OF DFD:

The complexity of the business system means that it is a responsible to represent the operations of any system of single data flow diagram. At the top level, an Overview of the different systems in an organization is shown by the way of context analysis diagram. When exploded into DFD

They are represented by:

- LEVEL-0: SYSTEM INPUT/OUTPUT
- LEVEL-1: SUBSYSTEM LEVEL DATAFLOW FUNCTIONAL
- LEVEL-2: FILE LEVEL DETAIL DATA FLOW.

The input and output data shown should be consistent from one level to the next.

LEVEL-0: SYSTEM INPUT/OUTPUT LEVEL

A level-0 DFD describes the system-wide boundaries, dealing inputs to and outputs from the system and major processes. This diagram is similar to the combined user-level context diagram.

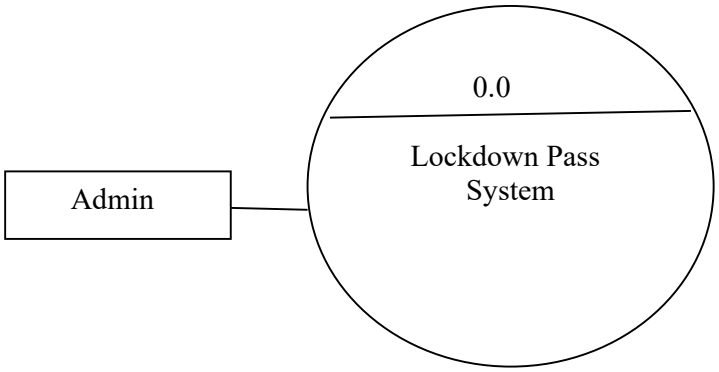
LEVEL-1: SUBSYSTEM LEVEL DATA FLOW

A level-1 DFD describes the next level of details within the system, detailing the data flows between subsystems, which makeup the whole.

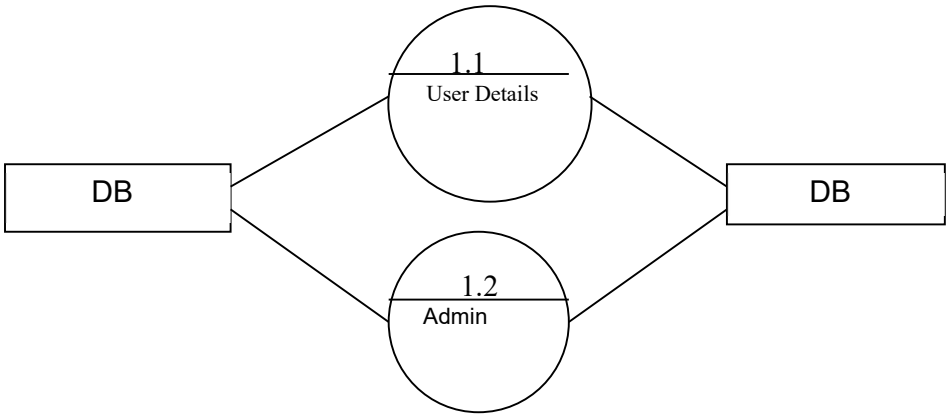
LEVEL-2: FILE LEVEL DETAIL DATA FLOW

All the projects are feasible given unlimited resources and infinite time. It is both necessary and prudent to evaluate the feasibility of the project at the earliest possible time. Feasibility and the risk analysis are pertained in many ways. If project risk is great.

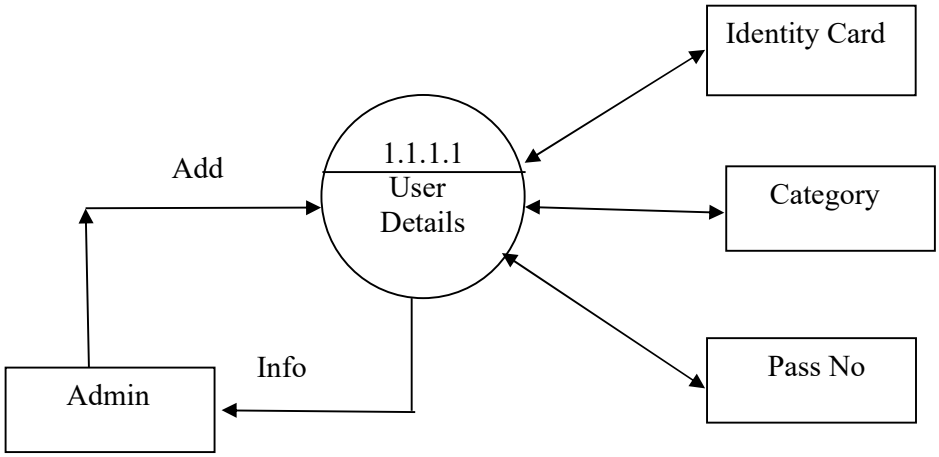
LEVEL 0 DFD



Level 1 DFD



Level 2 DFD


Identity Card
1.1.1.1
User
Details
Add
Category
Pass No
Info
Admin

5. DATABASE DESIGN

The data in the system has to be stored and retrieved from database. Designing the database is part of system design. Data elements and data structures to be stored have been identified at analysis stage. They are structured and put together to design the data storage and retrieval system.

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 database access easy, quick, inexpensive and flexible for the user. Relationships are established between the data items and unnecessary data items are removed. Normalization is done to get an internal consistency of data and to have minimum redundancy and maximum stability. This ensures minimizing data storage required, minimizing chances of data inconsistencies and optimizing for updates. The MS Access database has been chosen for developing the relevant databases.

Lockdown e-Pass System Project contains 3 MySQL tables

tbladmin table Structure : This table store the login and personal details of admin.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	ID	int(10)			No	*None*		AUTO_INCREMENT
2	AdminName	varchar(120)	utf8mb4_general_ci		Yes	*NULL*		
3	UserName	varchar(120)	utf8mb4_general_ci		Yes	*NULL*		
4	MobileNumber	bigint(10)			Yes	*NULL*		
5	Email	varchar(120)	utf8mb4_general_ci		Yes	*NULL*		
6	Password	varchar(200)	utf8mb4_general_ci		Yes	*NULL*		
7	AdminRegdate	timestamp			Yes	current_timestamp()		

tblcategory table Structure : This table store the category of pass.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	ID	int(10)			No	*None*		AUTO_INCREMENT
2	CategoryName	varchar(200)	utf8mb4_general_ci		Yes	*NULL*		
3	CreationDate	timestamp			Yes	current_timestamp()		

tblcategory table Structure : This table store the pass details of person.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	ID	int(10)			No	*None*		AUTO_INCREMENT
2	PassNumber	varchar(200)	utf8mb4_general_ci		Yes	*NULL*		
3	FullName	varchar(200)	utf8mb4_general_ci		Yes	*NULL*		
4	ContactNumber	bigint(10)			Yes	*NULL*		
5	Email	varchar(200)	utf8mb4_general_ci		Yes	*NULL*		
6	IdentityType	varchar(200)	utf8mb4_general_ci		Yes	*NULL*		
7	IdentityCardno	varchar(200)	utf8mb4_general_ci		Yes	*NULL*		
8	Category	varchar(100)	utf8mb4_general_ci		Yes	*NULL*		
9	FromDate	varchar(200)	utf8mb4_general_ci		Yes	*NULL*		
10	ToDate	varchar(200)	utf8mb4_general_ci		Yes	*NULL*		
11	PasscreationDate	timestamp			Yes	current_timestamp()		

6. IMPLEMENTATION

6.1 INTRODUCTION

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus, it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

6.2 SAMPLE CODE

6.2.1 Home Page Index

```
<?php
session_start();
//error_reporting(0);
include('includes/dbconnection.php');
 ?>
<!DOCTYPE html>
<html lang="en">
<head>

 <title>Lockdown Pass Checking - Home</title>

 <link rel="stylesheet" href="vendors/bootstrap/bootstrap.min.css">
 <link rel="stylesheet" href="vendors/themify-icons/themify-icons.css">
 <link rel="stylesheet" href="vendors/owl-carousel/owl.theme.default.min.css">
 <link rel="stylesheet" href="vendors/owl-carousel/owl.carousel.min.css">

 <link rel="stylesheet" href="css/style.css">
</head>
```

```
<body>

 <!--================ Header Menu Area start ================-->
<?php include_once('includes/header.php');?>
 <!--================Header Menu Area ================-->

 <!--================ Banner Section start ================-->
 <section class="hero-banner text-center">
  <div class="container">
   <h1>Lockdown Pass Checking</h1>

  </div>
 </section>
 <!--================ Banner Section end ================-->

 <!--================ Domain Search section start ================-->
 <section class="bg-gray domain-search">
  <div class="container">
   <div class="row no-gutters">
    <div class="col-md-5 col-lg-2 text-center text-md-left mb-3 mb-md-0">
     <h3>CHECK HERE👉</h3>
    </div>
    <div class="col-md-7 col-lg-10 pl-2 pl-xl-5">
     <form class="form-inline flex-nowrap form-domainSearch" method="post">
      <div class="form-group">
       <label for="staticDomainSearch" class="sr-only">Search</label>
       <input id="searchdata" type="text" name="searchdata" required="true" class
="form-control" placeholder="Enter Your Pass ID">
      </div>
      <button type="submit" class="button rounded-
0" name="search" id="submit">Search👆</button>
     </form>
      <?php
```

```
if(isset($_POST['search']))
{
$sdata=$_POST['searchdata'];
  ?>
  <h4 align="center">Result against "<?php echo $sdata;?>" keyword </h4>

    <table class="table table-striped table-bordered table-hover" id="dataTables-
example">
 <?php
$sql="SELECT * from tblpass where PassNumber like '%$sdata%'";
$query = $dbh -> prepare($sql);
$query->execute();
$results=$query->fetchAll(PDO::FETCH_OBJ);

$cnt=1;
if($query->rowCount() > 0)
{
foreach($results as $row)
{               ?>
     <tr align="center">
<td colspan="6" style="font-size:20px;color:blue">
 Pass ID: <?php  echo ($row->PassNumber);?></td></tr>

  <tr>
    <th scope>Full Name</th>
    <td><?php  echo ($row->FullName);?></td>
    <th scope>Mobile Number</th>
    <td><?php  echo ($row->ContactNumber);?></td>
    <th scope>Email</th>
    <td><?php  echo ($row->Email);?></td>
  </tr>
<tr>
    <th scope>Identity Type</th>
```

```
    <td><?php  echo ($row->IdentityType);?></td>
    <th scope>Identity Card Number</th>
    <td><?php  echo ($row->IdentityCardno);?></td>
    <th scope>Category</th>
    <td><?php  echo ($row->Category);?></td>
  </tr>
<tr>
    <th scope>From Date</th>
    <td><?php  echo ($row->FromDate);?></td>
    <th scope>To Date</th>
    <td><?php  echo ($row->ToDate);?></td>
    <th scope>Pass Creation Date</th>
    <td><?php  echo ($row->PasscreationDate);?></td>
  </tr>

    <?php
$cnt=$cnt+1;
} } else { ?>
  <tr>
    <td colspan="8"> No record found against this search</td>

  </tr>

<?php } }?>
     </table>

        </div>
      </div>
    </div>
  </section>
  <!--================ Domain Search section end ================-->

  <!-- ================ start footer Area ================ -->
```

```
  <?php include_once('includes/footer.php');?>
  <!-- ================ End footer Area ================= -->
  <script src="vendors/jquery/jquery-3.2.1.min.js"></script>
  <script src="vendors/bootstrap/bootstrap.bundle.min.js"></script>
  <script src="vendors/owl-carousel/owl.carousel.min.js"></script>
  <script src="js/jquery.ajaxchimp.min.js"></script>
  <script src="js/mail-script.js"></script>
  <script src="js/main.js"></script>
</body>
</html>
```

6.2.2 Admin Index

```
<?php
session_start();
error_reporting(0);
include('includes/dbconnection.php');

if(isset($_POST['login']))
  {
    $username=$_POST['username'];
    $password=md5($_POST['password']);
    $sql ="SELECT ID FROM tbladmin WHERE UserName=:username and Password
=:password";
    $query=$dbh->prepare($sql);
    $query-> bindParam(':username', $username, PDO::PARAM_STR);
$query-> bindParam(':password', $password, PDO::PARAM_STR);
    $query-> execute();
    $results=$query->fetchAll(PDO::FETCH_OBJ);
    if($query->rowCount() > 0)
{
foreach ($results as $result) {
$_SESSION['cpmsaid']=$result->ID;
}
```

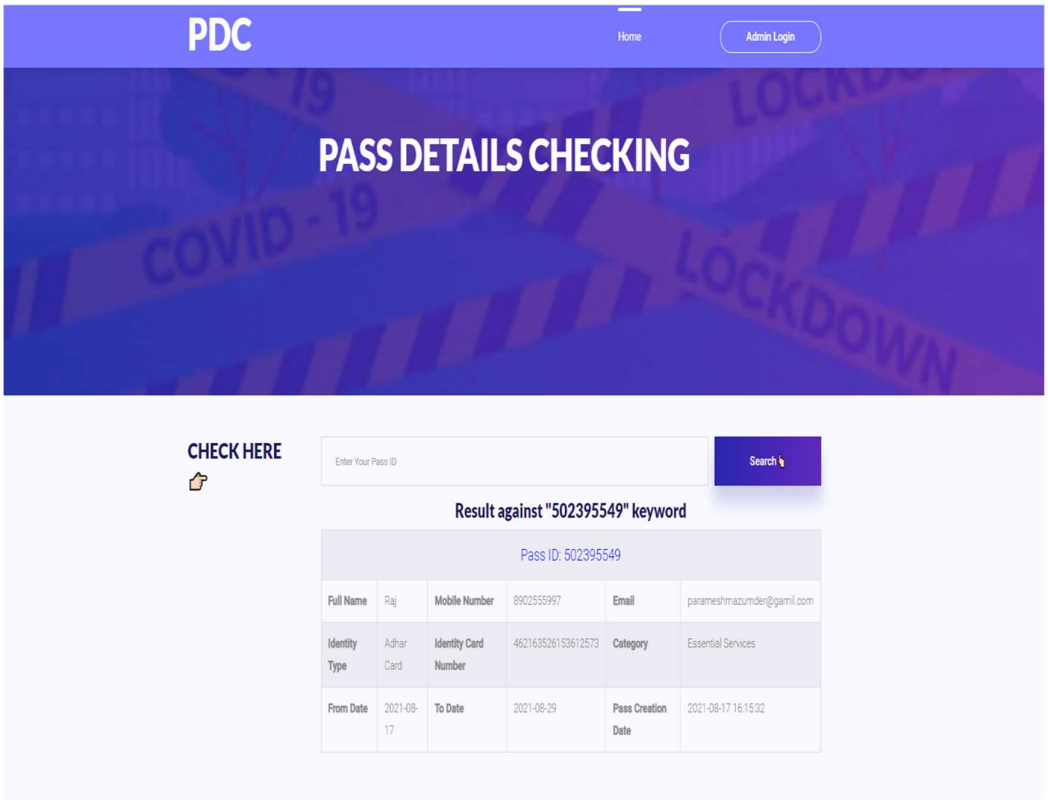
```
 if(!empty($_POST["remember"])) {
//COOKIES for username
setcookie ("user_login",$_POST["username"],time()+ (10 * 365 * 24 * 60 * 60));
//COOKIES for password
setcookie ("userpassword",$_POST["password"],time()+ (10 * 365 * 24 * 60 * 60));
} else {
if(isset($_COOKIE["user_login"])) {
setcookie ("user_login","");
if(isset($_COOKIE["userpassword"])) {
setcookie ("userpassword","");
      }
    }
}
$_SESSION['login']=$_POST['username'];
echo "<script type='text/javascript'> document.location ='dashboard.php'; </script>";
} else{
echo "<script>alert('Invalid Details');</script>";
}
}

?>
<!DOCTYPE html>
<html>

<head>

   <title>Lockdown e-Pass | Login Page</title>
   <!-- Core CSS - Include with every page -->
   <link href="assets/plugins/bootstrap/bootstrap.css" rel="stylesheet" />
   <link href="assets/font-awesome/css/font-awesome.css" rel="stylesheet" />
   <link href="assets/plugins/pace/pace-theme-big-counter.css" rel="stylesheet" />
  <link href="assets/css/style.css" rel="stylesheet" />
```

```
    <link href="assets/css/main-style.css" rel="stylesheet" />

</head>

<body class="body-Login-back">

  <div class="container">

    <div class="row">
      <div class="col-md-4 col-md-offset-4 text-center logo-margin ">
       <h3 style="color: white;">Curfew e-Pass Management System</h3>
        </div>
      <div class="col-md-4 col-md-offset-4">
        <div class="login-panel panel panel-default">
          <div class="panel-heading">
            <h3 class="panel-title">Please Sign In</h3>
          </div>
          <div class="panel-body">
            <form role="form" method="post" name="login">
              <fieldset>
                <div class="form-group">
                  <label for="login-username">Username</label>
                   <input type="text" class="form-
control"  required="true" name="username" value="<?php if(isset($_COOKIE["user_
login"])) { echo $_COOKIE["user_login"]; } ?>">

                </div>
                <div class="form-group">
                  <label for="login-password">Password</label>
                  <input type="password" class="form-
control" name="password" required="true" value="<?php if(isset($_COOKIE["userp
assword"])) { echo $_COOKIE["userpassword"]; } ?>">
```

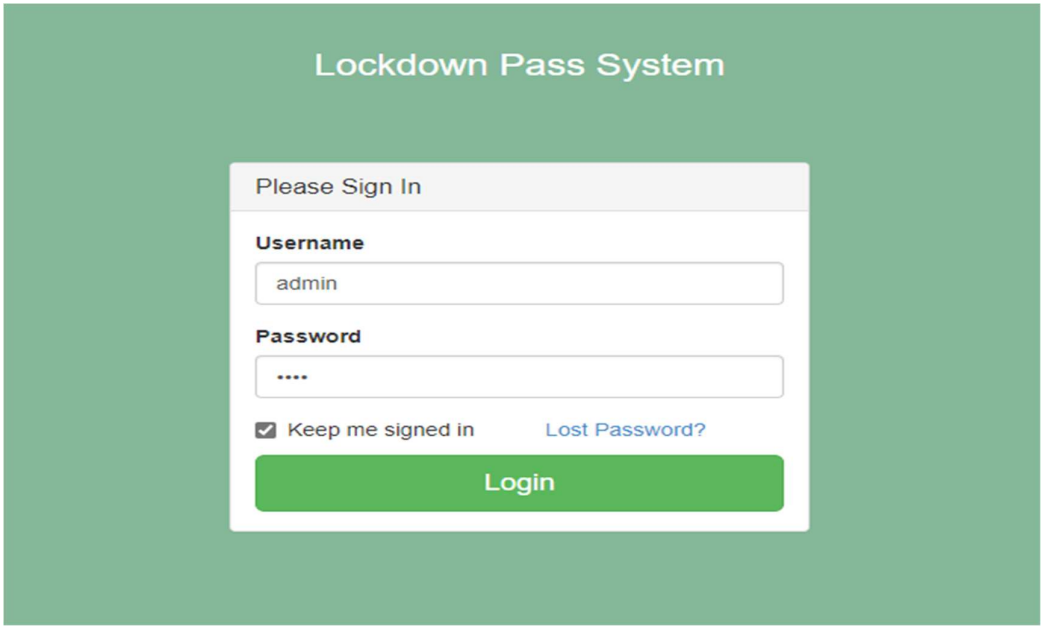
```
                        </div>
                        <div class="checkbox">

                            <input type="checkbox" id="remember" name="remember"
<?php if(isset($_COOKIE["user_login"])) { ?> checked <?php } ?> />
            <label for="keep_me_logged_in">Keep me signed in</label>


<label style="padding-left: 40px">
   <a href="forgot-password.php">Lost Password?</a></label>
                        </div>

                        <!-- Change this to a button or input when using this as a form -->

                        <input type="submit" value="Login" class="btn btn-lg btn-
success btn-block" name="login" >
                    </fieldset>
                </form>
              </div>
            </div>
          </div>
        </div>
   </div>

    <!-- Core Scripts - Include with every page -->
   <script src="assets/plugins/jquery-1.10.2.js"></script>
   <script src="assets/plugins/bootstrap/bootstrap.min.js"></script>
   <script src="assets/plugins/metisMenu/jquery.metisMenu.js"></script>

</body>

</html>
```

6.3 SAMPLE SCREENSHOT

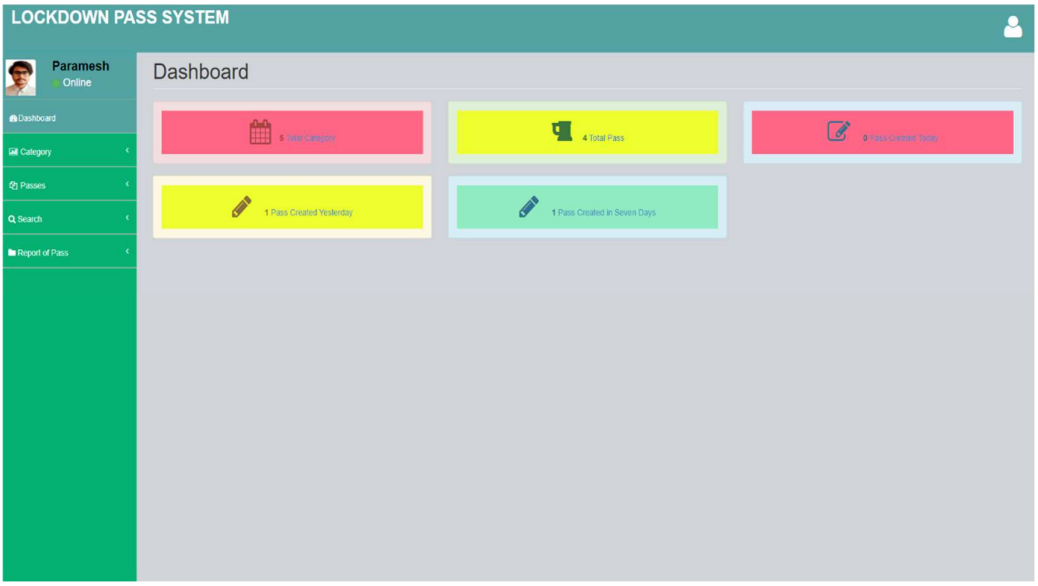
Home Page



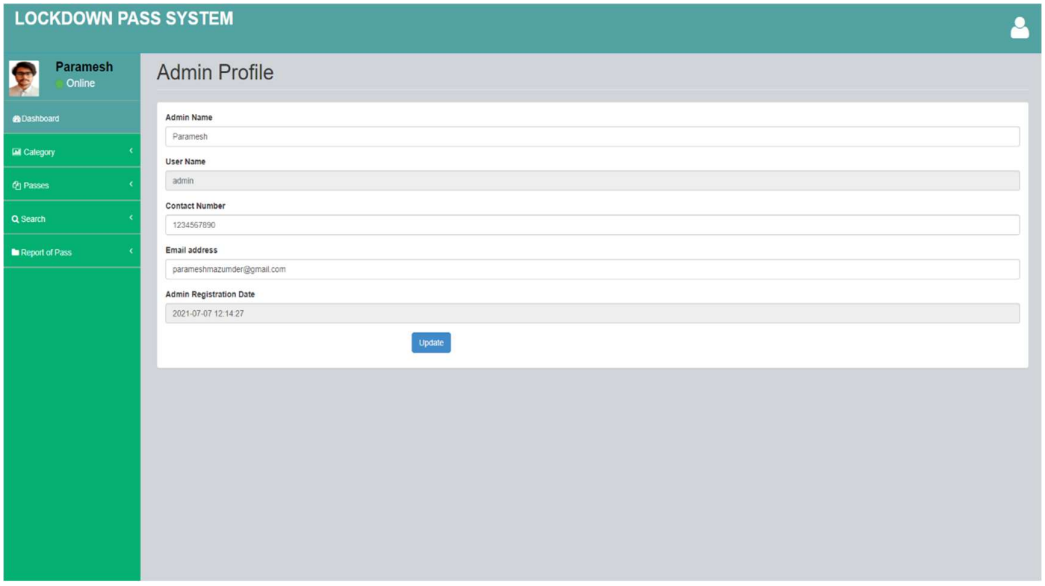
Admin Login



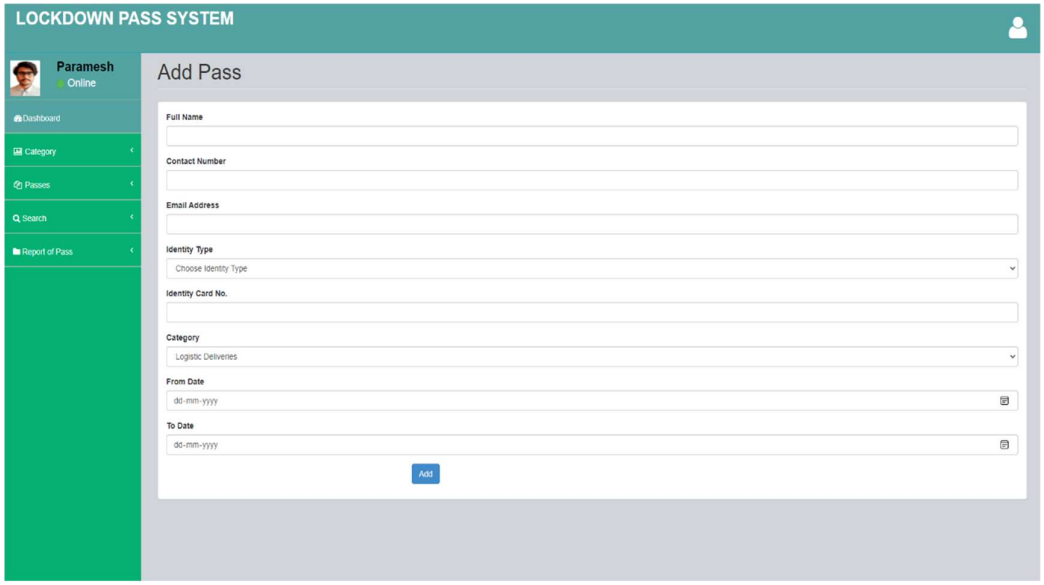
Dashboard



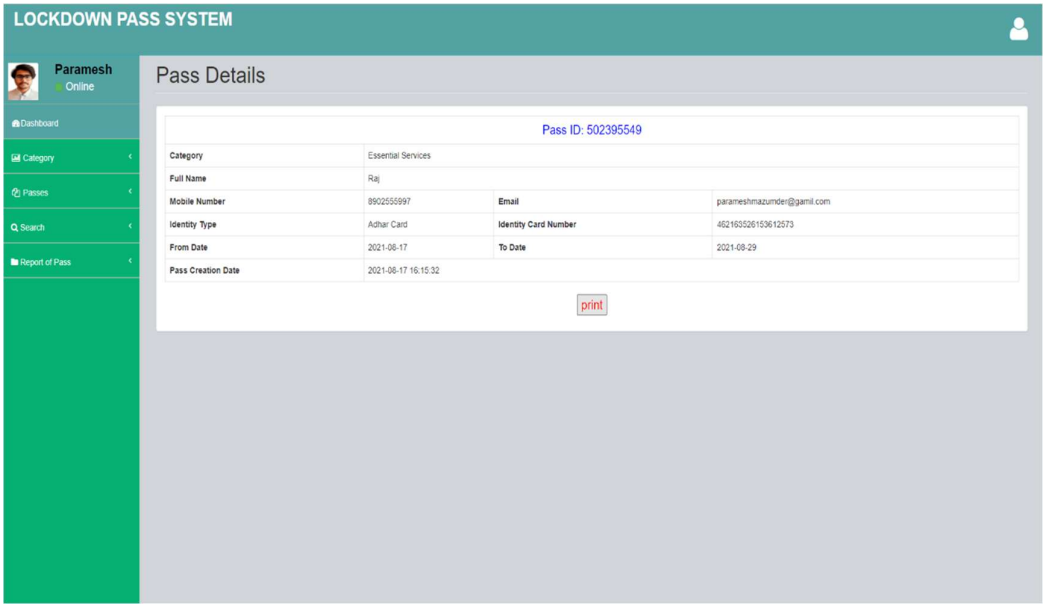
Admin Profile



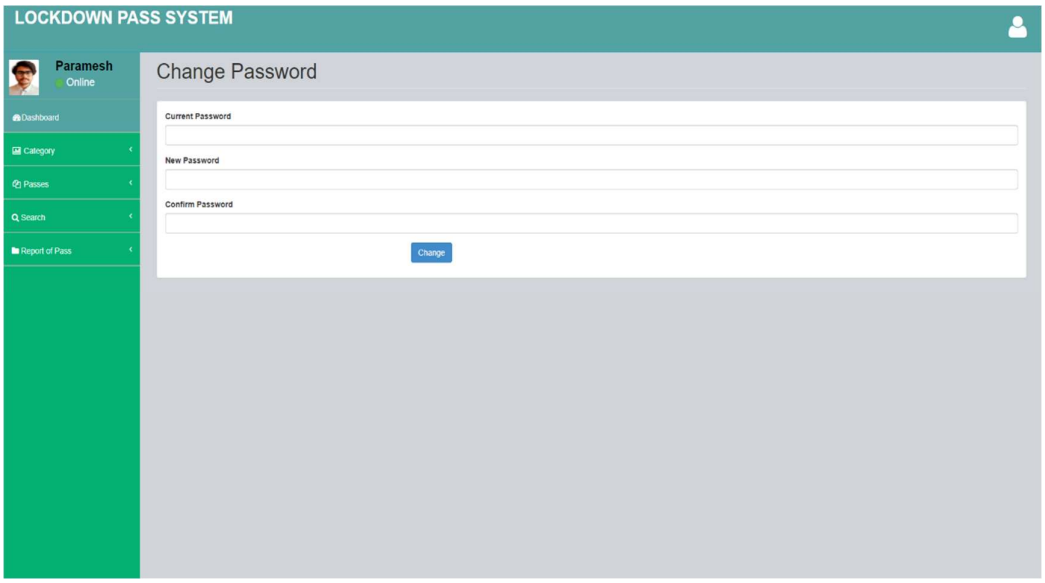
Add Pass



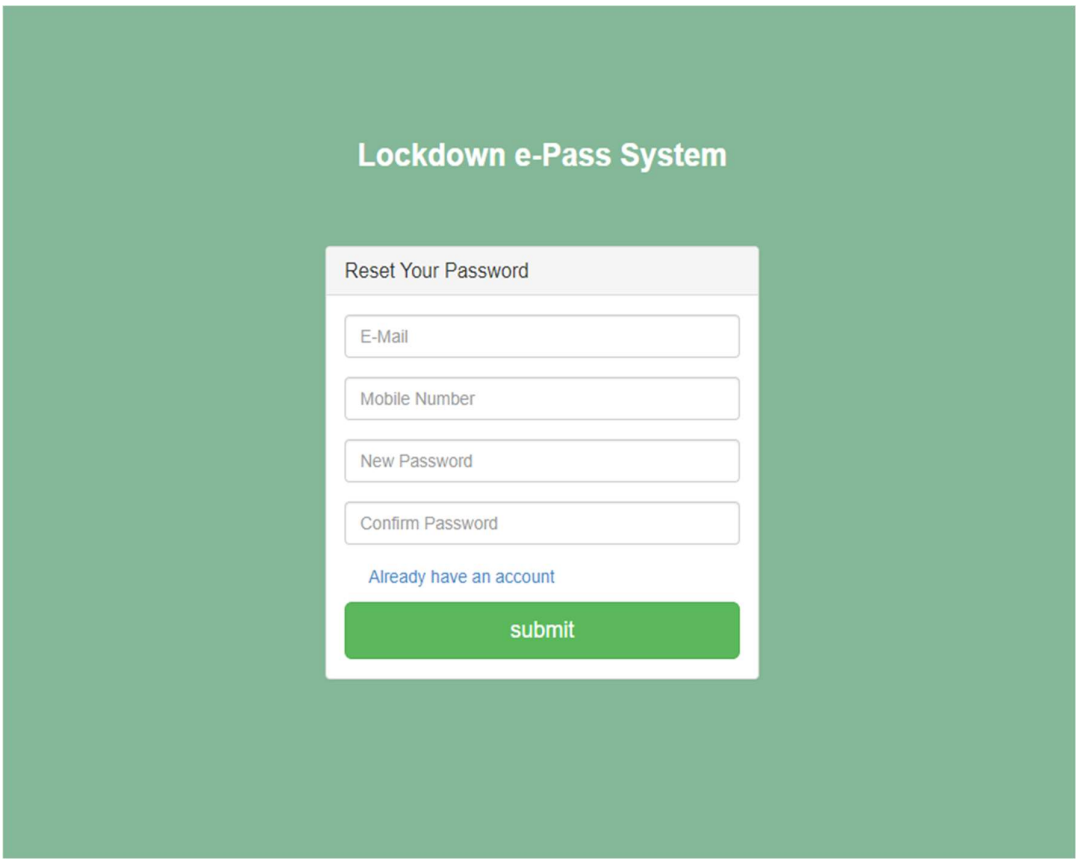
Print Pass



Change Password



Forget Password



7.TESTING AND REPORT

7.1 INTRODUCTION TO SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

7.2 TYPES OF TESTING

7.2.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

7.2.2 Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

7.2.3 Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input	: identified classes of valid input must be accepted.
Invalid Input	: identified classes of invalid input must be rejected.
Functions	: identified functions must be exercised.
Output	: identified classes of application outputs must be exercised.
Systems/Procedures	: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

7.2.4 System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

7.2.5 White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

7.2.6 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or

requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

7.2.7 Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g., components in a software system or – one step up – software applications at the company level – interact without error.

Test Results

All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results

All the test cases mentioned above passed successfully. No defects encountered.

Function	Description	%TCs Executed	%TCs Passed	TCs Pending	Priority
Admin Login	Check Admin Login	100%	100%	0	High
Add Category /pass	Check Add Category/pass	100%	100%	0	High
Manage category/ pass	Check Manage category/pass	100%	100%	0	High
Search Pass	Check Search Pass	100%	100%	0	High
Generate Report	Check Generate Report	100%	100%	0	High
Change Password	Verify Change Password	100%	100%	0	High
Password Recovery	Verify Password Recovery	100%	100%	0	High
Report Print	Check the Report Print	100%	100%	0	High

8. CONCLUSION

Lockdown e-Pass has many powerful features and is certainly more than a "simple" diagramming tool. With its support of MDA (Model Driven Architecture), it is more aimed at people using UML in an intensive way and with some code generations objectives than for simply drawing diagrams to document requirements. However, using Lockdown e-Pass just as a diagramming tool work fine, especially on Windows as the tool is built with Delphi and might execute faster than the Java-based tools. Modern world is computer world where the things have to be done promptly that requires optimal resources and optimal methods. Due to this inevitable requirement, computerization of each and every sector in the main stream is must, so that it can be held itself in the race. Few eye-catching features of our project are its simplicity, accuracy, and its user-friendly interface. Our software incorporates all the features and facilities provided by the Visual Studio software. This project has been developed to manage the entire working of the Lockdown Pass administrative. Our software simplifies and replaces all the manual effort and the paper works done by the administrative to a completely electronically environment. Hence both the user and the administrative are at their ease. The user is serviced at his footsteps wherein he just types in the request and he is just a click away. We would like to convey our sincere gratitude and thanks to all, who stood as our backbone, in designing, formatting and helping us in executing this project successfully.

9. FUTURE ENHANCHEMENTS

- Emergency booking slot.
- Volunteer pass zone.
- Address tracker of user.
- Special pass for differently abled person.
- Aadhar attachment system.
- Medical status update along with personal detail.
- Daily travel record update.

10. BIBLOGRAPHY

For PHP

- https://www.w3schools.com/php/default.asp
- https://www.sitepoint.com/php/
- https://www.php.net/

For MySQL

- https://www.mysql.com/
- http://www.mysqltutorial.org

For XAMPP

- https://www.apachefriends.org/download.html