**COURIER TRACKING SYSTEM**

**ABSTRACT**

The Project entitled “Courier Tracking System” is to develop a software product that is used in courier service to maintain their details systematically. The project on Intranet Module shall be a useful product for managing the courier services within a organization. The product is developed to maintain the sender and the receiver details, and also to get the information of the status of the courier.

It Uses java as Front End and SQL as Back End. The main benefit of this project is that it facilitate user to communicate in a faster manner in comparison of manual system. Through this system the current status of the courier can be known easily whereas in manual system it is a difficult task.

The Administrator department has all rights to provide or restrict to any department or employee to the functionality of the project. The working department will enter the information of the new courier into the data base. This Project facilitates to make calculation of the amount for sending a courier on the basis of the destination, weight & rate/gm. The working department employee can cancel the courier as per the request of the user with in the fixed time.

The customers can complaint about late delivery and damaged goods. Those complaints are registered and the information will be sent to the required department. The project is reliable only when the database is maintained properly.

**1. INTRODUCTION**

The main objective of this project is to managing the courier details itself. Customer can finding the up-to-date courier detail. Courier administrator login and fill the courier details, then the user having some unique id, it’s an courier id for that courier, then user just giving the courier number this system will help us to track courier.The project is aimed to develop by **JAVA** as Front end and **MS SQL SERVER** as Back end. The back end is used to store the information in this system.

**1.1 SYSTEM SPECIFICATION**

**1.1.1 HARDWARE SPECFICATION:**

* Processor : P 4 700 GHz.
* RAM : 4GB RAM
* Hard Disk Drive : 40 GB HDD

**1.1.2 SOFTWARE SPECIFICATION:**

* + Operating System : Windows XP/7/8/10
  + Front End : JAVA
  + Back End : MY SQL

1. **SYSTEM STUDY**

**2.1 EXISTING SYSTEM:**

In this system customer can’t track their courier details. If we know about the courier details we should visit the store and ask about the details. Otherwise we can’t track the details. Sometimes customer has feel about the courier

**2.1.1 DRAWBACKS:**

The existing system has the following drawbacks.

* we doesn’t know about the courier details.
* We don’t know about the current status of the courier
* It’s fully managed by the courier office team

**2.2 PROPOSED SYSTEM:**

This courier tracking system may help to track the courier by the customer. It can be very useful to tracking our courier detail. We don’t bother about the courier. The customer have one unique id with the help of that id we can track our courier details

**2.2.1 FEATURES:**

* We know about the courier details.
* No need visit the courier office.
* Fully manually find the courier details

1. **SYSTEM DESIGN AND DEVELOPMENT**

**3.1 FILE DESIGN**

The selection of the file system design approach is done according to the needs of the developers what are the needed requirements and specifications for the new design. It allowed us to identify where our proposal fitted in with relation to current and past file system development. Our experience with file system development is limited so the research served to identify the different techniques that can be used. The variety of file systems encountered show what an active area of research file system development is. The file systems may be from one of the two fundamental categories. In one category, the file system is developed in user space and runs as a user process. Another file system may be developed in the kernel space and runs as a privileged process. Another one is the mixed approach in which we can take the advantages of both aforesaid approaches. Each development option has its own pros and cons. In this article, these design approaches are discussed.

**3.2 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 in to 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?
* The dialog to guide the operating personnel in providing input.
* Methods for preparing input validations and steps to follow when error occur.

**OBJECTIVES**

* Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
* It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
* When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user
* will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

**3.3 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.

1. 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.

2. Select methods for presenting information.

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

The output form of an information system should accomplish one or more of the following objectives.

* Convey information about past activities, current status or projections of the
* Future.
* Signal important events, opportunities, problems, or warnings.
* Trigger an action.
* Confirm an action.

**3.4 DATABASE DESIGN**

Today's businesses depend on their databases to provide information essential for day-to-day operations, especially in case of electronic commerce businesses who has a definite advantage with up-to-date database access. Good design forms the foundation of any database, and experienced hands are required in the automation process to design for optimum and stable performance.

Software Solutions have been constantly working on these platforms and have attained a level of expertise. We apply proven methodologies to design, develop, integrate and implement database systems to attain its optimum level of performance and maximize security to meet the client's business model.

### Business needs addressed:

* Determine the basic objects about which the information is stored
* Determine the relationships between these groups of information and the objects
* Effectively manage data and create intelligent information
* Remote database administration or on site administrative support
* Database creation, management, and maintenance
* Information retrieval efficiency, remove data redundancy and ensure data security

**3.5 SYSTEM DEVELOPMENT**

**3.5.1 DESCRIPTION OF MODULES**

1. Register Customer
2. Booking order
3. Track order
4. Billing details

**Register Customer**

Administrator will register their customer. They can see all the details about the customer. After creating an account they will track their courier manually.

**Booking order**

This module is used to booking the order, after the order was booked the it will be tracking.

**Track Order**

Once the order was confirmed then customer have an one unique id, this will helps to track an order and order details.

**Billing Details**

This module will manage the billing details, administer will see the daily billing details.

1. **SYSTEM TESTING AND IMPLEMENTATION**

**SYSTEM TESTING**

System testing is the process of exercising software with the intent of finding and ultimately correcting errors. This fundamental philosophy does not change for web applications, because Web-based systems and application reside on a network and interoperate with many different operating system, browsers, hardware platforms, and communication protocols; the search for errors represents a significant challenge for web application.

The distributed nature of client\server environments, the performance issues associated with transaction processing, the potential presence of a number of different hardware platforms, the complexities of network communication, the need to serve multiple clients from a centralized database and the requirements imposed on the server all combine to make testing of client\server architectures.

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer based system. System testing is the state of implementation that is aimed at assuring that the system works accurately and efficiently. Testing is the vital to the success of the system. System testing makes the logical assumption that if all the parts of the system are correct, the goal will be successfully achieved.

**The objective of testing is as follows:**

* + Testing is the process of executing a program with the intent of finding an error.
  + A successful test is that one of the cover of undiscovered error.

### TESTING ISSUES

* Client GUI considerations
* Target environment and platform diversity considerations
* Distributed database considerations
* Distributed processing considerations

**TESTING METHODOLOGIES**

System testing is state of implementation, which is aimed at ensuring that the system works accurately and efficiently as expect before live operation commences. It certifies that the whole set of programs hang together.

System testing requires a test plan that consists of several key activities and step for run program, string, system and user acceptance testing. The implementation of newly designed package is important in adopting a successful new system

Testing is the important stage in software development. the system test in implementation stage in software development process. The system testing implementation should be confirmation that all is correct and an opportunity to show the users that the system works as expected. It accounts the largest percentage of technical effort in the software development process.

Testing phase in the development cycle validates the code against the functional specification testing is vital to achievement of the system goals. The objective of the testing is to discover errors to fulfills this objective a series of test step unit, integration. validation and system tests were planned and executed the test steps are:

**System Testing**

Testing is an important phase in project development. System testing makes a logical assumption that if all parts of the system are correct, and the goal will be achieved successfully. The software must meet the user specification and it must satisfy according to the needs of the users.

Testing is the process of executing a project within the intend of finding errors. A good test case is one that has a high probability of finding an undiscovered error.

**Unit Testing**

Unit testing focuses verification efforts on the smallest unit of software design of the module. This is also known as “module testing”. This testing is carried out during programming stage itself. In this testing step, each module is found to be working satisfactorily as regards to the expected output of the modules.

**In Project**, Each module such customer registration module, request module, employee details module, stock module, vehicle module and area detail modules are tested individually for example, Customer details module can contain the more forms to maintain the information so all forms could be tested like entered information store appropriately in database access page or not. If correctly accessed means the testing of registration module successfully completed. Likewise all modules are tested successfully.

**Integration Testing**

Data can be lost across an interface, one module can have adverse effect on another sub function when combined it may not produce the desired major functions. Integration testing is a systematic testing for constructing test to uncover errors associated within an interface.

The objectives taken from unit tested modules and a program structure is built for integrated testing. All the modules are combined and the test is made.

A correction made in this testing is difficult because the vast expenses of the entire program complicated the isolation of causes. In this integration testing step, all the errors are corrected for next testing process.

**In Project,** Integration of two modules can be tested together such as customer registration details and customer login module for verification purposes providing proper accessibility to users. The communication of Registration and Login module can test and executed successfully.

**Validation Testing**

After the completion of the integrated testing, software is completely assembled as a package; interfacing error has been uncovered and corrected and a final series of software test validation begins.

Validation testing can be defined in many ways but a simple definition is that validation succeeds when the software function in a manner that can be reasonably expected by the customer. After validation test has been conducted, one of two possible conditions exists:

**In this project,** Admin login details form Enter without username and password in textbox enter the submit button then Login failed message otherwise checks the both textbox value that is true means valid page displayed. Enter Password Displaying password character \*.if it displays the characters security is not availed so testing of software is failed.

**Output Testing**

The next process of validation testing, is output testing of the proposed system, since no system could be successful if it does not produce the required output in the specified format. Asking the user about the format required, list the output to be generated or displayed by the system under considerations.

Output testing is a different test whose primary purpose is to fully exercise the computer based system although each test has a different purpose all the work should verify that all system elements have been properly integrated and perform allocated functions.

The output format on the screen is found to be corrected as the format was designed in the system design phase according to the user needs for the hard copy also; the output testing has not resulted in any correction in the system.

**In project** All the forms are tested as it gives the necessary output to the user’s search such as view response details.

1. **CONCLUSION**

This project will help us to tracking the courier details, which ca be updated by courier managers for that particular courier and they will update the current location of the courier, after updating the courier will be changed current location

**FUTURE ENHANCEMENT**

In this project we have plan to move next way of tracking the courier, it’s like to tracking the courier using GPS. It’s giving an lattest location to the database, so every courier has tracking up-to-date. Admin no need to update the courier current place. it’s fully GPS based tracking the courier form. It’s is an user friendly so every one can use this application.

**BIBLIOGRAPHY**

* **CH97**  
  Gary Cornell and Cay S. Horstmann, Core Java, second ed., SunSoft Press, 1997
* **ELW98**  
  Robert Eckstein and Marc Loy and Dave Wood, Java Swing, O'Reilly, 1998.
* **Flanagan99**  
  David Flanagan, Java Foundation Classes in a Nutshell, O'Reilly, 1999.

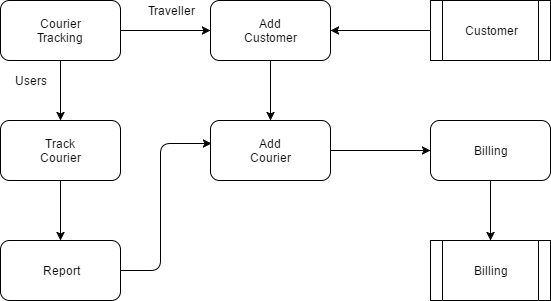
**APPENDICES**

1. **DATA FLOW DIAGRAM**

LEVEL 0:

****

LEVEL 1:

****

1. **TABLE STRUCTURE**

**TABLE NAME : ADMIN**

**PRIMARY\_KEY : ID**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| Id | Int |  | Admin id |
| Username | Varchar | 30 | Admin username |
| password | Varchar | 30 | Admin password |

**TABLE NAME : CUSTOMER**

**PRIMARY KEY : cid**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| Cid | Int | 10 | Customer id |
| Name | Varchar | 30 | Customer name |
| Address | Varchar | 30 | Address |
| Contactno | Varchar | 10 | Contact number |
| Gender | Varchar | 10 | Gender |
| Email | Varchar | 10 | email |

**TABLE NAME : COURIER**

**PRIMARY KEY : ID**

**FOREIGN KEY : UID**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| Id | Int | 10 | Courier id |
| Uid | Int | 10 | User id |
| Send location | Varchar | 30 | Send location |
| Current location | Varchar | 30 | Current location |
| Address | Varchar | 30 | address |

**TABLE NAME : BILLING**

**PRIMARY KEY : ID**

**FOREIGN KEY : CID,UID**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| Id | Int | 10 | Courier id |
| Uid | Int | 10 | User id |
| Cid | Int | 30 | Courier Id |
| Amount | Int | 10 | Amount |
| Date | Varchar | 30 | Date and time |

1. **SAMPLE CODING**

<html>

<head>

<link href="http://maxcdn.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css" rel="stylesheet" id="bootstrap-css">

<link href="/bootstrap/datatables.css" rel="stylesheet"/>

<link rel="stylesheet" type="text/css" href="/home/login.css">

</head>

<body>

<div class="container register">

<div class="row">

<div class="col-md-3 register-left">

<img src="https://image.ibb.co/n7oTvU/logo\_white.png" alt=""/>

<h3>Courier Tracking</h3>

</div>

<div class="col-md-9 register-right">

<div class="tab-content" id="myTabContent">

<div class="tab-pane fade show active" id="home" role="tabpanel" aria-labelledby="home-tab">

<h3 class="register-heading">Login</h3>

<div class="row register-form">

<div class="col-md-6">

</div>

<div class="col-md-6">

<div class="form-group">

<input id="username" type="text" class="form-control" placeholder="User Name \*" value="" />

</div>

<div class="form-group">

<input id="password" type="password" class="form-control" placeholder="Password \*" value="" />

</div>

<input type="button" onclick="login()" class="btnRegister" value="Login"/>

<a href="/track" class="btnRegister" value="Login">Track Courier</a>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</body>

<script src="/bootstrap/jquery.min.js"></script>

<!--<script src="/home/home.js"></script>-->

<script src="/bootstrap/datatable.min.js"></script>

<!--<script src="/bootstrap/datatable.bootstrap.min.js"></script>-->

<script src="http://maxcdn.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js"></script>

<script src="/bootstrap/bootstrap.min.js"></script>

<script>

function login() {

var username = $("#username").val();

var password= $("#password").val();

$.ajax({

type: 'POST',

url: '/api/login',

data: {

username: username,

password: password

}

}).done(function (datas) {

if(datas[0].count==0){

alert("Invalid username and password")

}else{

window.location="/home"

}

});

}

</script>

</html>

<html>

<head>

<link href="/bootstrap/bootstrap.css" rel="stylesheet"/>

<link href="/bootstrap/datatables.css" rel="stylesheet"/>

<link rel="stylesheet" type="text/css" href="/home/home.css">

</head>

<body>

<div>

<nav class="navbar navbar-default">

<div class="container">

<div class="navbar-header">

<button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#navbar"

aria-expanded="false" aria-controls="navbar">

<span class="sr-only">Menu</span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

</button>

<a class="navbar-brand" href="index.php">Courier Tracking</a>

</div>

<div id="navbar" class="navbar-collapse collapse">

<ul class="nav navbar-nav">

<li><a href="#htab1" data-toggle="tab">Add Customer</a></li>

<li><a href="#htab2" data-toggle="tab">View Customer</a></li>

<li><a href="#htab3" data-toggle="tab">Add Courier</a></li>

<li><a href="#htab4" data-toggle="tab">Update Courier</a></li>

<li><a href="/">Logout</a></li>

</ul>

</div>

</div>

</nav>

<div class="row">

<div class="col-sm-12">

<div class="tab-content" style="margin: 30px">

<div role="tabpanel" class="tab-pane fade in active" id="htab1">

<h1>Add Customer</h1>

<div class="col-sm-4">

<div class="form-group">

<label>Name:</label>

<input type="text" class="form-control" id="name1">

</div>

<div class="form-group">

<label>Mobile Number:</label>

<input type="text" class="form-control" id="mobile1">

</div>

<div class="form-group">

<label>Full Address:</label>

<input type="text" class="form-control" id="address1">

</div>

<button onclick="addCustomer()" type="button" class="btn btn-primary">Add Customer</button>

</div>

</div>

<div role="tabpanel" class="tab-pane fade" id="htab2">

<h1>View Customers</h1>

<table class="table table-dark">

<thead>

<tr>

<th scope="col">#</th>

<th scope="col">Name</th>

<th scope="col">Mobile</th>

<th scope="col">Address</th>

</tr>

</thead>

<tbody id="customerbody">

</tbody>

</table>

</div>

<div role="tabpanel" class="tab-pane fade in" id="htab3">

<h1>Add Courier</h1>

<div class="col-sm-4">

<div class="form-group">

<label>Mobile:</label>

<input type="text" class="form-control" id="mobile3">

</div>

<div class="form-group">

<label>Name:</label>

<input type="text" class="form-control" id="name3">

</div>

<div class="form-group">

<label>Send Location:</label>

<input type="text" class="form-control" id="send3">

</div>

<div class="form-group">

<label>Full Address:</label>

<input type="text" class="form-control" id="address3">

</div>

<button type="button" onclick="addCourier()" class="btn btn-primary">Add Courier</button>

</div>

<div class="col-sm-8">

<table class="table table-dark">

<thead>

<tr>

<th scope="col">#</th>

<th scope="col">Name</th>

<th scope="col">Mobile</th>

<th scope="col">Address</th>

<th scope="col">Send Location</th>

<th scope="col">Current Location</th>

</tr>

</thead>

<tbody id="courierbody">

</tbody>

</table>

</div>

</div>

<div role="tabpanel" class="tab-pane fade in" id="htab4">

<h1>Update Courier</h1>

<div class="col-sm-4">

<div class="form-group">

<label>Courier No:</label>

<input type="text" class="form-control" id="courierno" >

</div>

<div class="form-group">

<label>Current place:</label>

<input type="text" class="form-control" id="currentlocation">

</div>

<button type="button" class="btn btn-primary" onclick="updateCourier()">Update Courier</button>

</div>

</div>

</div>

</div>

</div>

</div>

</body>

<script src="/bootstrap/jquery.min.js"></script>

<script src="/home/home.js"></script>

<script src="/bootstrap/datatable.min.js"></script>

<script src="/bootstrap/datatable.bootstrap.min.js"></script>

<script src="/bootstrap/bootstrap.min.js"></script>

</html>

<html>

<head>

<link href="/bootstrap/bootstrap.css" rel="stylesheet"/>

<link href="/bootstrap/datatables.css" rel="stylesheet"/>

<link rel="stylesheet" type="text/css" href="/home/home.css">

</head>

<body>

<div>

<nav class="navbar navbar-default">

<div class="container">

<div class="navbar-header">

<button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#navbar"

aria-expanded="false" aria-controls="navbar">

<span class="sr-only">Menu</span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

</button>

<a class="navbar-brand" href="index.php">Brand</a>

</div>

<div id="navbar" class="navbar-collapse collapse">

<ul class="nav navbar-nav">

<li><a href="#htab1" data-toggle="tab">Track Courier</a></li>

</ul>

</div>

</div>

</nav>

<div class="row">

<div class="col-sm-12">

<div class="tab-content" style="margin: 30px">

<div role="tabpanel" class="tab-pane fade in active" id="htab1">

<h1>Track Courier</h1>

<div class="col-md-4">

<div class="form-group">

<label>Courier Number:</label>

<input type="text" class="form-control" id="courierno">

</div>

<button onclick="search()" type="button" class="btn btn-primary">Search Courier</button>

</div>

<div class="col-md-12">

</div>

<div class="col-sm-4">

<div class="form-group">

<label>To Location:</label>

<input type="text" disabled class="form-control" id="tolocation">

</div>

<div class="form-group">

<label>Current Location:</label>

<input type="text" disabled class="form-control" id="currentlocation">

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</body>

<script src="/bootstrap/jquery.min.js"></script>

<!--<script src="/home/home.js"></script>-->

<script src="/bootstrap/datatable.min.js"></script>

<script src="/bootstrap/datatable.bootstrap.min.js"></script>

<script src="/bootstrap/bootstrap.min.js"></script>

<script>

function search() {

var courierno = $("#courierno").val()

$.ajax({

type: 'POST',

url: '/api/search',

data: {

courierno: courierno

}

}).done(function (datas) {

var data = datas[0];

$("#tolocation").val(data.sendlocation);

$("#currentlocation").val(data.currentlocation);

update();

});

}

</script>

</html>

1. **SAMPLE INPUT**
2. **SAMPLE OUTPUT**