|  |
| --- |
| Project Report on |
| Hire Truck |
| SSIP – Hackathon – 2018/19 |

**PROJECT PROFILE**

**Project title:**  HireTruck

**Project objective:**  To Provide Transportation Facility.

**Submitted to:** SSIP – GIH - Hackathon.

**University:** RakshaShakti University

**Mentor:** Prof. Archana B Gondaliya

**Team size:** 5

**Team members:** ParmarViral

ShahPrince

JoshiRidham

K K Gayatri Priyadarsini

Kesha Kaneria

**Roll of team member:**

Viral Parmar (Analyst, Developer)

Prince Shah (Analyst, DBAdm, Developer)

Ridham Joshi (Developer, DBAss)

K K Gayatri Priyadarsini (Designer, Developer)

Kesha Kaneria (Developer, Designer)

**Operating system:** Windows 10, 8.1, Linux.

**Front-end:** PHP (Server side)

JavaScript, BootStrap, CSS, Html (Client Side)

**Back-end:** MYSQL

**ACKNOWLEDGEMENT**

Firstly, we are grateful to all them who contributed to my project directly or indirectly and help me to complete this project satisfactorily. I sincerely feel that the credit of the project work could not be narrowed down to only one individual. The development of this project involves many valuable contributors. I like to thanks **Prof. Archana Gondaliya** (Team mentor) for their precious suggestions and continue support for our project work. At last, I expressed my feelings to all my friends and family for providing critical and caustic feedback and sources of inspiration for my project work.

We would also Thank SSIP – GIH – Hackathon to provide us Platform.

**ABSTRACT**

* In a developing country like India, there is a need for transporting goods from one place to another in every aspect of life, be it for business, shifting or any other need. So, we introduce "HIRE TRUCK” to provide easy access to any person who would like to ship their goods.
* It acts as a communication interface between the shipper and the transport companies and reduces the hassle in doing it otherwise without compromising the security of the users.
* It lets the shippers mention their requirements (like vehicle capacity, number, type etc) and post it in the form of an advertisement and being known to different transport companies.
* The transport companies as well have the option of choosing the advertisement they can provide transport for. The customer can choose the company and finalize the deal accordingly.
* This project uses PHP for front-end development and MySQL for Back-End development which is well known for providing an easier GUI based interface for users who are not that familiar with using computers.
* For designing the portal, we have used HTML-5 and CSS-3.

**NAMING CONVENTIONS**

|  |  |
| --- | --- |
| **Names and Codes** | **Conventions** |
| Comments | Comments are available on last of each and every page. |
| Variables | Variables are given under stable names.  Variables are declared in a consistent manner.  Always declare variables as a group at the beginning of each procedure, and always declare such variable on a separate line.  Boolean variable names should contain Is which implies Yes/No or True/False values.  Variable name begins with a small letter. |
| Classes | There are different event handling events from different components.  Class name begins with a small letter. |
| Functions | Functions names have their first letter as capital ‘func’ as a prefix string.  Use comments on functions explaining its task and working that what the function does.  Each function begins with a small letter and is given a meaningful name. |
| Name | Naming tables, express the name in the singular form.  All the table names started with ‘tbl’. |
| Key | Primary key start with ‘pk’ and its value is not null means null is not allowed for it. |

|  |  |
| --- | --- |
| **Variable Type** | **Naming Convention** |
| Character | Char |
| Integer | Int |
| Double | Dbl |
| Float | Flt |
| Boolean | Bln |
| Long | Lng |
| Short | Sht |
| Date | Dt |
| Time | Tm |
| Object | Obj |

**NOTATION**

**NOTATIONS FOR USE CASE DIAGRAMS:-**

Actor

Association <<Include>>, <<Exits>>, <<Uses>>

**NOTATIONS FOR E-R DIAGRAM:**



**NOTATIONS FOR ACTIVITY DIAGRAM:-**

Final State Initial State

Control Flow Object Flow

Note

Activity

Note Activity

Fork Join

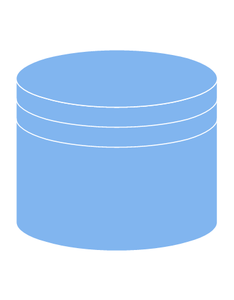
NOTATION FOR DATAFLOW DIAGRAM:

1. **Notations for Data Flow Diagram:**

Proccess

Entities

Flow of the Data

****

Database

**Introduction**

1. **Introduction**

* **Project summary**
* **Project Purpose**
* **Scope**
* **Technology & Literature Review**

**1**

**INTRODUCTION**

* 1. **PROJECT SUMMARY**

This Project entitled “Hire Truck” is developed using PHP as frontend and MySQL as the backend to computerize the process of transportation of goods through carriers by shippers.

This project help shippers to have easy access to transport their goods from one place to another, According to their comfortability, shippers can choose their desired vehicle that is available around their location. The bid system enhances the usage as per Transport Company to bid their luggage with respect to the price decided by the Shipper, User friendly interface help shippers to interact with Transport Company easily, New registration of Transport Company are also done through this project which allows them to involve in this activity as goods' carriers and interact with shippers to bid their goods.

* **Project Title: - Hire Truck**
* **Internal Guide: -** Prof. Archna B Gondaliya
* **Front End: - PHP**
* **Back End: -** MySQL
* **Operating System : -** Windows **10,Windows 8.1,Linux**
* **How does this System work?**
* Firstly the User (Shipper or transport) needs to register themselves.
* Shipper posts Ad using their requirements (Truck type, Luggage type, Weight, Source and destination).
* Transport Company bids on a particular Ad which is posted by the Shipper.
* Shipper sees the company list which is there to bid for that Ad.
* The Shipper chooses the suitable Transport Company to transport their Luggage.
* Shipper finals one Transport Company, soon the transport company gets an Acknowledgement from the shipper and finally, the Transport Company finalizes the deal with the Shipper.
* Shipper and Transport company then gets the E-Receipt.
  1. **PROJECT PURPOSE**
* Purpose of this project is to have easy access by shippers to contact the Transport Company.
* Bid system gives the best possible output to the shippers.
* The system is fair enough with both (Shipper and Transport Company) and finalizes the deal after the confessions of both sides.
* No waste of time, no waste of additional money and effortless work is done through this system.

**1.3 SCOPE**

* This system is useful to all those who want to transport their goods from one place to another.
* Users can find suitable and reliable transport company of their choice according to the location without any problem.
* Transport Company can reach their interested customers without even advertising at different places. They just have to register their company to the website.
* This website is made user-friendly so that even a naïve user can use it easily.
* This removes the hassle of actually traveling and talking to transport companies for shipping their goods and advertising the company.

**1.**4 **TECHNOLOGY AND LITERATURE REVIEW**

* **Requirement Determination & Specification**

Requirement determination involves studying the current business system to how it works and where improvements should be made. A requirement includes a way of capturing or processing data, producing information, controlling a business activity, or supporting management.

* **HTML**

In 1992, Berners-Lee and CERN team released the first version of HTML(HTML1.0). HTML stands for Hyper-Text Markup Language. A website consists of a collection of web pages. Using a special application tools called as HTML creates every web page. The browser will extension and interpret the HTML document. The set of instruction written in HTML is interpreted by the web browser. The only thing HTML does is classification parts of our document so that a browser of platforms. HTML can be used to put a document on not just computer screens, but also printers, fax machine etc.

**Advantages of HTML**

* Flexible
* Easy to use
* Inexpensive
* Platform independent
* Not case sensitiv**e.**
* **Client side**
* **CSS**

Cascading Style Sheets (CSS) is a style sheet language used to describe the presentation **of** the look and formatting of a document written in a markup language. It’s most common application is to style web pages written in HTML and XHTML, but the language can also be applied to any kind of XML document, including plain XML andSVG.CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colors, and fonts **etc**.

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.

* **PHP**

PHP stands for Hypertext Preprocessor. PHP is a widely used free, open source, the general purpose scripting language that is especially suited for web development that can be embedded into HTML and interact with the database. **PHP is a server-side scripting language.** PHP is efficient for making dynamic and attractive web pages**/websites**. PHP applications are normally found on the Linux server and in conjunction with the MySQL database. PHP can perfect in any task that any CGI program can do, but its strength lies in its compatibility with many types of database.

In an HTML document, PHP script is enclosed within special PHP tags. Because PHP is embedded within tags, the developer can jump between HTML, PHP instead of having to rely on heavy amounts of code to out HTML. And because PHP is executed on the server, the client cannot view the PHP code.

* **MySQL**

MySQL is a widely used backend database. MySQL is free and open source. The data in MySQL has stored objects called tables. A table is a collection of related data entries and it consists of columns and rows. The database is useful when storing information.

Here a database to be maintained to carry out the tables needed for this project. Those tables which are created should be maintained and updated if necessary.

Project Management

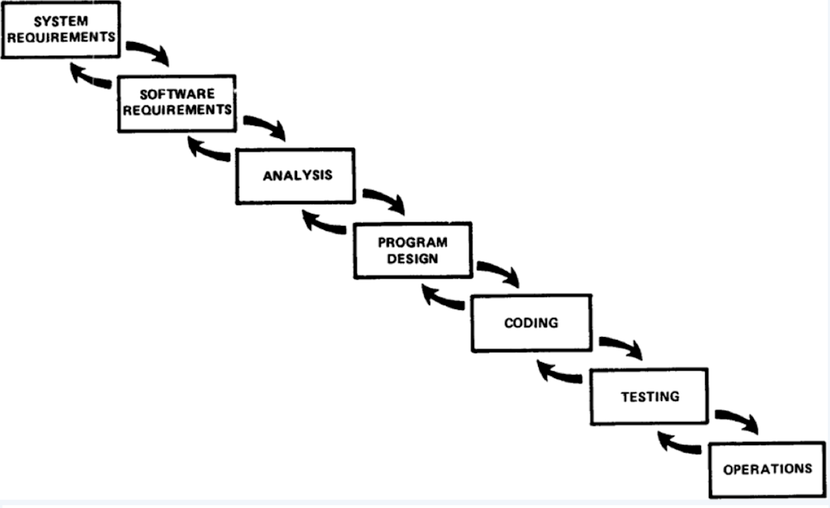
1. **Project Management**

* **Project Planning and Scheduling**
* **Risk Management**

**2 PROJECT MANAGEMENT**

**2.1 PROJECT PLANNING AND SCHEDULING**

**2.1.1 Project Development Approach and Justification**

****

[FIGURE 2: ITERATIVE WATERFALL MODEL]

An iterative [**life cycle model**](http://istqbexamcertification.com/what-are-the-software-development-models/) does not attempt to start with a full specification of requirements. Instead, development begins by specifying and implementing just part of the software, which can then be reviewed in order to identify further requirements. This process is then repeated, producing a new version of the software for each cycle of the model

When we work **iteratively**we create a rough product or product piece in one iteration, then review it and improve it in the next iteration and so on until it's finished. As shown in the image above, in the first iteration the whole painting is sketched roughly, then in the second iteration colors are filled and in the third iteration, finishing is done. Hence, in the iterative model, the whole product is developed step by step.

**Advantages of the Iterative model:-**

* In the iterative model, we are building and improving the product step by step. Hence we can track the defects at early stages. This avoids the downward flow of the defects.
* In the iterative model, we can get reliable user feedback. When presenting sketches and blueprints of the product to users for their feedback, we are effectively asking them to imagine how the product will work.
* In the iterative model, less time is spent on documenting and more time is given for designing.

The project members can work parallel and no need to wait for the other member to complete the dependent task.

**2.1.2 PROJECT PLAN**

* Project scheduling is one of the main key aspects of any project. Any project must be scheduled before developing it.
* When project developer works on a scheduled project it is more advantageous for him/her to compare to an unscheduled project. It gives us a timeline for finishing the particular activity. Scheduling gives us an idea about project length, its cost, its normal duration of completion and we can also find out the shortest way to complete the project with the less overall cost of the project.
* Project schedule describes dependency between activities. The estimated time required to reach each milestone and allocation of people to activities.

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.**  **No.** | **Milestones** | **Milestone**  **(days)** | **Duration**  **(days)** |
| **1** | Requirements | 15 | 17 |
| **2** | Analysis and Module Specification | 2 | 7 |
| **3** | Designing | 20 | - |
| **4** | Coding | 15 | - |
| **5** | Testing & Implementation | 10 | - |
| **6** | Documentation | 35 | 27 |
|  | Total days | 42 | 51 |

* **Roles and Responsibilities:-**

We are three persons involved in this project. For any software development, there is always human organization structures. Our team structure is Democratic Decentralized, so there is no permanent leader in our team. We take decisions regarding our project's problems discussing with each other. Project partners are also interested to do something innovated and put marvelous efforts. It has been found that this project would require a large amount of communication sociability

There is no particular role has been defined between us. We have take responsibilities according to situation occur during the development of this project. System Design, coding, testing, and documentation these all task have been completed by us. No task has been completed by an individual.

|  |  |
| --- | --- |
| **RESPONSIBILITY** | **MEMBER** |
| **Analysis** | Viral, Prince, Gayatri |
| **Design** | Gayatri, Kesha |
| **Coding** | Viral, Prince, Ridham, Gayatri, Kesha |
| **Testing & Implementation** | Viral, Prince, Ridham |
| **Documentation** | Viral, Prince |

**2.1.3 SCHEDULE REPRESENTATION:**

The project schedule provides a road map for a software project manager. If it has been properly developed, the project schedule defines the task and milestones that must be tracked and controlled as the project proceeds. Tracking can be accomplished in a number of different ways.

* Conducting periodic project status meeting in which each team member reports progress and problems.
* Evaluating the results of all reviews conducted throughout the software engineering process.
* Determining whether formal project milestone has been accomplished by the scheduled date.
* Comparing actual start date for each project task listed in the resource table.
* Meeting informally with practitioners to obtain their subjective assessment of progress to date and problems on the horizon.
* Using earned value analysis to assess progress quantitatively.

Control is employed by a software project manager to administrator project resources, cope with problems, and direct project staff. If things are going well, control is light. But when problems occur, the project manager must exercise control to reconcile them as quickly as possible.

* **Schedule Table**

|  |  |  |
| --- | --- | --- |
| **Description** | **Estimated days** | **Actual Days** |
| System Study | 13 | 20 |
| Requirement Gathering | 10 | 12 |
| Analysis | 30 | 25 |
| System Design | 20 | 19 |

**2.2 RISK MANAGEMENT**

Project Risk Management involves conducting risk management planning, engaging in risk identification, completing risk analysis, creating a risk response action plan, and monitoring and controlling risk on a project. Project Risk Management is a continuous process to be engaged in throughout the entire project. The purpose of project risk management is to increase the likelihood and impact of positive events and to decrease the probability and impact of negative events. Each Risk Management process results in a specific deliverable which is used as the foundations for the subsequent process. Combined the risk management processes provide a best practice pattern for managing risk on a project.

#### **Characteristics of Risk**

#### **Uncertainty** – the risk may or may not happen i.e., there aren’t 100% probable risks.

#### **Loss** – if the risk becomes a reality, unwanted consequences or losses will occur.

#### **2.2.1 RISK IDENTIFICATION:**

Risk Identification is a systematic attempt to specify threats to the project plan. By identifying the known and predictable risks, the project manager takes a first step towards avoiding them when possible and controlling them when necessary.

One method of identifying risks is to create a risk item checklist. The checklist can be used for risk identification and focuses on some subset of known and predictable risks in the following subcategories.

#### Following risks were identified and taken under consideration for our project

#### **Project Risk:**

#### Threaten the project plan. That is, if project risks become real, it is likely that the project schedule will slip and that costs will increase. Project risks identify potential budgetary, schedule, personnel staffing and organization, resource, customer, and requirements problems and their impact on a software project.

#### **Technical Risks:**

#### Threaten the quality and timeliness of the software to be produced. If a technical risk becomes a reality, the implementation may become difficult or impossible. Technical risks identify potential design, implementation, interface, verification, and maintenance problems. In addition, specification ambiguity, technical uncertainty, technical uncertainty, technical obsolescence, and "leading-edge" technology are also risk factors. Technical factors occur because the problem is harder to solve than we thought it would be.

**2.2.2 RISK ANALYSIS**

*Regardless of the prevention techniques employed, possible threats that could arise inside or outside the organization need to be assessed. Although the exact nature of potential disasters or their resulting consequences is difficult to determine, it is beneficial to perform a comprehensive risk assessment of all threats that can realistically occur to the organization.*

#### **Project Risk: -** The project risks for our project include that our project schedule will slip and the project may not be completed within the deadlines. This risk poses threat to the project plan. The costs of post risk measures and development may increase.

#### **Technical Risk: -** The technical risks for our project may include the errors that can be generated in the code or the drawbacks that the system may possess. This includes the failure of the programmers to implement certain functionalities like a tooltip, dynamic display of temperature, pressure etc. If technical risk becomes real, the desired implementation may become difficult or impossible.

* **Performance Risk: -** The degree of uncertainty that the product will meet its requirements and be fit for its intended use. As we are trainees, the performance risk is there in our project.
* **Support Risk: -** The degree of uncertainty that the resultant software will be easy to correct, adapt, and enhance. The client of this project is on different premises.
* **Technical Risks: -** Threaten the quality and timeliness of the s/w to be produced. If technical risk becomes real, the implementation may become difficult or impossible. Technology is already decided for this project.
* **Business Risks: -** Threaten the viability of the s/w to be built. This is a live project so no business risk is there.
* **Known Risks -** Are those that can be uncovered after careful evaluation of the project plan.

**2.2.3 RISK PLANNING**

#### Following measures are taken by the developers to avoid and manage the occurrence of risks:

#### **Project Risk:**

#### The schedule for analysis, design, coding, and testing should be pre-planned. The steps should be taken to ensure that the schedule is maintained and we do not lack behind in time for completing the project within deadlines. The various cost-cutting measures should be implemented in advance.

#### **Technical Risks:**

#### To avoid technical risks from becoming real, the coding should be started in advance and longer time slots should be allocated for tougher code portions. In case technical risks arise, then immediate steps to allocate more programmers to the task, managing the errors before launching the system in the market should be taken.

* **Business Risks** – Threaten the viability of the s/w to be built. This is a live project so no business risk is there.
* **Known Risks** – are those that can be uncovered after careful evaluation of the project plan.
* **Experience Risks** – Are extrapolated from past project experience. As we are trainees, make this project no such experience is there.
* Meet with current staff to determine causes for turnover (e.g. poor working conditions, low pay competitive job market)
* Mitigate those causes that are under our control before the project starts.
* Organize project teams so that information about each development activity is widely dispersed.
* Conduct peer reviews of all work. Organize project teams so that information about each development activity is widely dispersed.

System Requirement Study

1. **System Requirement Study**

* **User Characteristics**
* **Hardware and Software Requirements**
* **Constraints**

**3**

**SYSTEM REQUIREMENT STUDY**

**3.1 USER CHARACTERISTICS**

Analyzing user characteristics is an important aspect of any project. It allows us to clearly define and focus on who the end users are for the project. Also, it allows checking the progress of the project to ensure that we are still developing the system for the end users. The user must have the following characteristics:

* The user must have a basic knowledge of Computers.
* The user should understand the use of all modules.
* The user can easily interact with the proposed system.
* The user must know the technical terms used in the company for performing different tasks.
* The user should also be aware of the running process of the system.

The HireTruck have 3 types of users.

**1) Admin**

Admin can create a new admin user to maintain the site in a better way. Only admin have that right to Block User. Admin can manage the Shipper and Transport Company. Admin can maintain their personal profile & can update as per needed. Admin can make updatation as per requirement. Admin can also view user’s feedback and changes in this site.

**2) Shipper**

Shipper will first registration to the site and then log in. After login, they can able to post Ad on site for transport their luggage. Shipper checks Terms & Conditions. Shipper also posts Their feedback on Site.

**3) Transport Company**

Transport Company will first register on site. After login by Transport Company can find Customer on his/her city. And Transport Company show which Shipper in his city on his/her area. If the deal will be done then they show the Shipper's basic information like Name, City, Area, and contact no, Transport Company checks Terms & Conditions, also, post his/her feedback.

**3.2 HARDWARE AND SOFTWARE REQUIREMENTS**

* **Hardware requirements**
* Intel Core i3 2.0 GHz and more.
* 2GB & 4 GB RAM.
* Windows 7, 8.1, 10, Linux.
* 1TB HDD.
* All Standard types Monitors and Laptops.
* **Software requirements**
* Windows 7, Windows 8.1, Windows 10 and Linux Operating System.
* Front End: - PHP (7.2.3).
* Back End: MySQL.
* Server: Apache
* IDE: Atom
* Browser: Chrome, Firefox, Edge

3.3 **CONSTRAINTS**

**3.3.1 Hardware Limitations**

This system work with the hardware defined above or higher. This system is available for any registered user to use this site.

3.3.2 **Criticality of the Application:-**

To run this project, it is necessary that you have the following components installed:

* Xamp server
* Browser

## 3.3.3 Regulatory Policies

As all the documents are confidential they will be only available to me only.

Documents will be uploaded by the Administrator only.

## 3.3.4 Interfaces to Other Applications

This system is a web application and can be used in future in the online dealing company's website so that it can access anywhere. The company is also planning to interface the applications to record different technical parameters automatically. In the first phase, both manual & automatic deals will be posted which would then buy. However, if the reliability of the system & effectiveness of the software is ensured then the company would manual managing deal process & would make deal updating & report generation automatic.

## 3.3.5 Parallel Operations

At a time users can access the System for the same deal. It means at a time any number of deal can be bought at any place. The Quick Deal application has different modules. Admin module is used for keeping records of customers, merchant, deals. It will manage all customers and merchant. It is also managing deals, payment. It also manages the successful deal buying module generating the successful deals and success id generating success id. The data would then be stored in a database. Since all of the modules work simultaneously, it requires concurrent updating of the database. This problem can thus be solved by effective use of the database management system. Use of transaction remove this problem and would also ensure updating & deletion of the database.

3.3.6 **Reliability Requirements**

The main reliability requirement is the validation used. Without proper validation, the system does not allow entering that value into the database.

In this project, there are many forms. In forms, if there are any mandatory fields, then for that alert message have been used. So it is for the user to fill that information if the user is unaware.

## 3.3.7 Reliability Requirements

The main reliability requirement is a validation of the entered data. Without proper validation, the system does not allow to enter that value in the database. For example in email id the user cannot enter any dummy value, the validation checks that whether there is ‘@' or ‘.' symbol. Also, any null value is not allowed in place of compulsory fields. The system also uses transaction features such as whenever the transaction is not fully completed (i.e. due to power failure) it is rollbacked. The only complete transaction allows successful updating of data to the data store.

## 3.3.8 Reliability Requirements

As the application is using the latest technology, defects in the technology can create big issues. Moreover, the application requires the provision of data manually; hence human errors can be found critical. As the application is in its initial phase of development, hence it's prone to many hidden errors, which arise on late of the application. The user is required to be trained before the application is being used. Illiteracy in handling the application can create problems.

## 3.3.9 Reliability Requirements

Without proper authentication, unauthorized users will be unable to alter the database. For e.g.: without administrator password able to create or delete the allocated and user updating of records. Unauthorized access, revelation, or destruction of data can violate individual privacy. Corruption of business data can result in significant and potentially catastrophic losses to companies. In this software, security threats are at the user level as well as on data level.

**System Design**

1. **System Design**

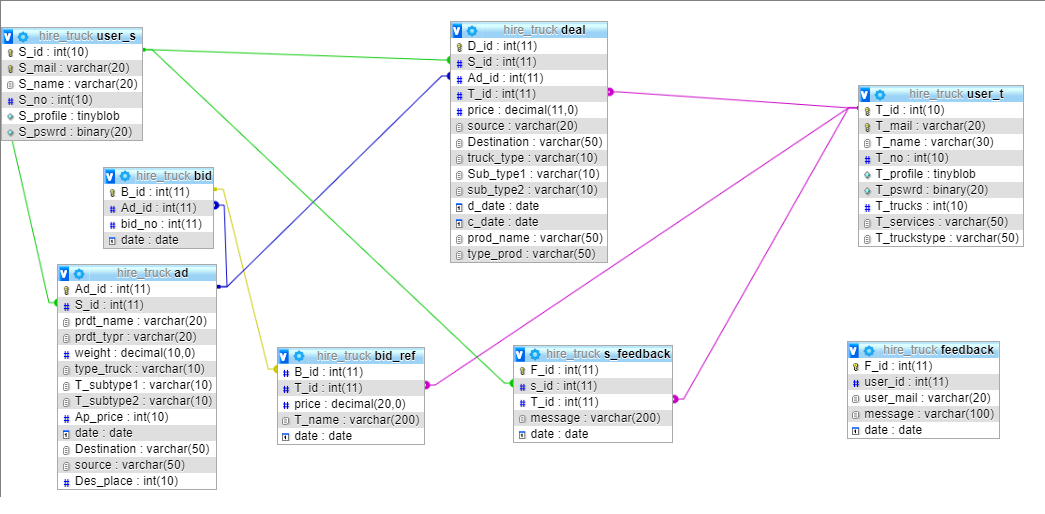
* **Database design / Data Structure Design**
* **System Procedural Design**
* **Input / Output Interface Design**

**4**

**SYSTEM DESIGN**

**4.1 DATABASE DESIGN/DATA STRUCTURE DESIGN**

**4.1.1 TABLES AND RELATIONSHIP**

****

**Figure 4.1:Tables & Relationship**

**4.2 SYSTEM PROCEDURAL DESIGN**

**4.2.1 CODE OR ALGORITHM FOR METHOD OR OPERATIONS**

1. Type the Appropriate URL into the address bar of the web browser.
2. The display registration page for users to register their company.
3. Open the login page of Website.
4. Open the home page of Website.
5. Select the appropriate module.
6. Open any module
7. Upload Ad here.
8. The user enters there requirement details.
9. Contents are submitted and shoe relates bid.
10. Select search ads.
11. Show bid on the ad or final deal with the company.
12. Logout
13. Else
14. No change.
    1. **INPUT/OUTPUT AND INTERFACE DESIGN**

**4.3.1 SAMPLE OF FORMS**

**4.3.2 ACCESS CONTROL AND SECURITY**

Visitor users can access this site. But Visitor user can only view Shipper and search as well as they can give feedback and show contact us and about us. They cannot search Transport company and Post ad But Registered User can do All process of Visited user as well as They feel the Shipper registration But They cannot edit or update any other details of this site. But admin can do edit, update and view detail of all users (Shipper as well Transport Company).

**Implementation Planning and Details**

1. **Implementation Planning and Details**

* **Implementation Environment.**
* **Program / Modules Specification.**
* **Security Features**
* **Coding Standards**
* **Sample Coding**

**5**

**IMPLEMENTATION PLANNING AND DETAILS**

5.1 IMPLEMENTATION ENVIRONMENT

The project was a result of a group consensus. The team was having five members. The team structure depends on the management style of the organization, the number of people in the team, their skill levels, and the problem difficulty.

This website is basically GUI related, there is well-structured menu and images have been provided for the user, for identifying the information displayed on Screen.

It allows multi-user facility if there are a number of the client machine and on the server there is a Transportation site, which can be accessed by any client at the same time.

For implementing code, we use Wamp2.5. Xampp is open source, that is preinstalled with

|  |  |
| --- | --- |
| Apache (2.4.9) | Work as Web Server handle request and response |
| MySQL (5.6.17) | Manage database |
| PHP (7.2.3) | Server Side Scripting Language |
| phpMyAdmin (4.1.14) | Manipulate MySQL Database |

6.2 PROGRAMS/ MODULES SPECIFICATION

* This application is divided into four categories.

1. Shipper/Transport Company Registration.
2. Post Ad
3. Bid on Ad
4. Search Users (Shipper/Transport Company)
5. Feedback (Shipper/Transport Company)
6. Generate E-Receipt
7. Show Deal(User)

* In HireTruck, User can fill up the registration form and user’s information or and they are allowed for Post Ad and Bid.
* Admin can see the information about all registrations and uploaded data, the admin has right to Block or Unblock the User (Transport company/Shipper).
* Admin can also react on Report.

5.3 SECURITY FEATURES

Security is a very wide-reaching term during every step of application building process you must, without doubt, be aware of how mischievous end users might attempt to bypass your lockout measures we are using here Respect Library to Validate the user for security.

Not every page that you build with PHP is meant to be open and accessible for only everyone on the Internet. Sometimes you want to build pages or section of a website that are accessible to only a select group of your choosing. For these reasons you need security measures.

Security can help to protect data behind your website from fraudulent use. We must take a step to ensure that no one can take over the application or gain access to its resources.

* **Two types of security measures are:**
* **Authentication**

Authentication is a process that determines the identity of user we also identify the user is not a robot (for security reason or to avoid Bot).

* **Authorization**

Authorization is a process of determining whether an authenticated user is permitted access to any part of an application. Or access only to specified data sets that the application provides.

* **Basic Authentication:**

This authentication requires a user name and password from the client. The main advantage of basic authentication is that is supported by most browsers and then a negative aspect of basic authentication is that it passes the user name and password to the server as clear text.

**5.4 CODING - STANDARDS**

Normally, good website development organization requires their programmers to add here to some well-defined and standard style of coding called coding standard.

In BISAG, they have also some representatives regarding coding standard

1. For each module, which is added into existing software, must be private.
2. Do not modify global variables.
3. Whenever required to add a new method, specify for what purpose you are going

To add it, date, author name.

1. All constant variables must have to write into all capital letter.
2. If need to changes into any existing method then specify for what purpose

Changes are required and modification date with author name.

1. Specify the name of the module at the starting of the module.

Besides this there are also some coding guidelines regarding project development.

1. Do not use a coding style that is too difficult to understand.
2. Do not use an identifier for the multiple purposes.
3. Each variable should be given a descriptive name.
4. The code should be well documented.
5. Avoid the side effect of the function call.

**Coding Style**

**PHP Coding Demarcation**

PHP code must always be delimited by the full-form, standard PHP tags. Short tags are only allowed within view scripts.

**String Literals**

When a string is literal (contains no variable substitutions), the apostrophe or "single quote" must always use to demarcate the string:

**String Literals Containing Apostrophes**

When a literal string itself contains apostrophes, it is permitted to demarcate the string with quotation marks or "double quotes".

**String Concatenation**

Strings may be concatenated using the "." operator. Space must always be added before and after the "." operator to improve readability.

When concatenating strings with the "." operator, it is permitted to break the statement into multiple lines to improve readability. In these cases, each successive line should be padded with whitespace such that the "." operator is aligned under the "=" operator:

**Numerically Indexed Arrays**

Negative numbers are not permitted as array indices. An indexed array may be started with any non-negative number, however, this is discouraged and it is recommended that all arrays have a base index of 0. When declaring indexed arrays with the array construct, a trailing space must be added after each comma delimiter to improve readability.

It is also permitted to declare multi-line indexed arrays using the array construct. In this case, each successive line must be padded with spaces such that beginning of each line and each value aligns as shown below.

**Associative Arrays**

When declaring associative arrays with the array construct, it is encouraged to break the statement into multiple lines. In this case, each successive line must be padded with whitespace such that both the keys and the values are aligned

**Function and Method Declaration**

Functions and class methods must be named by following the naming conventions. Methods must always declare their visibility by using one of the private, protected, or public constructs. As for classes, the opening brace for a function or method is always written on the line underneath the function or method name ("one true brace" form). There is no space between the function or method name and the opening parenthesis for the arguments.

The return value must not be enclosed in parentheses. This can hinder readability and can also break code if a function or method is later changed to return by reference.

**Function and Method Usage**

Function arguments are separated by a single trailing space after the comma delimiter. Call-time pass-by-reference is prohibited. Arguments to be passed by reference must be defined in the function declaration. For functions whose arguments permit arrays, the function call may include the "array" construct and can be split into multiple lines to improve readability.

**If / Else / Else if**

Control statements based on the "if", "else", and "else if" constructs must have a single space before the opening parenthesis of the conditional, and a single space between the closing parenthesis and opening brace.

Within the conditional statements between the parentheses, operators must be separated by spaces for readability. Inner parentheses are encouraged to improve the logical grouping of larger conditionals.

The opening brace is written on the same line as the conditional statement. The closing brace is always written on its own line. Any content within the braces must be indented four spaces. For "if" statements that include "else if" or "else", the formatting must be as in these. Use of the "else if" construct is not allowed in favor of the "else if" combination.

**Testing Strategy**

1. **Testing Strategy**

* **Testing Plan**
* **Testing**
* **Testing Method**
* **Test Cases**

**6**

**TESTING STRATEGY**

* 1. **TESTING PLAN:-**

A test plan is the cornerstone of a successful testing implementation. The test plan represents the overall approach to the test. In many ways, the test plan serves as a summary of the test activities that will be performed. It shows how the tests will be organized and outlines all of the testers' needs that must be met in order to properly carry out the test.

The goal of test planning is to establish the list of tasks that, if performed, will identify all of the requirements that have not been met in the software. There are many standards that can be used for developing test plans. Early in the deployment planning phase, the testing team creates a test plan. The test plan defines the objectives and scope of the testing effort and identifies the methodology that your team will use to conduct tests. It also identifies the hardware, software, and tools required for testing and the features and functions that will be tested. A well-rounded test plan notes any risk factors that jeopardize testing and includes a testing schedule.

So we can say that Test Planning details the activities, dependencies, and effort required to conducting the System Test.

* **Unit testing**

Software products are normally tested first at the individual component (unit) level. Unit testing (or module testing) is the testing of different units (or modules) of a system in isolation.

* **Integration testing**

After testing all the components individually, the components are slowly integrated and tested at each level of integration. That is called integration testing.

* **System testing**

Finally, the fully integrated system is tested that is called system testing.

**Software testing is one of the “verification and validation,” or V&V, software practices:**

Ø **Verification** is the process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase*.* Verification activities include testing and reviews.

**Validation** is the process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements. At the end of development validation (the second V) activities are used to evaluate whether the features that have been built into the software satisfy the customer requirements and are traceable to customer requirements. Through verification, we make sure the product behaves the way we want it to.

Through validation, we check to make sure that somewhere in the process a mistake hasn’t been made such that the product build is not what the customer asked for; validation always involves comparison against requirements.

**6.2 TESTING**

We begin by ‘testing-in-the-small’ and move toward ‘testing-in-the-large’

For conventional website

The module (component) is our initial focus

Integration of modules follows

* + For OO software

Our focus when “testing in the small” changes from an individual module (the conventional view) to an OO class that encompasses attributes and operations and implies communication and collaboration

State testing objectives explicitly.

Understand the users of the software and develop a profile for each user category.

Develop a testing plan that emphasizes “rapid cycle testing.”

Build “robust” software that is designed to test itself

Use effective formal technical reviews as a filter prior to testing

Conduct formal technical reviews to assess the test strategy and test cases themselves.

Develop a continuous improvement approach for the testing process.



**unit test**

**integration**

**test**

**validation**

**test**

**system**

**test**

***Fig 6.1 Testing strategy***

**6.3 TESTING METHODS**

Software testing methods are traditionally divided into [black box testing](http://en.wikipedia.org/wiki/Black_box_testing) and [white box testing](http://en.wikipedia.org/wiki/White_box_testing). These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

**Black box testing**

[Black box testing](http://en.wikipedia.org/wiki/Black_box_testing) treats the software as a "black box," without any knowledge of internal implementation. Black box testing methods include [equivalence partitioning](http://en.wikipedia.org/wiki/Equivalence_partitioning), [boundary value analysis](http://en.wikipedia.org/wiki/Boundary_value_analysis), [all-pairs testing](http://en.wikipedia.org/wiki/All-pairs_testing), [fuzz testing](http://en.wikipedia.org/wiki/Fuzz_testing), [model-based testing](http://en.wikipedia.org/wiki/Model-based_testing), [traceability matrix](http://en.wikipedia.org/wiki/Traceability_matrix), [exploratory testing](http://en.wikipedia.org/wiki/Exploratory_testing), and specification-based testing.

**Fig 6.2**

**Input**

**Output**

**?**

* + - **Specification-based testing**

Specification-based testing aims to test the functionality of software according to the applicable requirements. Thus, the tester inputs data into, and only see the output from, the test object. This level of testing usually requires thorough test cases to be provided to the tester, who then can simply verify that for a given input, the output value (or behavior), either "is" or "is not" the same as the expected value specified in the test case.

Specification-based testing is necessary, but it is insufficient to guard against certain risks.

* + - **Advantages and disadvantages**

The black box tester has no "bonds" with the code, and a tester's perception is very simple: a code must have bugs. Using the principle, "Ask and you shall receive," black box testers find bugs where programmers don't. But, on the other hand, black box testing has been said to be "like a walk in a dark labyrinth without a flashlight," because the tester doesn't know how the software being tested was actually constructed. That's why there are situations when

1. a black box tester writes many test cases to check something that can be tested by only one test case, and/or
2. Some parts of the back end are not tested at all.

Therefore, black box testing has the advantage of "an unaffiliated opinion," on the one hand, and the disadvantage of "blind exploring," on the other.

* + **White box testing**

[White box testing](http://en.wikipedia.org/wiki/White_box_testing), by contrast to black box testing, is when the tester has access to the internal data structures and algorithms.

Types of white box testing

The following types of white box testing exist:

[API testing](http://en.wikipedia.org/wiki/Api_testing) - Testing of the application using Public and Private APIs.

[Code coverage](http://en.wikipedia.org/wiki/Code_coverage) - creating tests to satisfy some criteria of code coverage. For example, the test designer can create tests to cause all statements in the program to be executed at least once.

[Fault injection](http://en.wikipedia.org/wiki/Fault_injection) methods.

[Mutation testing](http://en.wikipedia.org/wiki/Mutation_testing) methods.

[Static testing](http://en.wikipedia.org/wiki/Static_testing) - White box testing includes all static testing.

**Code completeness evaluation**

White box testing methods can also be used to evaluate the completeness of a test suite that was created with black box testing methods. This allows the software team to examine parts of a system that are rarely tested and ensures that the most important [function points](http://en.wikipedia.org/wiki/Function_points) have been tested.

Two common forms of code coverage are:

* + - function coverage, which reports on functions executed
    - And statement coverage, which reports on the number of lines executed to complete the test.

They both return coverage metric measured as a percentage.

**Grey Box Testing**

In recent years the term **grey box testing** has come into common usage. This involves having access to internal data structures and algorithms for purposes of designing the test cases, but testing at the user, or black-box level. Manipulating input data and formatting output does not qualify as "grey-box," because the input and output are clearly outside of the "black box" that we are calling "the software under test." (This distinction is particularly important when conducting [integration testing](http://en.wikipedia.org/wiki/Integration_testing) between two modules of code written by two different developers, where only the interfaces are exposed for the test.) Grey box testing may also include [reverse engineering](http://en.wikipedia.org/wiki/Reverse_engineering#Reverse_engineering_of_software) to determine, for instance, boundary values or error messages.

**Acceptance testing**

Main article: [acceptance testing](http://en.wikipedia.org/wiki/Acceptance_testing)

Acceptance testing can mean one of two things:

1. A [smoke test](http://en.wikipedia.org/wiki/Smoke_test) is used as an acceptance test prior to introducing a build to the main testing process.
2. Acceptance testing performed by the customer is known as [user acceptance testing](http://en.wikipedia.org/wiki/Acceptance_testing#User_acceptance_testing) (UAT). 
   * **Non Functional Software Testing**

Special methods exist to test non-functional aspects of software.

[Performance testing](http://en.wikipedia.org/wiki/Software_performance_testing) checks to see if the software can handle large quantities of data or [users](http://en.wikipedia.org/wiki/Load_testing). This is generally referred to as software [scalability](http://en.wikipedia.org/wiki/Scalability). This activity of Non-Functional Software Testing is often times referred to as Load Testing.

[Usability testing](http://en.wikipedia.org/wiki/Usability_testing) is needed to check if the user interface is easy to use and understand.

[Security testing](http://en.wikipedia.org/wiki/Security_testing) is essential for software which processes confidential data and to prevent [system intrusion](http://en.wikipedia.org/wiki/Backdoor_%28computing%29) by [hackers](http://en.wikipedia.org/wiki/Hacker_%28computer_security%29).

[Internationalization and localization](http://en.wikipedia.org/wiki/Internationalization_and_localization) are needed to test these aspects of software, for which a [pseudo localization](http://en.wikipedia.org/wiki/Pseudolocalization) method can be used.

In contrast to functional testing, which establishes the correct operation of the software (correct in that it matches the expected behavior defined in the design requirements); non-functional testing verifies that the software functions properly even when it receives invalid or unexpected inputs. [Software fault injection](http://en.wikipedia.org/wiki/Fault_injection), in the form of [fuzzing](http://en.wikipedia.org/wiki/Fuzz_testing), is an example of non-functional testing. Non-functional testing, especially for software, is designed to establish whether the device under test can tolerate invalid or unexpected inputs, thereby establishing the robustness of input validation routines as well as error-handling routines. Various commercial non-functional testing tools are linked from the [Software fault injection](http://en.wikipedia.org/wiki/Fault_injection) page; there are also numerous open-source and free software tools available that perform non-functional testing.

In software engineering, the most common definition of a test case is a set of conditions or variables under which a tester will determine if a requirement or use case upon an application is partially or fully satisfied. In the situation, each sub-requirement must have at least one test case.

* **Path Coverage:**

The path coverage based testing strategy requires us to design test cases such that all linearly independent paths in the program are executed at least once.

* **Alpha Testing:**

Alpha testing refers to the system testing carried out by the test team within the developing organization.

* **Beta Testing:**

Beta testing refers to the system testing performed by a select group of friendly customers.

**6.3 TEST CASE**

**Test Case Design :**

The design of tests for software and other engineered products can be as challenging as the initial design of the product itself. Software engineers often treat testing as an afterthought, developing test cases that may “feel right” but have little assurance of being complete. Recalling the objectives of testing, one must design tests that have the highest likelihood of finding the most errors with a minimum amount of time and effort.

*Any engineering product can be tested in one of the two ways:*

* Knowing the specified function that a product has been designed to perform, tests can be conducted that demonstrate each function is fully operational, at the same time searching for errors in each function.
* Knowing the internal workings of a product, tests can be conducted to ensure that “all gears mesh,” that is, that internal operation performs according to specification and all internal components have been adequately exercised. The first test approach is called black box testing and the second, white box testing.

In software engineering, the most common definition of a test case is a set of conditions or variables under which a tester will determine if a requirement or use case upon an application is partially or fully satisfied. In the situation, each sub-requirement must have at least one test case.

* A good test has a high probability of finding an error. To achieve this goal the tester must understand the software and attempt to develop a mental picture of how the software might fail.
* A good test is not redundant. Testing time and resources are limited. There is no point in conducting that has the same purpose as another test.
* A good test should be the best of them all. In a group tests that have a similar intent, time and resources limitation may mitigate the execution of only a subset of these tests.
* A good test should neither be too simple nor be too complex.

**Some of the Test Cases are as mentioned below :**

**Test Case 1: Login :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SR NO.** | **TEST CASE** | **EXPECTED O/P** | **ACTUAL O/P** | **RESULT** |
| 1 | Entered correct values of username & password &click “login” | Allow going the home page | Allow going the home page | Pass |
| 2 | Missing values of username & password | Display error message | Display error message | Pass |
| 3 | Incorrect values of username & password | Display error message | Display error message | Pass |

***Table 6.1***

**6.4.2 Registration Test Cases**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SR.No. | Test Case | Expected Output | Actual Output | Test Case Status |
| 1. | Enter Valid email, password. | Success! | Send control to another page. | Test Passed. |
| 2. | If Email is not in proper format. | Register Failed. | Validation occurs. Message:  “Enter valid email  address.” | Test Passed. |
| 3. | If one or all fields are NULL. | Register Failed. | Validation occurs. Message:  “This field is  required” | Test Passed. |

**Test Case 2: GUI Testing**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No.** | **Test Case** | **Expected Result** | **Actual Result** | **Actual Result** |
| 1. | Check the field's name, type, size, screen layout etc. | It should be as per the program specs. | It should be as per the program specs. | PASS |
| 2. | Check field’s name, size, type and size for each database table against the database design. | Field's name, type, and size should be consistent throughout the database for a particular field. | Field's name, type, and size should be consistent throughout the database for a particular field. | PASS |
| 3. | Check the tab order of the controls | The tab order should be properly set so that the user can navigate through the keyboard also. | The tab order should be properly set so that the user can navigate through the keyboard also. | PASS |
| 4. | Check validation | The values entered should be properly entered | The values entered should be properly entered | PASS |

**Table 6.2**

**LIMITATION AND FUTURE ENHANCEMENT**

1. **Limitation and Future Enhancement**

* **Limitations**
* **Future Enhancement**

**7**

**INTRODUCTION**

**7.1 LIMITATION**

* Since it is an online project, shippers are confined by the use of good internet connection to have interaction with Transport Company and bid their goods.
* Fake registrations are one problem that cannot be restricted until and unless strict verification of shipper and transport company.
* The user must have their mail address.
* On the portal, the user can't bypass without validation and ReCaptcha.
* Without registration, any site viewer can't see any information on site, like ad details, user profile etc.
* Visitor only sees the home page and Feedback to see.
* If any user has complained to another user than admin has a right to block the user. And take action on the complaint.

**7.2 FUTURE SCOPE AND FURTHER ENHANCEMENT**

* This website involves almost all the features of online transportation for shipper’s goods.
* According to their suitability, the website creates a good platform to bid luggage as per prices fixed by the Transport Company.
* The future implementation will be online help for the shippers to chat with website administrators and the Shipper/Transport Company for giving feedback about Shipper/Transport Company.
* Its easy accessibility and usability scope of spreading this website is more because of its purpose.
* Directly search nearest Transport Company via Gio location.
* In future more number of authentication type to provide for registration and login.

**Conclusion**

**8 Conclusion & Reference**

* **Conclusion**
* **Reference**

**8**

**CONCLUSION & REFERENCE**

**8.1 CONCLUSION**

* HireTruck is developed such that it can extremely useful for both the Shipper and the Transport Company in order to maintain all the important information and retrieving the desired data anytime any were.
* Making an application for online registration for it greatly reduce the paper work of the administration and other details and helps about the companies provided helps the Shipper to prepare themselves for the Transfer their Luggage.
* E-Receipt can be generated as per the requirement.

**8.2 REFERENCE**

* <http://www.google.com>
* <http://www.Bootstrap.com>
* <http://www.W3School.com>
* <http://www.StackOverflow.com>
* <http://www.TutorialPoint.com>