## INTRODUCTION

The object of this project is to design and implement Aircraft Management System with user interface and administrator and staff interface using PHP technology. The system also allows us to add, edit and delete records or information on services offered by the company, information about the Engineers/Mechanics working for the company. All this information is stored in MySQL database which employ use of encrypted passwords to enhance the security of the data in it. For data storage and retrieval we use the file-handling facility of PHP Language. It enables us to add any number of records in our database. The retrieval of information from database is much faster when we Search a particular record in the database, because record is searched sequentially.

### About Project

The project named “Aircraft Management System ” is written in PHP, mainly because of it’s suitability for this type of application. Its user friendly nature and in-built documentation, complication, error detection, binding facilities and interaction with other system packages make it most powerful tool for system development. Moreover, PHP consists of all the technologies that help in creating and running robust, scalable systems.

Assistance is provided to the user at each and every step so that no problem is faced during using it. Further the details of every process and the user manuals attached in the report make it very easy to understand. Every possible care has been taken to make the software and the report clear, simple and error free which makes it so special, secure and one of its kind.

## Objective of the System

The objective of the Aircraft Service Management System  includes:

* provide better and secure information for the users of this system for better results for their maintenance in offering the services.
* To provide some amount of automation in aircraft service mangement.
* .To help aircraft service mangement in making their business more efficient.
* . An added attraction for their potential customers.
* . It will also show the attitute of the management that they are aware to the newly

introduced technology and ready to adopt them.

## SYSTEM SPECIFICATION

### **Hardware Requirements**:-

* Pentium-IV(Processor).
* 256 MB Ram
* 512 KB Cache Memory
* Hard disk 10 GB
* Microsoft Compatible 101 or more Key Boar

### Software Requirements:

* Operating System :           Windows
* Programming  language:   PHP
* Front-End:                          HTML, CSS, JAVASCRIPT
* Back-End:                           SQL
* Web Server:                       Apache Server

## FEASIBILITY STUDY

**A feasibility study is carried out to select the best system that meets performance requirements.**

Feasibility is the determination of whether or not a project is worth doing. The process followed in making this determination is called a feasibility study. This type of study determines if a project can and should be taken. Since the feasibility study may lead to the commitment of large resources, it becomes necessary that it should be conducted competently and that no fundamental errors of judgment are made. Depending on the results of the initial investigation, the survey is expanded to a more detailed feasibility study. Feasibility study is a test of system proposal according to its workability, impact on the organization, ability to meet user needs, and effective use of resources. The objective of the feasibility study is not to solve the problem but to acquire a sense of its scope . During the study, the problem definition is crystallized and aspects of the problem to be included in the system are determined.

## SYSTEM ANALYSIS

The system analysis phase is considered to be one of the most important phases in the system development life cycle. It is immensely important that the software developer make through study of the existing system. Thorough study of the system is made and need i.e. features that are critical to system success and users wants (i.e. features that would be good but not essential) are brought out. The study will enable the developer to know the intricacies of the existing system.

## **SYSTEM DESIGN**

Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization.  
Once the software requirements have been analyzed and specified the software design involves three technical activities - design, coding, implementation and testing that are required to build and verify the software.  
The design activities are of main importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system. Design is the only way to accurately translate the customer’s requirements into finished software or a system.  
Design is the place where quality is fostered in development. Software design is a process through which requirements are translated into a representation of software. Software design is conducted in two steps. Preliminary design is concerned with the transformation of requirements into data.

### USE CASE DIAGRAMS

Administrator

Client

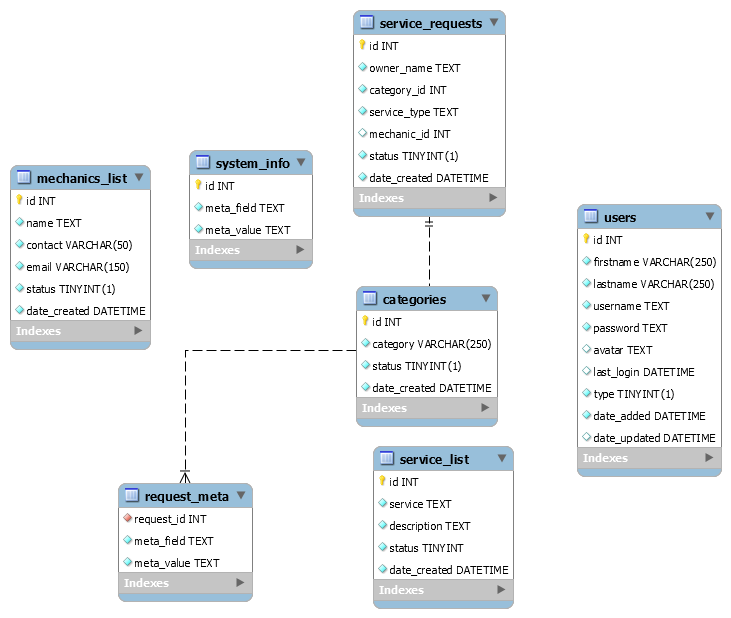
Staff

### E-R DIAGRAM

Peter proposed the Entity-Relationship model in 1976 [Chen76] as a mechanism to integrate network and relational database views. Simply said, the ER model is a conceptual data model that sees the real world in terms of entities and connections. The Entity-Relationship Diagram, which is used to visually represent data items, is a fundamental component of the model. Since Chen's article, the model has been expanded, and it is now widely used in database design. The ER model is useful for database designers because: It's a good fit for the relational model. The ER model's constructs are simply translated into relational tables. It's simple and straightforward to use, and it requires very little training. As a result, the database designer can use the model to convey the design to the end user. In addition, the model can be used as a design plan by the database developer to implement a data model in a specific database management software.

#### **CONNECTIVITY AND CARDINALITY**

The basic types of connectivity for relations are: one-to-one, one-to-many, and many-to-many. A *one-to-one*(1:1) relationship is when at most one instance of a entity A is associated with one instance of entity B. A *one-to-many*(1:N) relationships is when for one instance of entity A, there are zero, one, or many instances of entity B, but for one instance of entity B, there is only one instance of entity A. A *many-to-many*(M:N) relationship, sometimes called non-specific, is when for one instance of entity A, there are zero, one, or many instances of entity B and for one instance of entity B there are zero, one, or many instances of entity A. The connectivity of a relationship describes the mapping of associated. For our case we used the ER below:



## **SYSTEM TESTING AND IMPLEMENTATION**

Testing is a process of executing a program with the indent of finding an error. Testing is a crucial element of software quality assurance and presents ultimate review of specification, design and coding.  
System Testing is an important phase. Testing represents an interesting anomaly for the software. Thus a series of testing are performed for the proposed system before the system is ready for user acceptance testing.  
A good test case is one that has a high probability of finding an as undiscovered error. A successful test is one that uncovers an as undiscovered error.

Testing Objectives:  
 1. Testing is a process of executing a program with the intent of finding an error  
 2. A good test case is one that has a probability of finding an as yet undiscovered error  
 3. A successful test is one that uncovers an undiscovered error

**Testing Principles**

* All tests should be traceable to end user requirements
* Tests should be planned long before testing begins.
* Testing should begin on a small scale and progress towards testing in large.
* Exhaustive testing is not possible.
* To be most effective testing should be conducted by a independent third party

The primary objective for test case design is to derive a set of tests that has the highest livelihood for uncovering defects in software. To accomplish this objective two different categories of test case design techniques are used. They are  
 1. White box testing.  
 2. Black box testing.

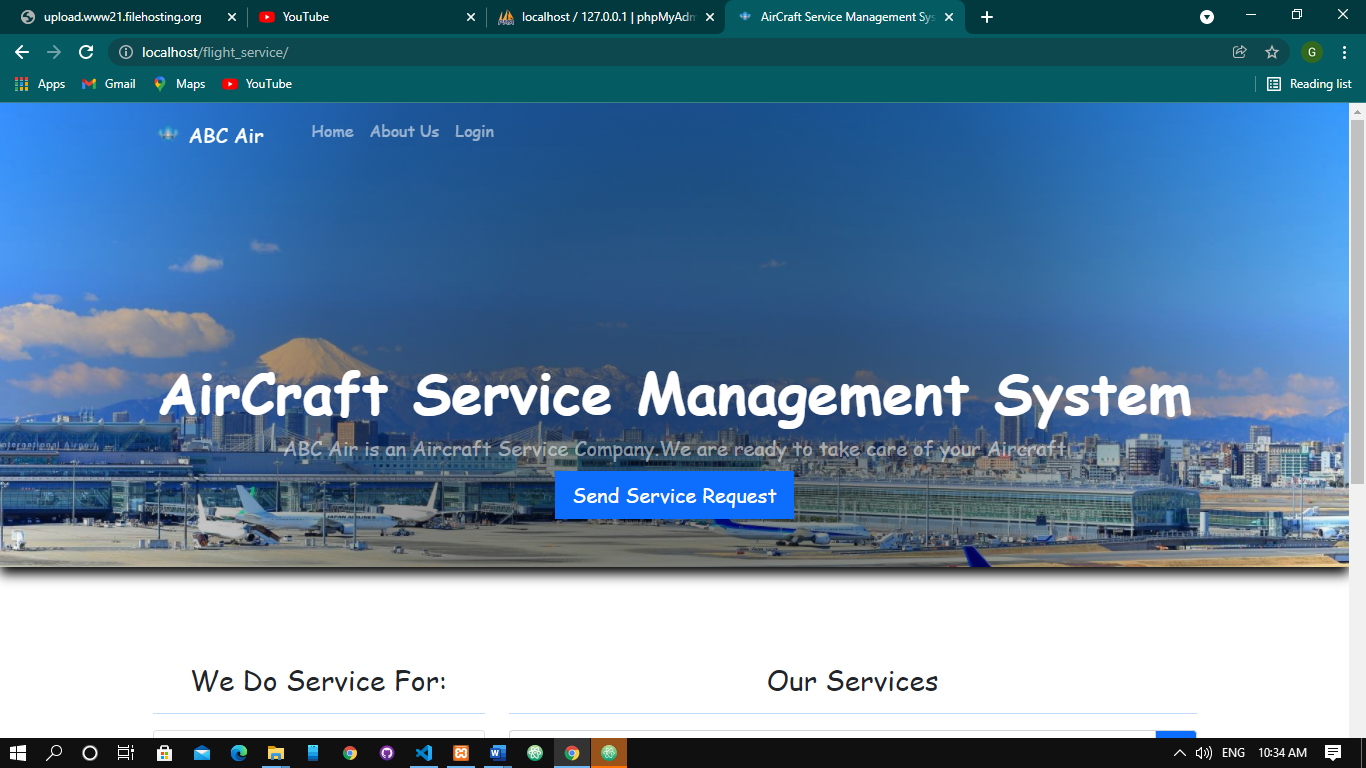
**Testing strategies**

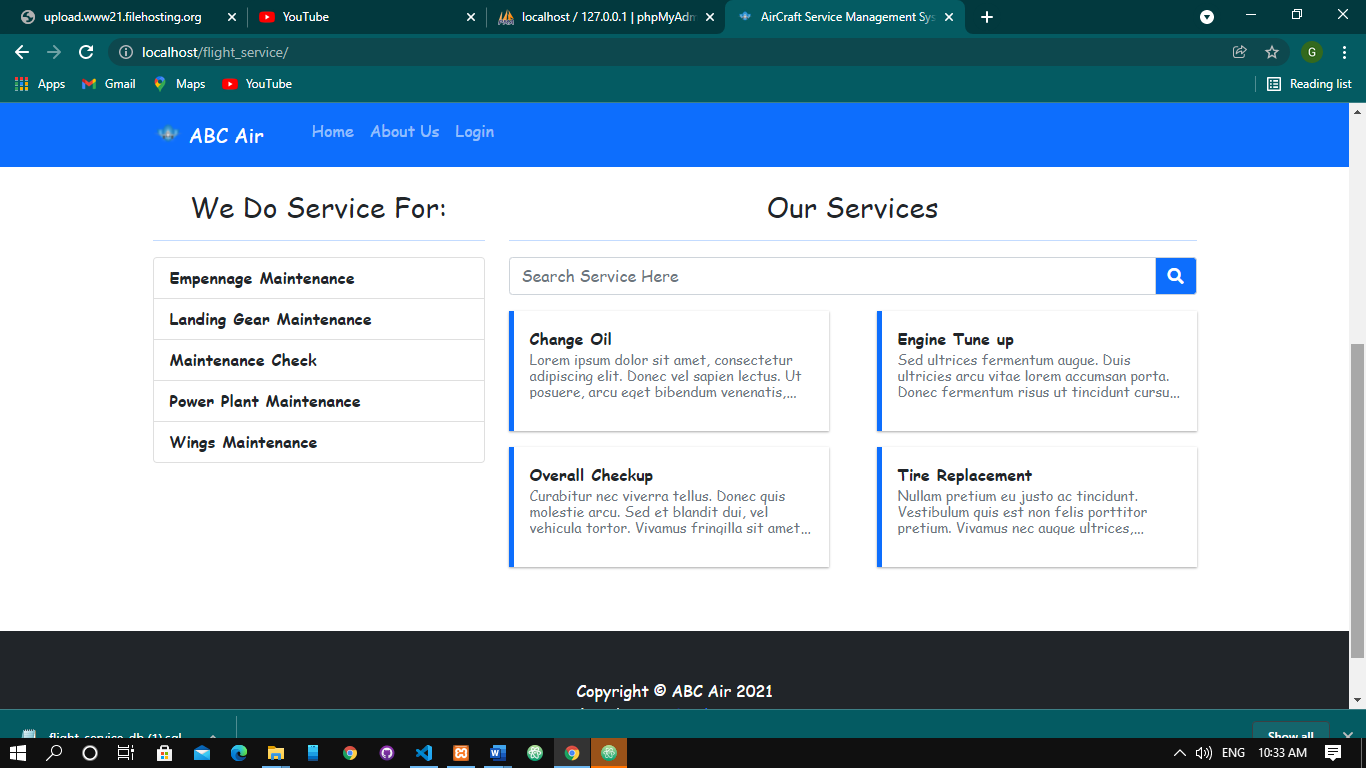
A strategy for software testing must accommodate low-level tests that are necessary to verify that all small source code segment has been correctly implemented as well as high-level tests that validate major system functions against customer requirements.  
**Testing fundamentals**  
Testing is a process of executing program with the intent of finding error. A good test case is one that has high probability of finding an undiscovered error. If testing is conducted successfully it uncovers the errors in the software.   
  
**Testing Information flow**  
Information flow for testing flows the pattern. Two class of input provided to test the process. The software configuration includes a software requirements specification, a design specification and source code.  
  
**Unit testing**  
Unit testing is essential for the verification of the code produced during the coding phase and hence the goal is to test the internal logic of the modules. Using the detailed design description as a guide, important paths are tested to uncover errors with in the boundary of the modules. These tests were carried out during the programming stage itself. All units of Vienna SQL were successfully tested.  
**Integration testing**   
Integration testing focuses on unit tested modules and build the program structure that is dictated by the design phase.  
**System testing**  
System testing tests the integration of each module in the system. It also tests to find discrepancies between the system and it’s original objective, current specification and system documentation. The primary concern is the compatibility of individual modules. Entire system is working properly or not will be tested here, and specified path ODBC connection will correct or not, and giving output or not are tested here these verifications and validations are done by giving input values to the system and by comparing with expected output. Top-down testing implementing here.  
**Acceptance Testing**  
This testing is done to verify the readiness of the system for the implementation. Acceptance testing begins when the system is complete. Its purpose is to provide the end user with the confidence that the system is ready for use. It involves planning and execution of functional tests, performance tests and stress tests in order to demonstrate that the implemented system satisfies its requirements.

## OUTPUT RESULTS

### Home page

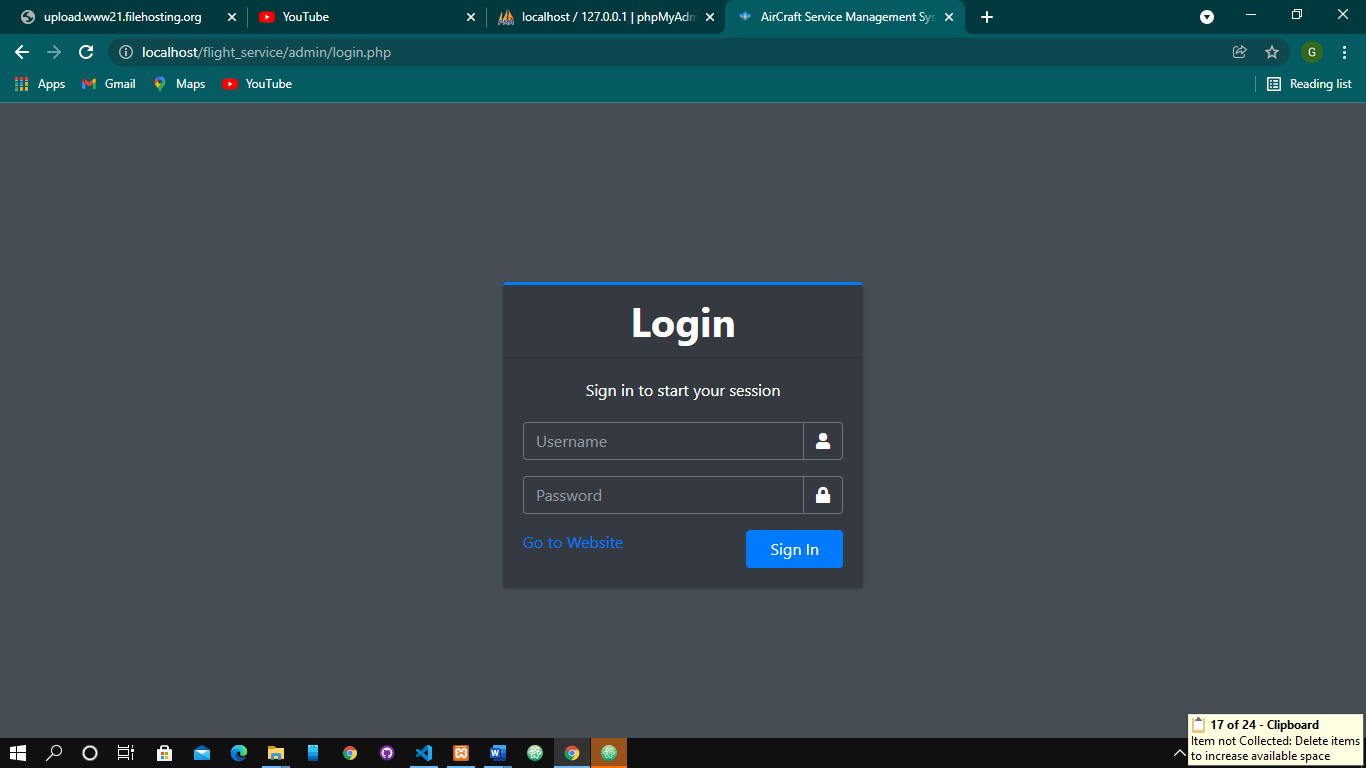
This is the user interface that first appears on running the website. Every user of the site is able to view the page whether signed in or not.





### Login page

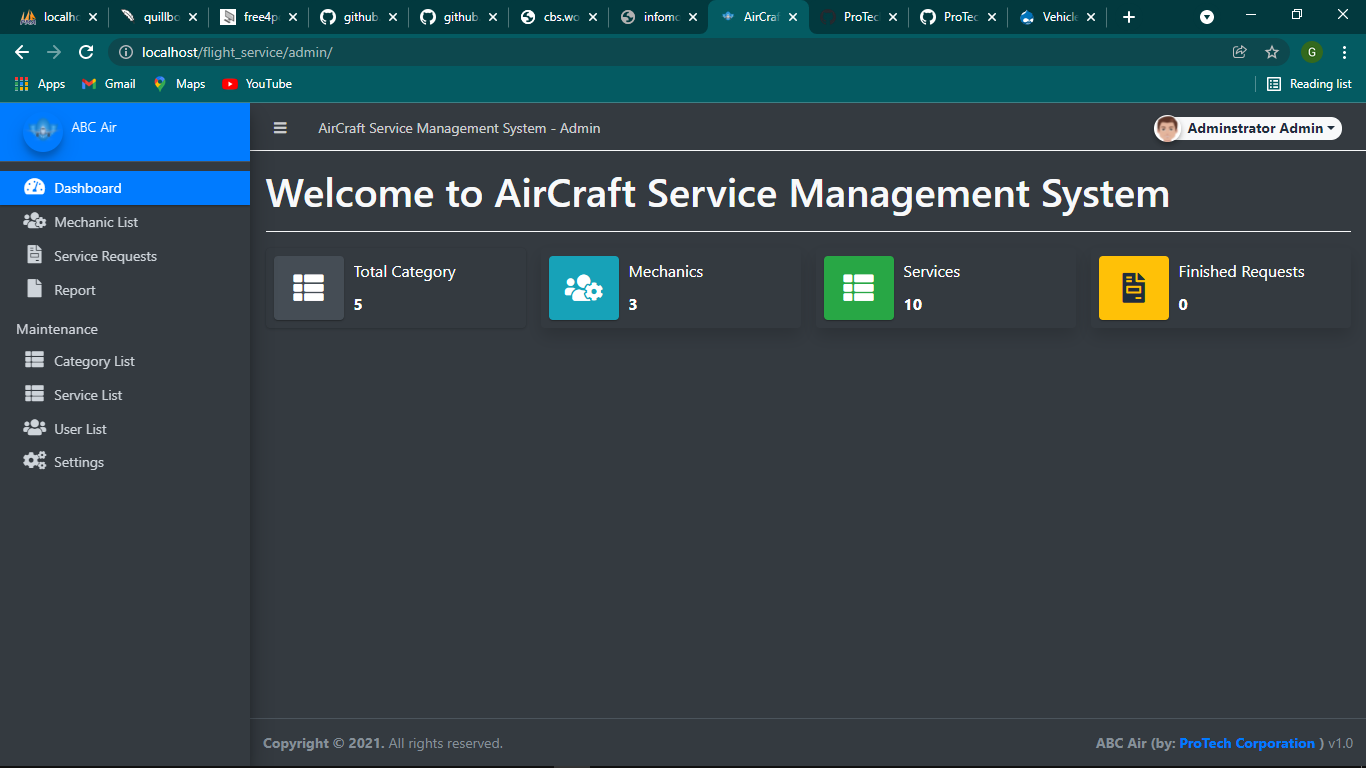
This page includes username and password field which is a requirement for logging in to the system. The password are encrypted for enhanced security.



### Administrator Dashboard

This is where everything on the website is managed from. Person in charge is the admin and manager of website. He or She is capable of

1. Adding new Engineers.
2. Adding new services to the system.
3. Assign Engineers/Mechanics tasks.
4. Generating Reports



## CONCLUSION

The system was designed in such a way that future modifications can be done easily. The following conclusions can be deduced from the development of the project.

* Automation of the entire system improves the efficiency
* It provides a friendly graphical user interface which proves to be much better.
* It gives appropriate access to the authorized users depending on their permissions.
* It effectively overcomes the delay in communications.
* Updating of information becomes so easier.
* System security, data security and reliability are the striking features.
* The System has adequate scope for modification in future if it is necessary.

## REFERANCES

### Books References

E. Balagurusamy , PHP Language ,  Tata McGraw-Hill Publishing house , Copyright 1998.

The ‘PHP’ Programming language by Kernighan & Ritchie, Prentice Hall Publishing house,  
Copyright 1988.

Teach yourself PHP by Yashwant Kanetkar, BPb Publishing house edition-6th.

Software engineering , K.K. Aggrawal, new age international publisher.

### Web References

www.**programming**.com/tutorial.html

www.plusplus.com/doc/tutorial/

www.coronadoenterprises.com/tutorials/php/index.html/