***MUTECH-ROBOTICS***

******

***Project Submitted in Partial fulfillment of the***

***Requirement for the Award of the Degree of***

Bachelor of Computer Application (Hons.)

***Semester VIII***

## Jan – May, 2015

**Under the guidance of: Submitted By:**

**Mr. Jugendra dongre Rahul Satal IC-2K11-60**

**Raunak Garud IC-2K11-65**

**Shivshankar Pindoriya IC-2K11-78**

**International Institute of Professional Studies**

**Devi Ahilya Vishwavidyalaya, Indore, M.P.**

**2015**

**International Institute of Professional Studies**

**Devi Ahilya Vishwavidyalaya, Indore, M.P.**

**DECLARATION**

I hereby declare that the project entitled “**Mutech-Robotics”** which is submitted by me for the partial fulfillment of the requirement for the award of Bachelor of Computer Applications (Hons.) (4 Years) VIII Semester to International Institute of Professional Studies, Devi Ahilya Vishwavidyalaya, Indore, comprises my own work and due acknowledgement has been made in text to all other material used.

Signature of Student:

Date:

Place: Indore

**International Institute of Professional Studies**

**Devi Ahilya Vishwavidyalaya, Indore, M.P.**

**CERTIFICATE FROM GUIDE**

It is to certify that project on “**Mutech-Robotics**”, submitted by Rahul Satal, Raunak Garud and Shivshankar Pindoriya to the International Institute of Professional Studies, DAVV, Indore has been completed under my supervision and the work is carried out and presented in a manner required for its acceptance in partial fulfillment for the award of the degree of “Bachelor of Computer Applications (Hons.)

(4 Years) VIII Semester”.

Project Guide

Signature:

Name:

Date:

**International Institute of Professional Studies**

**Devi Ahilya Vishwavidyalaya, Indore, M.P.**

**CERTIFICATE**

It is to certify that we have examined the project on “**Mutech-Robotics**”, submitted by Rahul Satal, Raunak Garud and Shiv Shankarto the International Institute of Professional Studies, DAVV, Indore and hereby accord our approval of it as a study carried out and presented in a manner required for its acceptance in partial fulfillment for the award of the degree of “Bachelor of Computer Applications (Hons.) (4 Years) VIII Semester”.

|  |  |
| --- | --- |
| **Internal Examiner:** | **External Examiner:** |
| Name: | Name: |
| Signature: | Signature: |
| Date: | Date: |

|  |  |
| --- | --- |
|  |  |
| **ACKNOWLEDGEMENT**  We would like to avail this opportunity to express our sincere thanks to all those who helped us in making this project. Even a most vivid collection of words, yield to express our heart fully thank towards one and all to have successfully assisted us in our expenditure of carrying out this project.  We wish to express our deep sense of gratitude to **our Director, Mr. Anand Sapre**, our Programme Incharge **Mr. Ramesh Thakur**, our project guide **Mr. Jugendra Dongre** and the whole faculty members of the department of Computer Science for encouraging and giving moral support, not only regarding this project but also throughout our studies at this institute. Also, to all our fellow classmates, friends and well wishers for their support and cooperation towards us.      **Rahul Satal**  **Raunak Garud**  **Shivshankar Pindoriya**      **TABLE OF CONTENTS**   |  |  |  | | --- | --- | --- | | **Chapter No.** | **Content** | **Page No.** | | **1** | **INTRODUCTION** | **7-12** | |  | 1.1 Problem Description  1.2 Project Aim  1.3 Project Description  1.4 Tools  1.5 Software Process Model | 8  8  9  10  11 | | **2** | **FEASIBILITY STUDY** | **13-16** | |  | 2.1 Existing System and Limitation  2.2 Proposed System Objective  2.3 Feasibility Study  2.3.1 Technical Feasibility Study  2.3.2 Economic Feasibility Study  2.3.3 Operational Feasibility Study | 14  15  16  16  17  17 | | **3** | **SYSTEM ANALYSIS** | **18-20** | |  | 3.1 System Analysis  3.2 Preliminary Investigation  3.3 Requirement Specification  3.3.1 Hardware Requirement  3.3.2 Software Requirement | 19  19  19  20  20 | | **4** | **SYSTEM DESIGN** | **21-34** | |  | 4.1 Entities Definition  4.2 Use Case Diagram  4.3 Use Case Description  4.4 Class Diagram  4.5 Data Flow Diagram  4.6 User Interface  4.7 Database Tables | 22  24  25  26  28  30 | | **5** | **SYSTEM TESTING** | **35-41** | |  | 5.1 Preparation of Test Data  5.2 White-Box Testing  5.3 Unit Testing  5.4 Integration Testing  5.5 Black-Box Testing  5.6 System Testing  5.7 Test Cases | 37  37  38  38  39  39  40 | | **6** | **CONCLUSION** | **42-44** | |  | 6.1 Findings  6.2 Limitations  6.3 Scope for Future Prospects | 43  43  44 | | **7** | **BIBLIOGRAPHY & REFERENCES** | **45-47** | |  | 7.1 Reference Books  7.2 Other Documentation and Resources | 46  46 | |  |
|  |  |
| CHAPTER 1  INTRODUCTION |  |
|  |  |

**Chapter 1**

**INTRODUCTION**

**1.1 Problem Description:**

The purpose of this system is to provide users to view various projects for various streams. Also the user can view a detailed information about the products created by the organization. The users also can get training information organized by the institute. The system also provides an interface to the user for any query via Contact Form .The goal of our project is to automate all the above tasks which were performed manually by the Organization.

**1.2 Project Aim:**

During the past several decades the manual work has been transformed into automation work. There are many factors that have influenced this transformation like technological advances, professionalism and general recognition of human beings as most important resources.

* The system is designed to provide easy interface to user to communicate with Organization about technical query or any other kind.
* Also this system helps to update the users by providing products and projects of various streams so that user can get a brief idea about the technical work.
* Also system provides user with training information organized by the institute.
* The project intends to introduce more user friendliness in various activities such as project, product, training, News, activities updation, insertion and deletion.
* The entire information has maintained in the database. Whomever wants to access can’t access, only admin can access the information.

The main objective of the entire activity to automate the process of activities of Institute like:

* Training activities
* Project related activities.
* Product development activities.
* Updating users with daily technical issues.
* Email facility (both user and admin).

**1.3 Project Description:**

This project is used by:

1. Administrator
2. User

User can view projects of various streams. The user can view products created by Organization. The user can view details about training scheduled by the Organization. Also the user can contact Organization by the interface Contact Form.

Admin can maintain the system by updating system modules like projects, products, training sessions, news, activities. Also admin can respond to the customers query and can solve their problems related to technical issues.

Mutech-Robotics has been designed to computerize following function:

* Awareness of technology to users.
* Also users can ask their technical queries to the experts using this system.
* Market ongoing projects details will be available to the users.
* Also users can brush up their skills by joining training sessions organized by institute.

**1.4 Tools:**

**1. Django (web Framework):**

Django  is a [free and open source](http://en.wikipedia.org/wiki/Free_and_open_source) [web application framework](http://en.wikipedia.org/wiki/Web_application_framework), written in [Python](http://en.wikipedia.org/wiki/Python_(programming_language)), which follows the [model–view–controller](http://en.wikipedia.org/wiki/Modelâ) (MVC)[architectural pattern](http://en.wikipedia.org/wiki/Architectural_pattern_(computer_science)). It is maintained by the [Django Software Foundation](http://en.wikipedia.org/wiki/Django_Software_Foundation).

Django's primary goal is to ease the creation of complex, database-driven websites. Django emphasizes reusability and "pluggability" of components, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings, files, and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.

Some well-known sites that use Django include Pinterest,Instagram,Mozilla, The Washington Times, Disqus, the Public Broadcasting Service, and Bitbucket.

**2. PhpMyAdmin:**

**PhpMyAdmin** is a free and open source tool written in PHP intended to handle the administration of MySQL with the use of a web browser. It can perform various tasks such as creating, modifying or deleting databases, tables, fields or rows; executing SQL statements; or managing users and permissions.

Features provided by the program include:

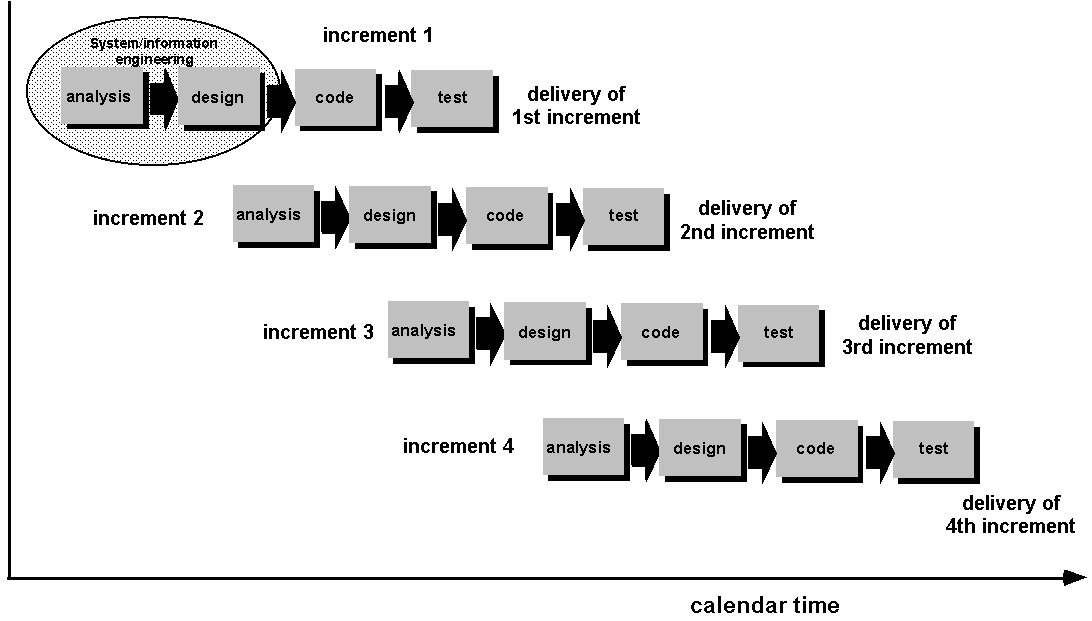
1. Web interface
2. MySQL database management
3. Import data from [CSV](http://en.wikipedia.org/wiki/Comma-separated_values) and [SQL](http://en.wikipedia.org/wiki/SQL)
4. Export data to various formats: [CSV](http://en.wikipedia.org/wiki/Comma-separated_values), [SQL](http://en.wikipedia.org/wiki/SQL), [XML](http://en.wikipedia.org/wiki/XML), [PDF](http://en.wikipedia.org/wiki/Portable_Document_Format) (via the [TCPDF](http://en.wikipedia.org/wiki/TCPDF) library), ISO/IEC 26300 - OpenDocument Text and Spreadsheet, Word, Excel, [LaTeX](http://en.wikipedia.org/wiki/LaTeX) and others
5. Administering multiple servers
6. Creating PDF graphics of the database layout
7. Creating complex queries using Query-by-Example (QBE)
8. Searching globally in a database or a subset of it
9. Transforming stored data into any format using a set of predefined functions, like displaying [BLOB](http://en.wikipedia.org/wiki/BLOB)-data as image or download-link
10. Live charts to monitor MySQL server activity like connections, processes, CPU/Memory usage, etc.
11. Working with different operating systems.

**1.5 Software Process Model:**

This project is a collaborative system and analyses the availability of resources, so the model chosen is Incremental model.

The Incremental model is a sequential software development process, on which progress is seen as flowing steadily downwards (like a waterfall) through the phases of requirements, design, implementation, verification and maintenance.

The incremental model proceeds from one phase to the next in a sequential manner. For example, one first completes requirement specification. When requirements are completed one proceeds to design and so on.



**Figure 1.5.1: Incremental Model**

The sequential phases in Waterfall model are:

* **Requirement Gathering and analysis:** All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification doc.
* **System Design:** The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.
* **Implementation:** With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.
* **Integration and Testing:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
* **Deployment of system:** Once the functional and non functional testing is done, the product is deployed in the customer environment or released into the market.
* **Maintenance:** There are some issues which come up in the client environment. To fix those issues patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

CHAPTER 2

FEASIBILITY

STUDY

**Chapter 2**

**FEASIBILITY STUDY**

System analysis is a detailed study of various operations performed by a system and their relationships within and outside of the system. Here the key question is-what all problems exist in the present system? What must be done to solve the problem? Analysis begins when a user or manager begins a study of the program using existing system. During analysis, data are collected on the various files, decision points and transactions handled by the present system. The commonly used tools in the system are data flow diagram, interviews, etc. Training, experience and common sense are required for collection of relevant information needed to develop the system.

The success of the system depends largely on how clearly the problem is defined thoroughly investigated and properly carried out through the choice of solution A good analysis model should provide not only the mechanisms of problem understanding, but also the framework of solution. Thus, it should be studied thoroughly by collecting data about the system. Then the proposed system should be analysed thoroughly in accordance of their needs.

System analysis can be categorised into five parts:

* System planning and initial investigation.
* Information gathering.
* Applying analysis tools for structured analysis.
* Feasibility study.
* Cost benefit analysis.

**2.1 Existing System and Limitation:**

This software product is eventually intended for the software developers. Product will be deployed to web site and all users of the product will access by use of the website. Website will be main user interface where users can operate all the provided functionality.

Website will only be the interface for the user data and the executionprovidedfunctionalities**.**

From admin point of view, admin will have functionality to insert, update or delete products , projects or technical Information according to its needs.

From user point of view, user will have functionality to contact with institute for any query related to technical issues or any suggestion for website. Also he can view products created by Organization or project info according to the streams.

**Limitations:**

* Internet is mandatorily required to access the system.
* It is highly dependent on database. In case database cause any problem it will effect the sytem to great extent.

**2.2 Proposed System Objectives:**

There are two personnels interacting with the system:

* Admin
* Users

**Admin:**

This module focuses on management of system. The admin has been provided with id and password for unique identification and security to the database. The options given to admin are:

* Manage Products.

.

* Manage Projects.
* Manage Trainings.
* Change Admin/Add Group.
* Manage News.
* Manage Activities.

**User:**

User can view all the components. User has a interface through which he can contact with the Organization ie Contact Form. The options given to user are:-

* View Products.
* View Projects.
* Contact Organization.
* View Trainings.
* View News.
* View Activities.

**2.3 Feasibility Study:**

A feasibility study could be used to test a proposal for new system, which could be used because:

* The current system may no longer carry its purpose.
* Technological advancement may have rendered the current system obsolete.
* The business is expanding, allowing it to cope with extra workload.
* Customers are complaining about the speed and quality of work the business provides.
* Competitors are now winning a big enough market share due to and effective integration of a computerized system.

A feasibility study should examine three main areas:

1. Technical feasibility.
2. Economical feasibility.
3. Operational feasibility.

**2.3.1. Technical Feasibility:**

This involves questions such as whether the technology needed for the system exists, how difficult it will be to build, and whether the firm has enough experience using that technology. The assessment is based on an outline design of system requirements in terms of input, processes, output, fields, programs and procedures. This can be qualified in terms of volumes of data, trends, frequency of updating, etc. in order to give an introduction to the technical system.

**2.3.2. Economical Feasibility:**

Economic justification is "bottom-line" consideration for most system. Economic justification includes a broad range of concern that includes cost benefit analysis, long term corporate income strategies, and cost of resources needed for development.

As the necessary hardware and software are available in the market at a low cost, the initial investment is the only cost incurred and does not need any further enhancements. Hence it is economically feasible. The system is feasible in all respects and hence it encourages taking up the system design.

**2.3.3. Operational Feasibility:**

Operational feasibility is mainly concerned with issues like, whether the system will be used if it developed and implemented? Whether there will be resistance from users that will affect the possible application benefits? The Answer to these questions is: Yes, as the system is developed for the convenience of the network administrator who is the main user of the system. In this software, all the above mentioned feasibility study has been conducted. It was found out that it is technically sound, there has to be no change in software or hardware existing already. It uses simple technical process. Economically, this software is much more cost effective. For operating this software, less training is required to the worker.

CHAPTER 3

SYSTEM

ANALYSIS

**Chapter 3**

**SYSTEM ANALYSIS**

**3.1 System analysis:**

System analysis by definition is a process of systematic investigation for the purpose of gathering data, interpreting the facts, diagnosing the problem and using this information to either build a completely new system or to recommend the improvements to the existing system.

The goal of [system analysis](http://en.wikipedia.org/wiki/Systems_analysis) is to determine where the problem is in an attempt to fix the system. This step involves [breaking down](http://en.wikipedia.org/wiki/Work_breakdown_structure) the system in different pieces to analyze the situation, analyzing project goals, breaking down what needs to be created and attempting to engage users so that definite requirements can be defined.

**3.2 Preliminary Investigation:**

The first step in system analysis is the identification of a need. This is a user's request to change, improve or enhance an existing system. Because there are likely to be a stream of such requests, standard procedures must be established to deal with them.

The preliminary investigation is one way of handling this. The objective of is to determine whether the request is valid and feasible before a recommendation is reached to do nothing, improve or modify the existing system or build a new one.

The user's request identifies the need for the change and authorizes the initial investigation. It may undergo several modifications before it becomes a written commitment. Once a request is approved, the following activities are carried out: background investigation, fact-finding and analysis and presentation of results- called project proposal.

**3.3 Requirement Specification:**

1. Tools : Django, Bootstrap (Web Development Framework).
2. Technology : MySql(PhpMyAdmin), HTML, CSS, Javascript.

**3.3.1 Hardware Requirements:**

# Processor : Pentium –IV

* + Speed : 1.1 GHz
  + RAM : 1 GB (min)
  + Hard Disk : 5 GB (at least excluding Data size)
  + Key Board : Standard Windows Keyboard
  + Mouse : Two or Three Button Mouse
  + Monitor : SVGA

**3.3.2 Software Requirements:**

* Operating System : Linux/Windows
* S/W Tools : Django, Bootstrap
* Database Connectivity : PhpMyAdmin

CHAPTER 4

SYSTEM DESIGN

**Chapter 4**

**SYSTEM DESIGN**

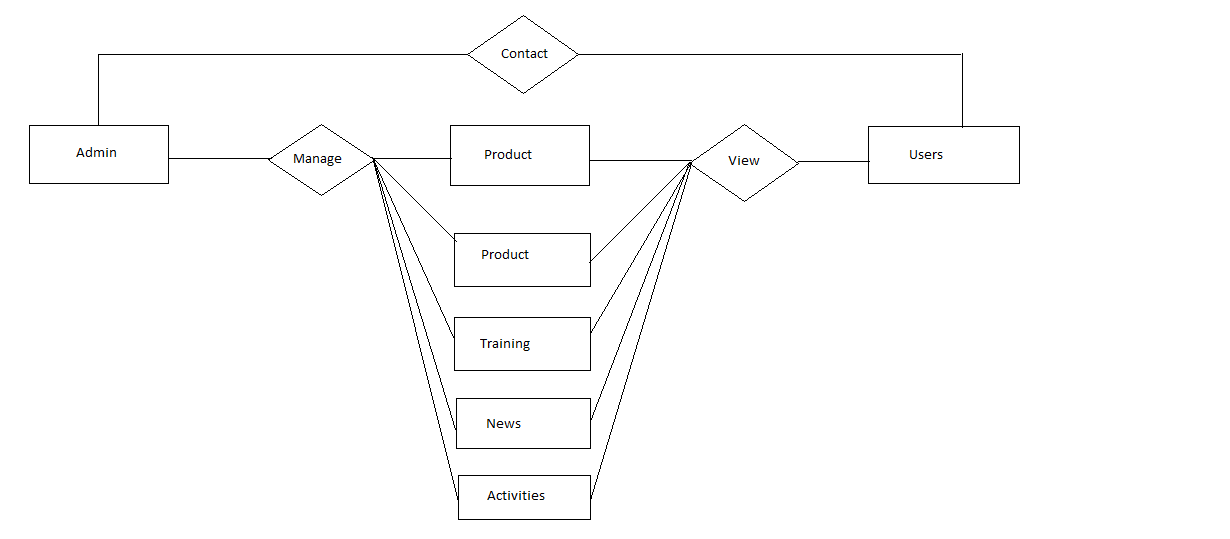
**4.1 Entities Definition:**

**1. Admin:**

* Manage Products.
* Manage Projects.
* Manage Trainings.
* Change Admin/Add Group.
* Manage News.
* Manage Activities.

**2. User:**

* View Product
* View Project.
* Contact Organization.
* View Training.
* View News.
* View Activities.

****

**Figure 4.1.1: ER Diagram**

**4.2 Use case Diagram-**

A use case diagram in the [Unified Modeling Language](http://en.wikipedia.org/wiki/Unified_Modeling_Language) (UML) is a type of behavioral diagram defined by and created from a [Use-case analysis](http://en.wikipedia.org/wiki/Use-case_analysis). Its purpose is to present a graphical overview of the functionality provided by a system in terms of [actors](http://en.wikipedia.org/wiki/Actor_(UML)), their goals (represented as [use cases](http://en.wikipedia.org/wiki/Use_case)), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. Use Case diagrams are formally included in two modeling languages defined by the [OMG](http://en.wikipedia.org/wiki/Object_Management_Group): the [Unified Modeling Language (UML)](http://en.wikipedia.org/wiki/Unified_Modeling_Language) and the Systems.

System

**Admin**

**Response Customer**

**Manage News/Activities**

**Manage Training**

**Change Admin**

**Manage Product**

**Manage Project**

**Fig 4.2.1: Use Case Diagram for Admin**

System

**User**

**View Training**

**Contact Organization**

**View Product**

**View Training**

**View Project**

**Fig 4.2.2: Use Case Diagram for User**

**4.3 Use Case Description:**

**1. Admin:**

* Manage Product
* Manage project.
* Manage Training.
* Change Admin.
* Manage News.
* Manage Activities.

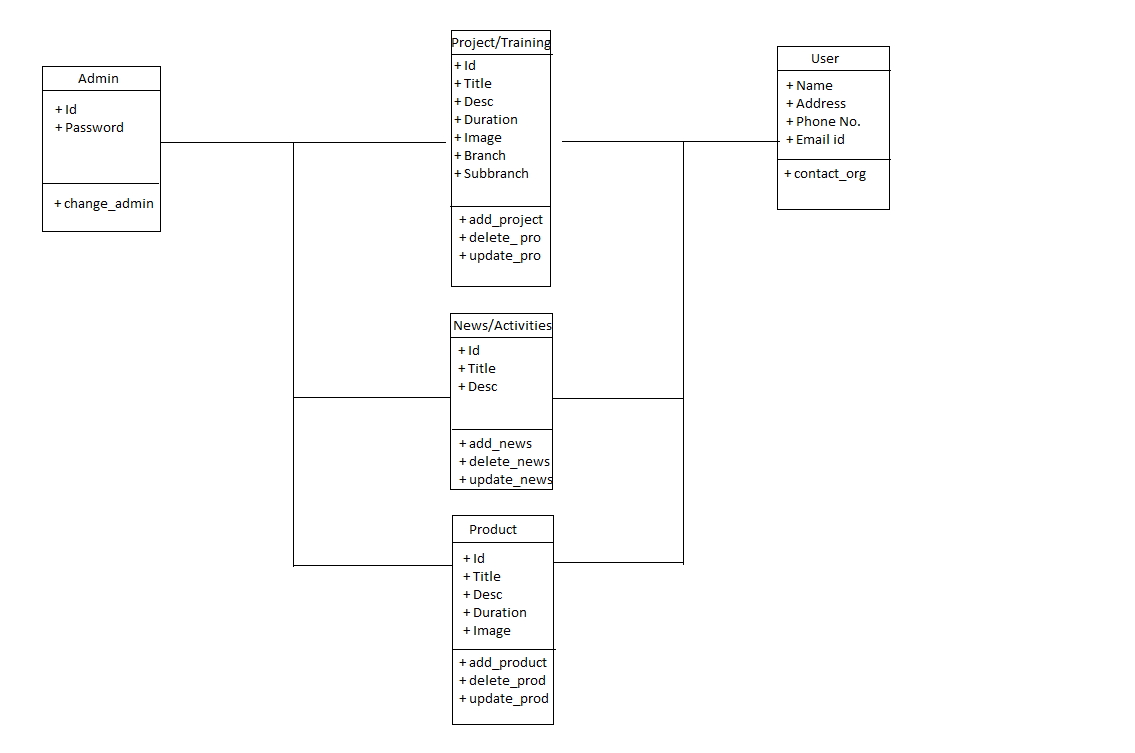
**2. User:**

* View Product
* View Project.
* Contact Organization.
* View Training.
* View News.
* View Activities.

**4.4 Class Diagram:**

A class diagram is a graph of classifier elements connected by their various static relationships. A “class” diagram may also contain interfaces, packages, relationships and even instances, such as objects and links. Perhaps a better name would be “static structural diagram” but “class diagram” is shorter and well established.

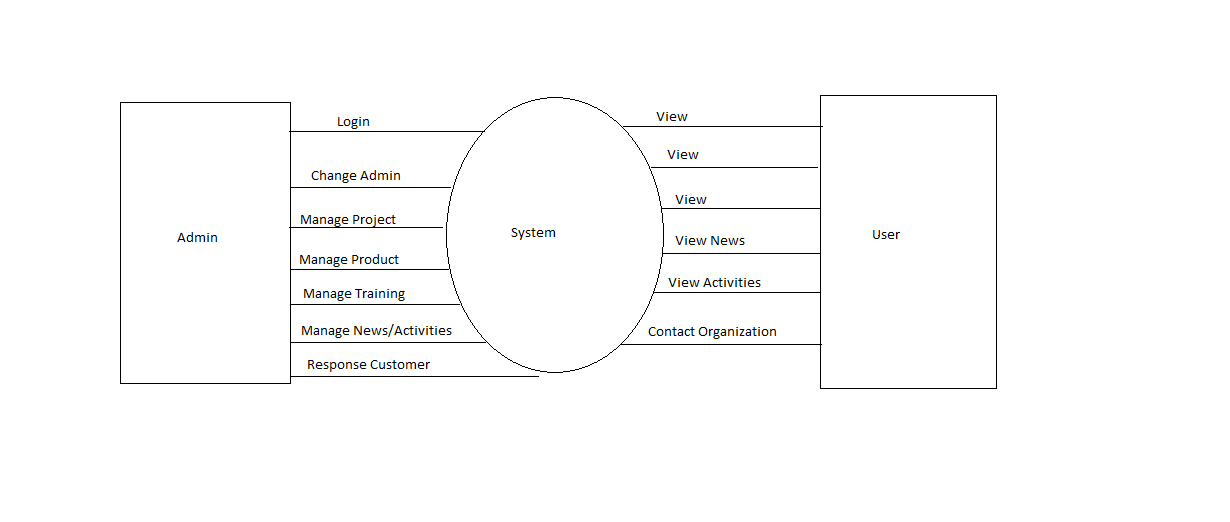
A class diagram is a graphic view of the static structural model. The individual class diagrams do not represents divisions in the underlying model. A class diagram is a collection of static model elements, such as classes, interfaces and their relationships, connected as a graph to each other and to their contents. Class diagrams may be organized into packages either with their underlying models or separate packages that build upon the underlying model packages.



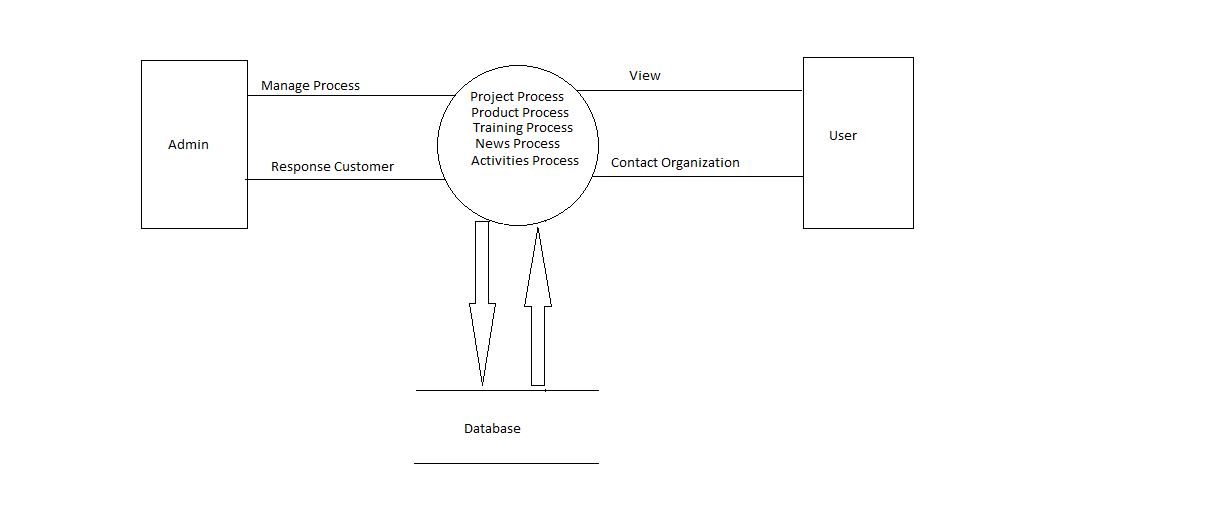
**Figure 4.4.1: Class Diagram**

**4.5 Data Flow Diagram:**

A Data Flow Diagram (DFD) is a graphical representation of the "flow" of data through an [information system](http://en.wikipedia.org/wiki/Information_system), modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the [visualization](http://en.wikipedia.org/wiki/Data_visualization) of [data processing](http://en.wikipedia.org/wiki/Data_processing) (structured design).

****

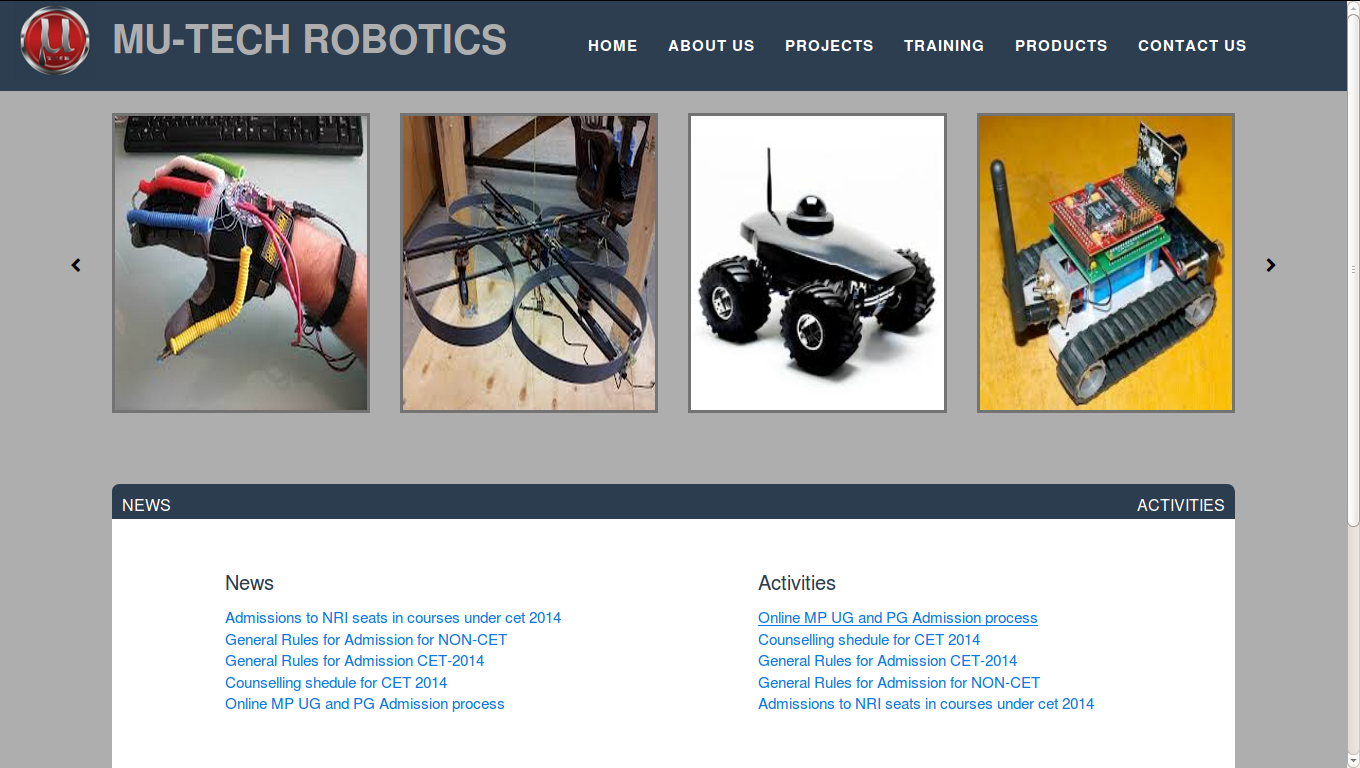
**Figure 4.5.1: Level 0 Context Model**

****

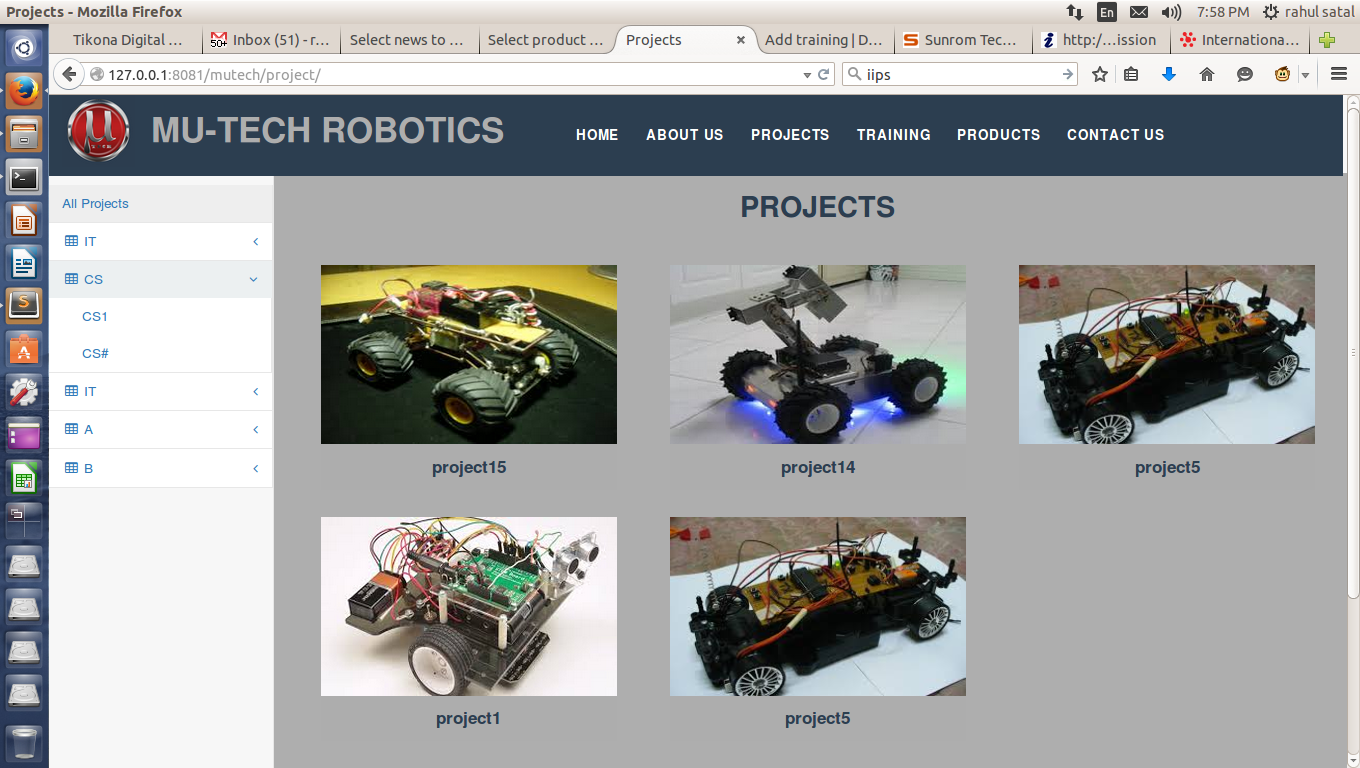
**Figure 4.5.2: Level 1 Data Flow Diagram**

**4.6 User Interface:**

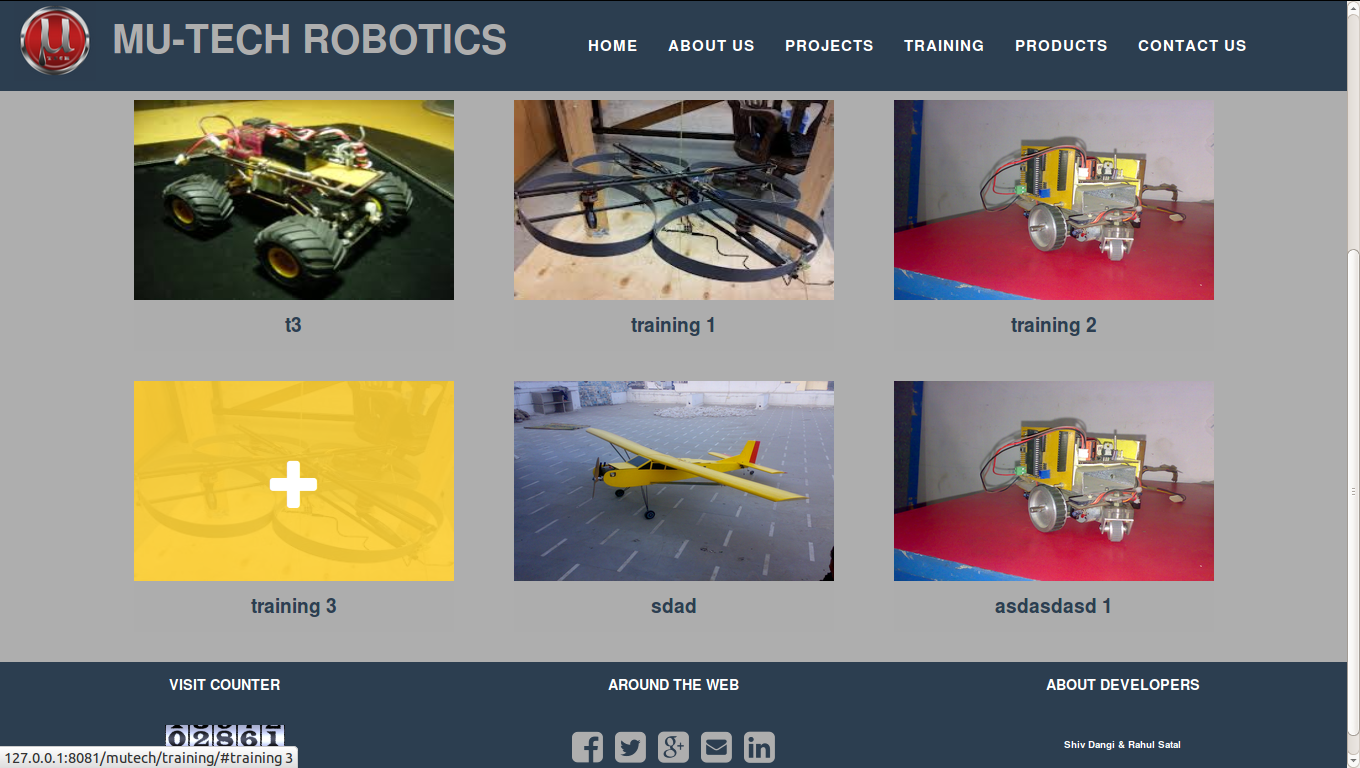
**4.6.1 Home Page**



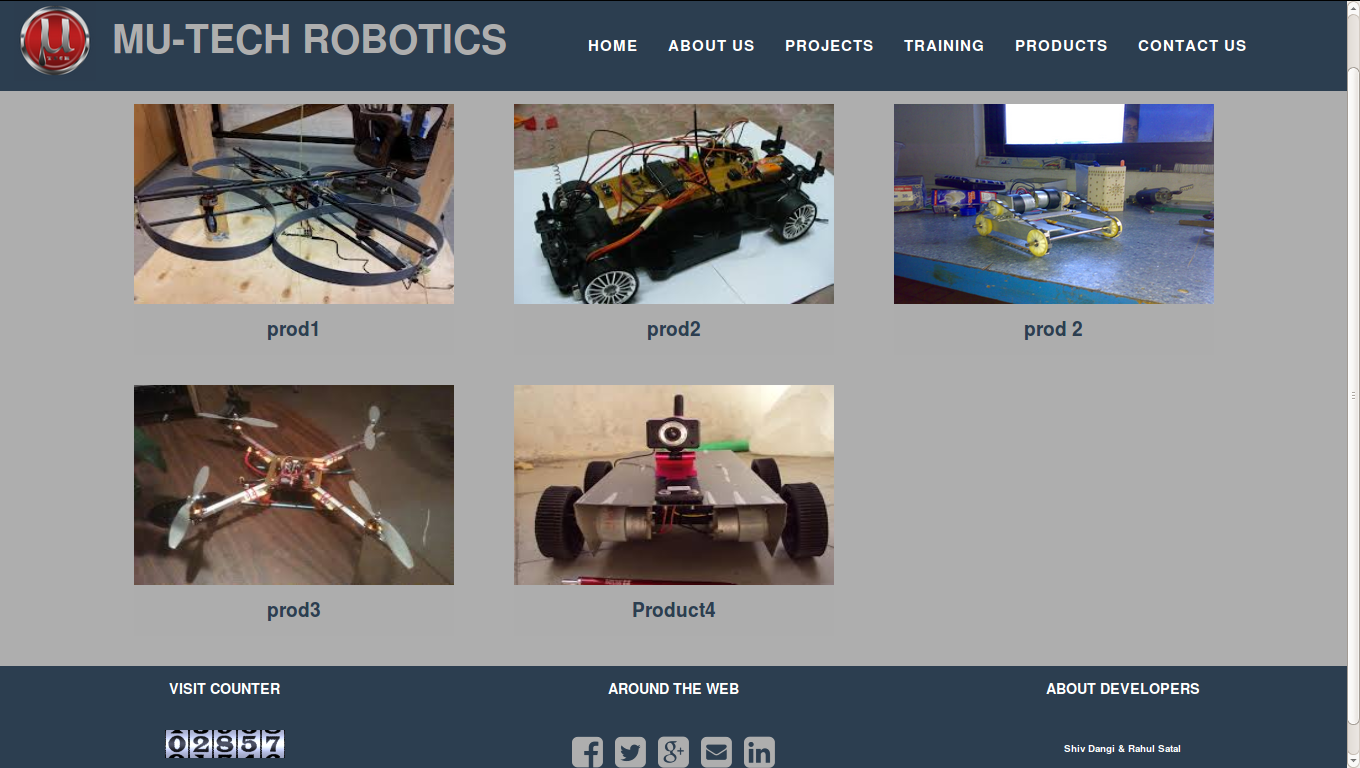
**4.6.2 Project Module**



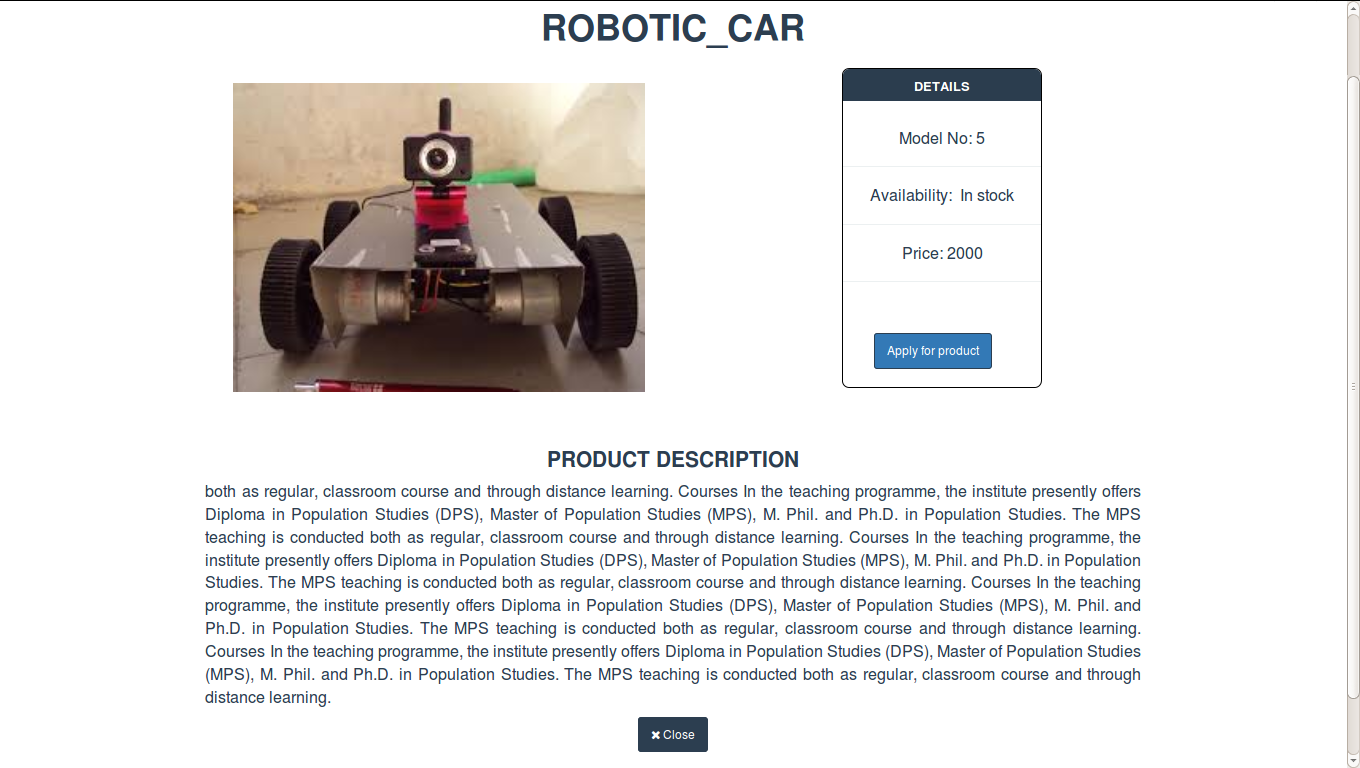
**4.6.3 Training Module**



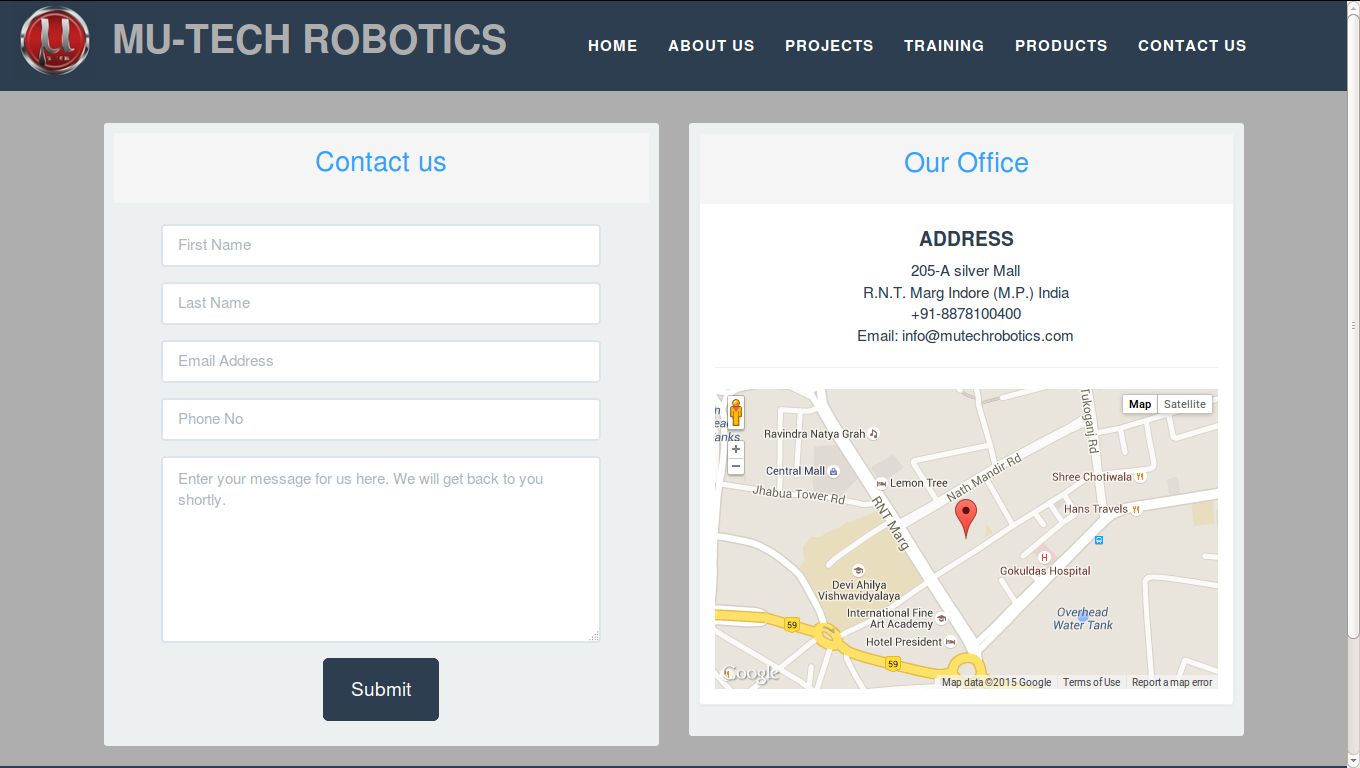
**4.6.4 Product Module**

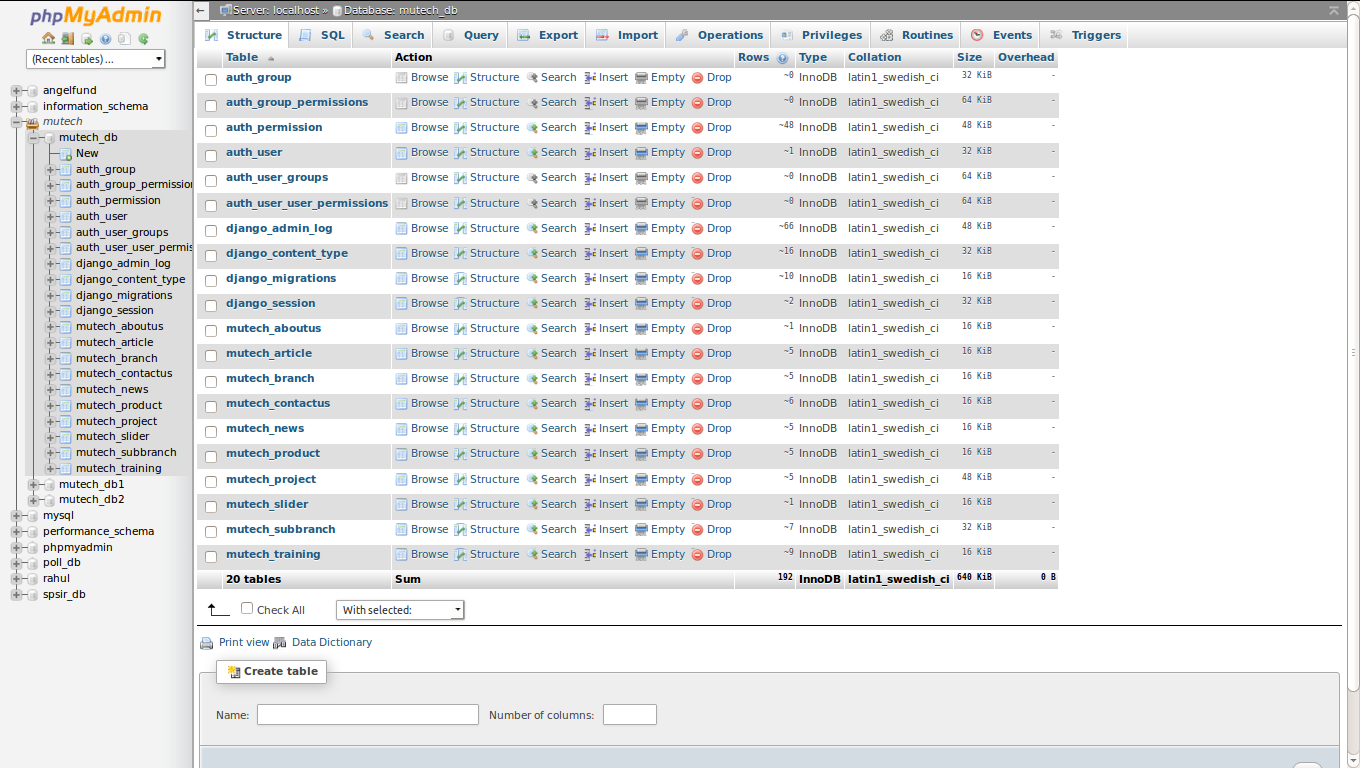


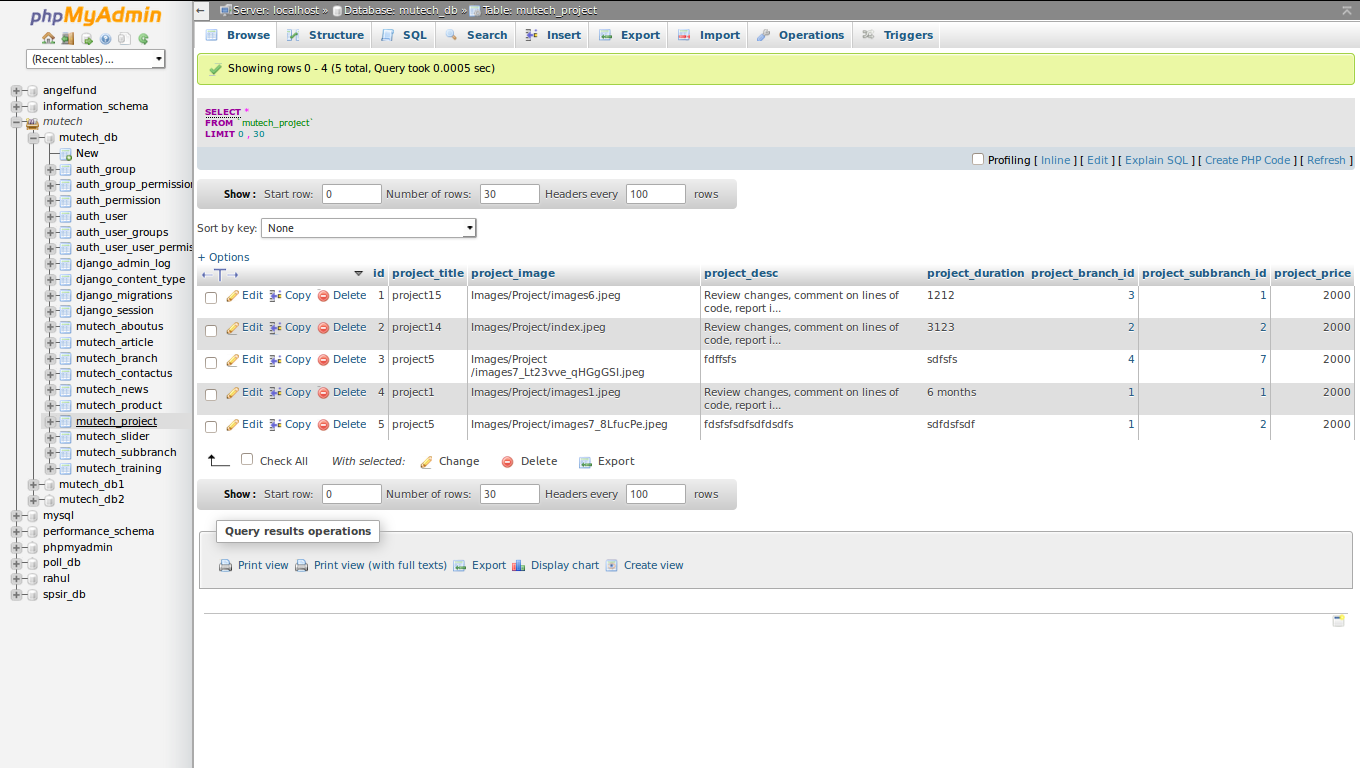
**4.6.5 Product Detail Module**

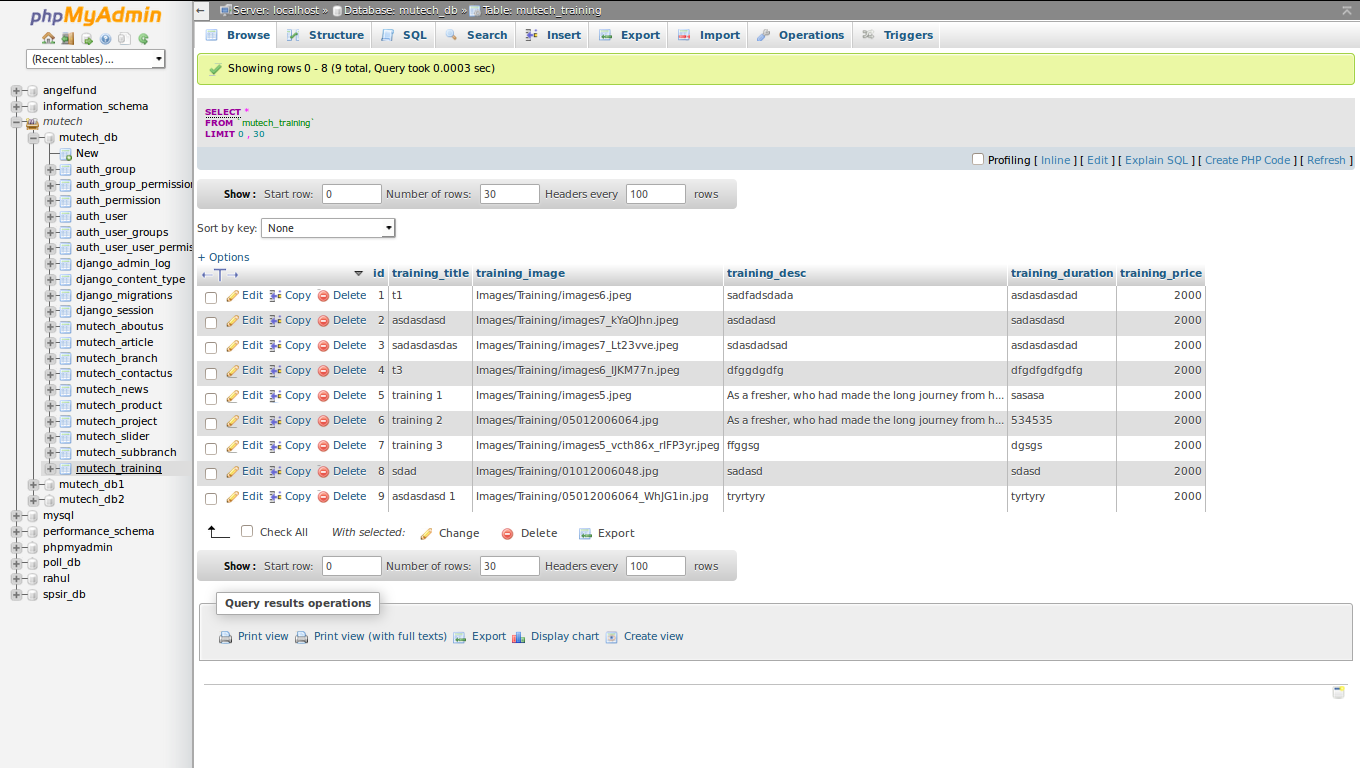


**4.6.6 Contactus Module**

**4.7 Database Tables:**

****





CHAPTER 5

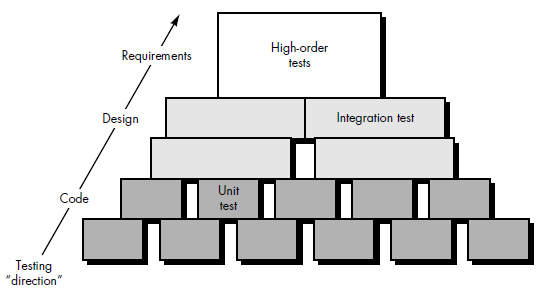
SYSTEM TESTING

**Chapter 5**

**SYSTEM TESTING**

The development of software systems involves a series of production activities where opportunities for injection of human fallibilities are enormous. Errors may begin to occur at the very inception of the process where the objectives may be erroneously or imperfectly specified, as well as [in] later design and development stages. Because of human inability to perform and communicate with perfection, software development is accompanied by a quality assurance activity.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design, and code generation. The increasing visibility of software as a system element and the attendant "costs" associated with a software failure are motivating forces for well-planned, thorough testing. It is not unusual for a software development organization to expend between 30 and 40 percent of total project effort on testing. In the extreme, testing of human-rated software (e.g., flight control, nuclear reactor monitoring) can cost three to five times as much as all other software engineering steps combined.



**Software testing steps**

**5.1 Preparation of test data:**

Is the menu bar displayed in the appropriate contested some system related features included either in

menu or tools? Do pull down menu operation and tool bars work properly? Is all menu function and

pull down sub function properly listed? Is it possible to invoke each menu function using a logical assumption that if all parts of the system are correct, goal will be successfully achieved? In the adequate testing or on-testing will leads to errors that may appear few months later.

This creates two problems:

1. Time delay between the cause and appearance of the problem.
2. The effect of the system errors on files and records within the system.

The purpose of the system testing is to consider all the likely variations to which it will be suggested and push the system limits.

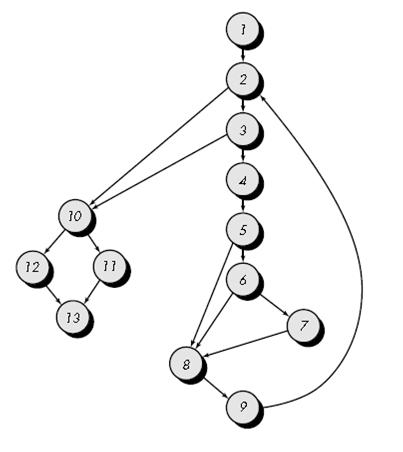
The testing process focuses on the logical intervals of the software ensuring that all statements have been tested and on functional interval is conducting tests to uncover errors and ensure that defined input will produce actual results that agree with the required results. Program level testing, modules level testing integrated and carried out.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation. Once source code has been generated, software must be tested to uncover and correct as many errors as possible before delivery to the customer. These techniques provide systematic guidance for designing tests that:

* Exercise the internal logic of software components.
* Exercise the input and output domains of the program to uncover errors in program function, behavior and performance.

**5.2 White box testing:**

White-box testing is when the tester has access to the internal data structures and algorithms including the code that implements these. 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 have been tested.

****

**Figure 5.2.1: White box testing**

**5.3 Unit Testing:**

Considering the process from a procedural point of view, testing within the context of software engineering is actually a series of four steps that are implemented sequentially. Initially, tests focus on each component individually, ensuring that it functions properly as a unit. Hence, the name unit testing.

Unit testingbegins at the vortex of the spiral and concentrates on each unit (i.e., component) of the software as implemented in source code. Unit testing makes heavy use of white-box testing techniques, exercising specific paths in a module's control structure to ensure complete coverage and maximum error detection.

**5.4 Integration Testing:**

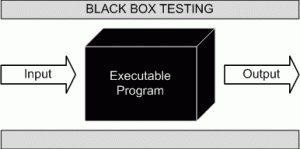
Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design.

There is often a tendency to attempt non incremental integration; that is, to construct the program using a "big bang" approach. All components are combined in advance. The entire program is tested as a whole. And chaos usually results! A set of errors is encountered. Correction is difficult because isolation of causes is complicated by the vast expanse of the entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop.

Incremental integration is the antithesis of the big bang approach. The program is constructed and tested in small increments, where errors are easier to isolate and correct; interfaces are more likely to be tested completely; and a systematic test approach may be applied

**5.5 Black box Testing:**

Black-box testing treats the software as a "black box"—without any knowledge of internal implementation. Black-box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, fuzz testing, model-based testing, exploratory testing and specification-based testing.



**Figure 5.5.1: Black Box Testing**

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 do not. 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. As a result, there are situations when (1) a tester writes many test cases to check something that could have been 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.

**5.6 System Testing:**

In system testing, the software and other system elements are tested as a whole. To test computer software, we spiral out along streamlines that broaden the scope of testing with each turn. System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work to verify that system elements have been properly integrated and perform allocated functions.

**5.7 Test Cases:**

In order to fully test that all the requirements of an application are met, there must be at least two test cases for each requirement: one positive test and one negative test. If a requirement has sub-requirements, each sub-requirement must have at least two test cases. Keeping track of the link between the requirement and the test is frequently done using a traceability matrix. Written test cases should include a description of the functionality to be tested, and the preparation required to ensure that the test can be conducted.

A formal written test-case is characterized by a known input and by an expected output, which is worked out before the test is executed. The known input should test a precondition and the expected output should test a post condition.

**Formal Test Cases:**

In order to fully test that all the requirements of an application are met, there must be at least two test cases for each requirement: one positive test and one negative test. If a requirement has sub-requirements, each sub-requirement must have at least two test cases. Keeping track of the link between the requirement and the test is frequently done using a traceability matrix. Written test cases should include a description of the functionality to be tested, and the preparation required to ensure that the test can be conducted.

A formal written test-case is characterized by a known input and by an expected output, which is worked out before the test is executed. The known input should test a precondition and the expected output should test a post condition.

**Informal Test Cases:**

For applications or systems without formal requirements, test cases can be written based on the accepted normal operation of programs of a similar class. In some schools of testing, test cases are not written at all but the activities and results are reported after the tests have been run.

In scenario testing, hypothetical stories are used to help the tester think through a complex problem or system. These scenarios are usually not written down in any detail. They can be as simple as a diagram for a testing environment or they could be a description written in prose. The ideal scenario test is a story that is motivating, credible, complex, and easy to evaluate. They are usually different from test cases in that test cases are single steps while scenarios cover a number of steps of the key.

**Sample Test Data and Results:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No** | **TEST CASES** | **INPUT** | **EXPECTED OUTPUT** | **ACTUAL OUTPUT** |
| 1. | Enter admin and password. | Invalid username but valid password. | Error Message | Error Message (Invalid username/  password) |
| 2. | Enter admin and password. | Valid username but invalid password. | Error Message | Error Message (Invalid username/  password) |
| 3. | Enter admin and password. | Valid username and valid password | Processed to next Module | Login Successful. |

CHAPTER 6

CONCLUSION

**Chapter 6**

**CONCLUSION**

It has been a matter of immense pleasure, honor and challenge to have this opportunity to take up this project and complete it successfully. This project has given us an ample opportunity to design, code, test and implement an application. While developing this project we have learnt a lot about shopping software. We have also learnt how to make it user friendly. This has helped us in putting into practice various Software Engineering principles and Database Management concepts like maintaining integrity and consistency of data.

During the development process, we studied carefully and understood each criteria for making a software more demanding. We also realized the importance of maintaining a minimal margin for error.

We thank our guide for his invaluable contribution in guiding me throughout the project. We also thank our parents and friends who have supported and motivated us to complete this project successfully.

**6.1 Findings:**

We have developed a Web application for an Institute(Mutech Informatics) which can be accessed anywhere with ease via any computer device with internet facility, its other facilities include:

* Making user aware about technological development across the world.
* Easy access to the information about projects, products and other technical issues.
* Direct interaction between user and the organization.

**6.2 Limitations:**

* Internet is mandatoriy required to access the system.
* It is highly dependent on database. In case database cause any problem it will effect the system to great extent.

**6.3 Scope for Future Prospects:**

This software is flexible in nature. There are many places in this software that can be enhanced in future. It has terminal points where more features can be added at that place. It can be made server based project in future, also with some additional facilities where users can access it.

There are some areas of improvement which includes:

* User can maintain his own profile.
* User will be able to order for the products online and do payments.
* Also additional features can be added for communication of user with organization via video Conferencing.

CHAPTER 7

BIBLIOGRAPHY AND REFERENCES

**Chapter 7**

**BIBLIOGRAPHY AND REFERENCES**

**7.1 Reference Books:**

1. Software Engineering by Roger S. Pressman.
2. Information Practices with Sumita Arora, Dhanpat Rai & Co., Fifth Edition.

3. PL/SQL Query Language by Ivan Bayross, BPB Publications, Fourth Edition.

4. System Analysis and Design by Elias M. Awad, Galgotia Publications Pvt. Ltd., Second Edition.

5. Database Management System, S. Sudarshan and Henry F. Korth.

**7.2 Other Documents and Resources:**

1. [www.wikipedia.org](http://www.wikipedia.org)

2. [www.google.co.in](http://www.google.co.in)

3. [www.stackoverflow.com](http://www.stackoverflow.com)

4. www.django.org

5. www.javatpoint.com

6. www.techwiz.com

7. [www.sunrom.com](http://www.sunrom.com)

8 .www.getbootstrap.com.