CHAPTER 1

INTRODUCTION

Event management is a process of organizing a professional and focused event, for a particular target audience. It involves visualizing concepts, planning, budgeting, organizing and executing events such as wedding, musical concerts, corporate seminars, exhibitions, birthday celebrations, theme parties, etc. Event Management is a multi-million dollar industry, growing rapidly, with events hosted regularly. Surprisingly, there is no formalized research conducted to access the growth of this industry. The industry includes fields such as the MICE (Meetings, Incentives and Events), exhibitions, conferences and seminars as well as live music and sporting events. On the profession side, event management is a glamorous and exciting profession that demands a lot of hard work and dynamism. The logistics side of the industry is paid less than the sales/sponsorship side, though some may say that these are two different industries.

Event management is the application of project management to the creation and development of large scale events. The process of planning and coordinating the event is usually referred to as event planning and which can include budgeting, scheduling, site selection, acquiring necessary permits, coordinating transportation and parking, arranging for speakers or entertainers, arranging decor, event security, catering, coordinating with third party vendors, and emergency plans. The events industry now includes events of all sizes from the Olympics down to business breakfast meetings. Many industries, charitable organizations, and interest groups hold events in order to market themselves, build business relationships, raise money, or celebrate achievement. An event refers to a social gathering or activity, such as a festival,(for example a musical festival), a ceremony(for example a marriage) and a party(for example a birthday party).There are mainly 3 types of event management:

- Corporate Event Management
- Product Launch Event Management
- Special Event Management

1.1 EVENT INCLUDED

Events which are included in this project are basics that are conducted at the colleges among the professors , assistant professors of same and other college professors and assistant professors.

Those included events are,

- ✓ Courses
- ✓ Journals
- ✓ Patterns
- ✓ Funds
- ✓ Others

1.2 SCOPE OF THE PROJECT

The objective of this application is to develop a system that effectively manages all the data related to the various events that take place in an organization. The purpose is to maintain a centralized database of all event related information. The goal is to support various functions and processes necessary to manage the data efficiently.

1.3 EVENT MANAGEMENT PROCESS

There are two stages of event management process namely, Event planning and Event control.

Event Planning/Control:

There no options for planning the events in this project.

Process

- ✓ Sign up for creating the account.
- ✓ Sign in process contains the information such as names , password , SIG ,unique password (for authorized entry) .
- ✓ After creating the account successfully you can login to your account.
- ✓ After entering into website ,you are landing on the view page where you can see the events you added.
- ✓ For the first time , you have to register by navigate to register page.
- ✓ You can navigate to a register page by a navigation bar .
- ✓ Slide out the navigation bar by sliding right from the left edge or make it appear by
- ✓ pressing the hamburger menu on left corner on top.
- ✓ In Register page, select the event which is popes out on the first input field.
- ✓ According to the event selected , it changes the contents followed by.
- ✓ Every event has their corresponding inputs.

- ✓ Fill all the fields and submit it to database.
- ✓ After those processes you can view your event in the view page.

1.4 EXISTING SYSTEM

This existing system is not providing secure registration and profile management of all the users properly. This system is not providing on-line help. This system doesn't provide tracking of users activities and their progress. This manual system gives us very less security for saving data and some data may be lost due to mismanagement. This system is not providing event management through internet. This system is not providing proper events information. The system is giving manual information through the event management executer.

1.5 FEASIBILITY STUDY

A feasibility study is a high-level capsule version of the entire System analysis and Design Process. The study begins by classifying the problem definition. Feasibility is to determine if it's worth doing. Once an acceptance problem definition has been generated, the analyst develops a logical model of the system. A search for alternatives is analyzed carefully. There are 3 parts in feasibility study.

1.5.1 Operational Feasibility

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.The operational feasibility assessment focuses on the degree to

which the proposed development projects fits in with the existing business environment and objectives with regard to development schedule, delivery date, corporate culture and existing business processes.To ensure success, desired operational outcomes must be imparted during design and development. These include such design-dependent parameters as reliability, maintainability, supportability, usability, producibility, disposability, sustainability, affordability and others. These parameters are required to be considered at the early stages of design if desired operational behaviours are to be realised. A system design and development requires appropriate and timely application of engineering and management efforts to meet the previously mentioned parameters. A system may serve its intended purpose most effectively when its technical and operating characteristics are engineered into the design. Therefore, operational feasibility is a critical aspect of systems engineering that needs to be an integral part of the early design phases.

1.5.2 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 outline design of system requirements in terms of input, processes, output, fields, programs and procedures. This can be qualified in terms of volume of data, trends, frequency of updating inorder to give an introduction to the technical system. The application is the fact that it has been developed on windows XP platform and a high configuration of 1GB RAM on Intel Pentium Dual core processor. This is technically feasible .The technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. It is an evaluation of the hardware and software and how it meets the need of the proposed system.

1.5.3 Economical Feasibility

Establishing the cost-effectiveness of the proposed system i.e. if the benefits do not outweigh the costs then it is not worth going ahead. In the fast paced world today there is a great need of online social networking facilities. Thus the benefits of this project in the current scenario make it economically feasible. The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. It includes quantification and identification of all the benefits expected. This assessment typically involves a cost/benefits analysis.

CHAPTER 2

REQUIREMENT ANALYSIS

2.1 HARDWARE REQUIREMENTS

Minimum requirements

• Hardware	: Processor Intel dual core and above
• Clock speed	: 3.0 GHz
• RAM size	: 512 MB
• Hard Disk capacity	: 400 GB
• Monitor type	: 15 inch color monitor

2.2 SOFTWARE REQUIREMENTS

Minimum requirements

• Operating System	: Windows XP.
• Internet connection	: Existing telephone lines, Data card.
• Browser	:Microsoft Edge
• Performance	: The turn-around time of the project is medium.

Possibilities

Operating Systems

- Windows XP
- Windows 7
- Windows 8
- Windows 9
- Windows 10

Browsers

- IE 7
- IE 8
- Microsoft Edge
- Google chrome
- Opera
- Mozilla Firefox

CHAPTER 3

PROPOSED SYSTEM

Event Management System is an Online event management software project that serves the functionality of an event manager. The system allows only registered users to login and new users are allowed to register on the application. This project is a web application that provides most of the basic functionality required for an event. It allows the user to select from a list of event types. Once the user enters an event type the system then allows the user to select the date of event and the event equipment's. All this data is logged in the database and the user is setting up his username and password while registering . The data is then sent to the database and his contact data stored in the database.

3.1 ADVANTAGES

- The system is useful as it has shown data of same group.
- It maintain privacy by hiding informations of other group members.
- The user gets all the resources at a single place instead of wandering around for these.
- This system is effective and saves time of the users.

CHAPTER 4

DESIGN AND ARCHITECTURES

4.1 DESIGN

Design is the first step in 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 requirement have been analyzed and specified the software design involves three technical activities-Design, Coding,Implementation, Testing that are required to build and verify the software.The design activities are of main importance in this phase,because in this activities decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decision has the final bearing upon reliability and maintainability of the a system.Design is only way to accurately transfer the customers requirements into finished software or system.Design is the place where quality is fostered in development.Software design is the process through which requirements are translated into a representation of software.Software requirement is conducted in two steps.Preliminary design is concerned with the transformation of requirements into data.

4.2 MODULE DESCRIPTION

The system after careful analysis has been identified to be presented with the following modules.

4.2 User Module

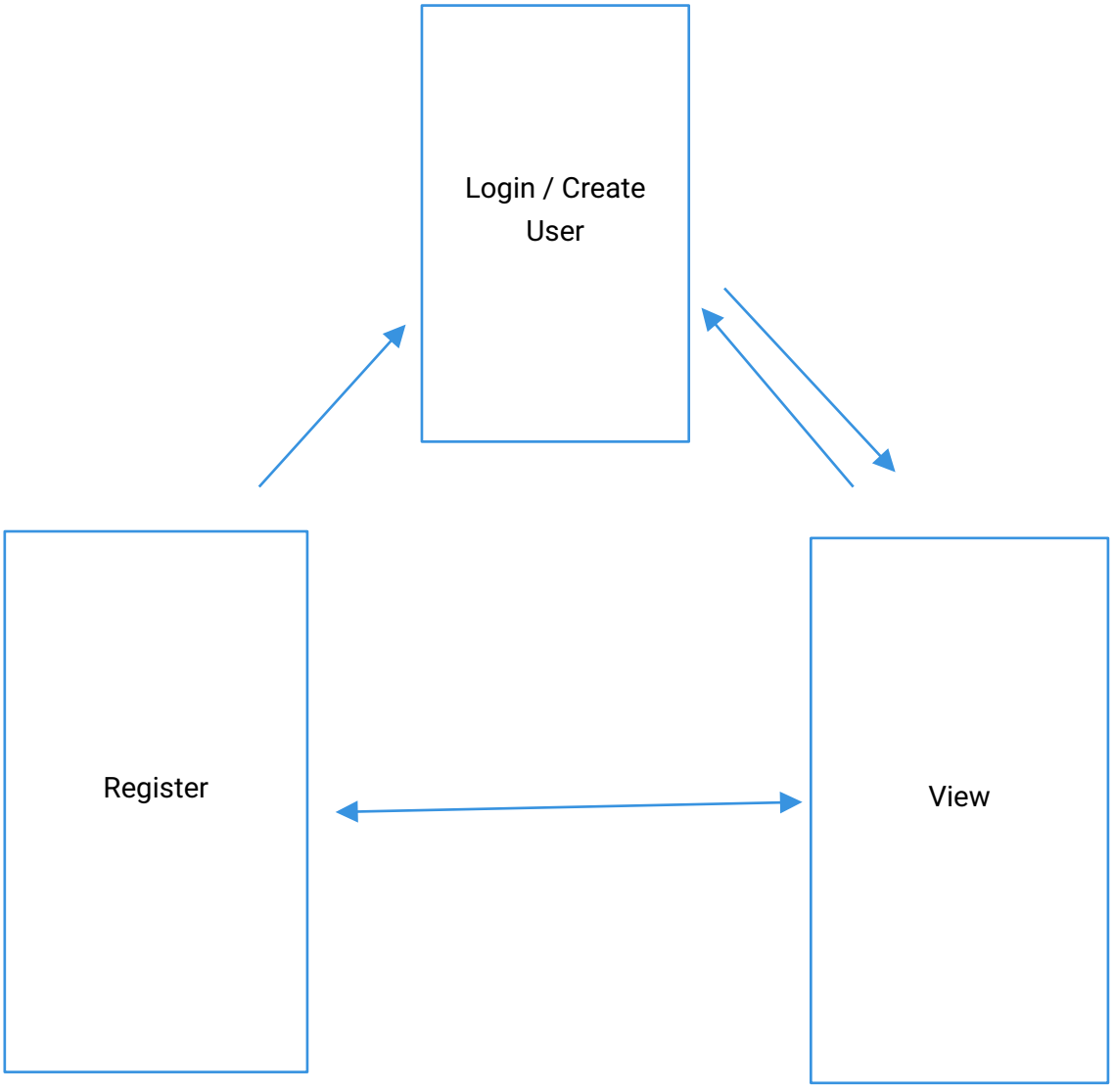


Figure 4.1 User Module

4.3 ARCHITECTURE

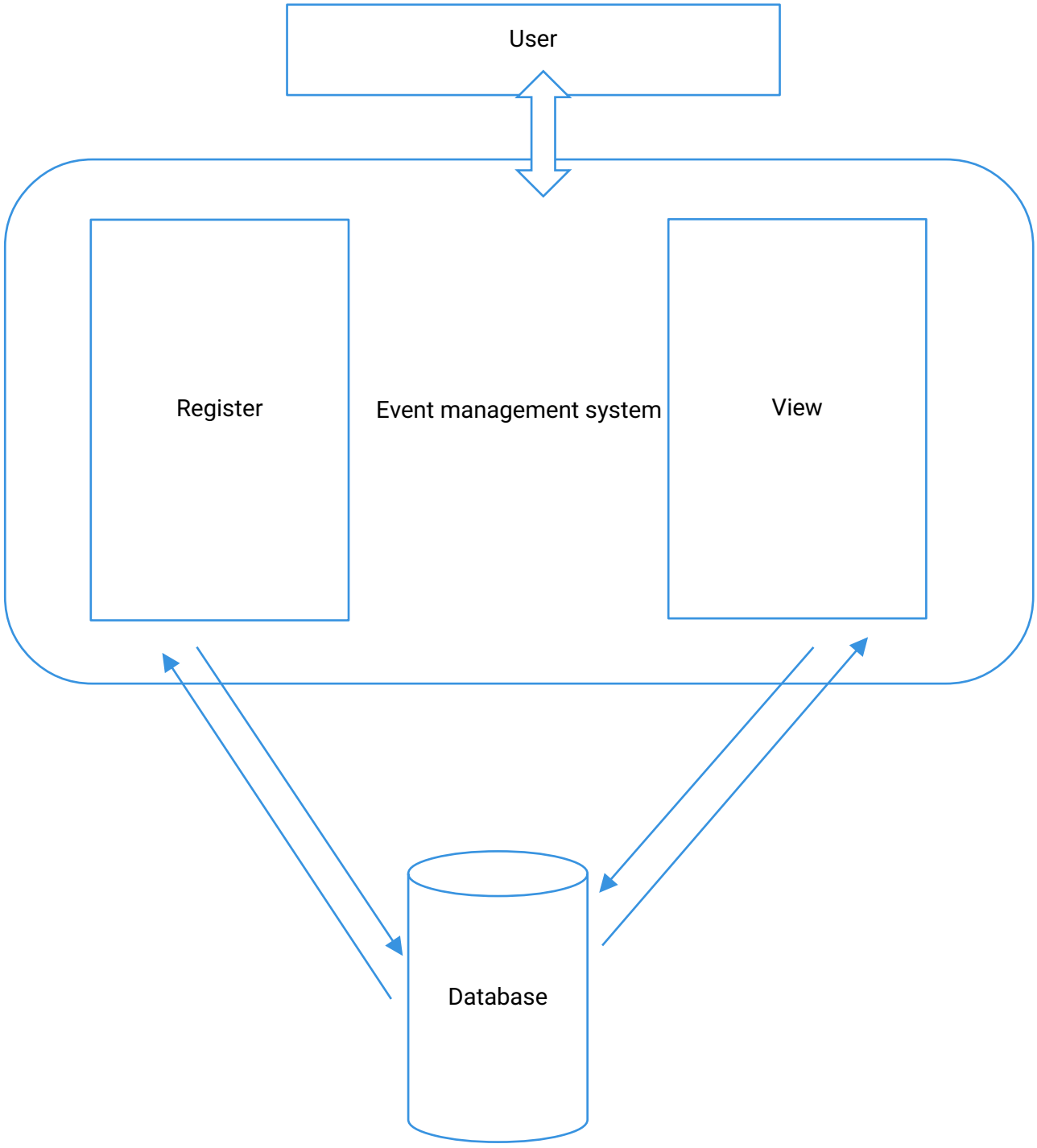
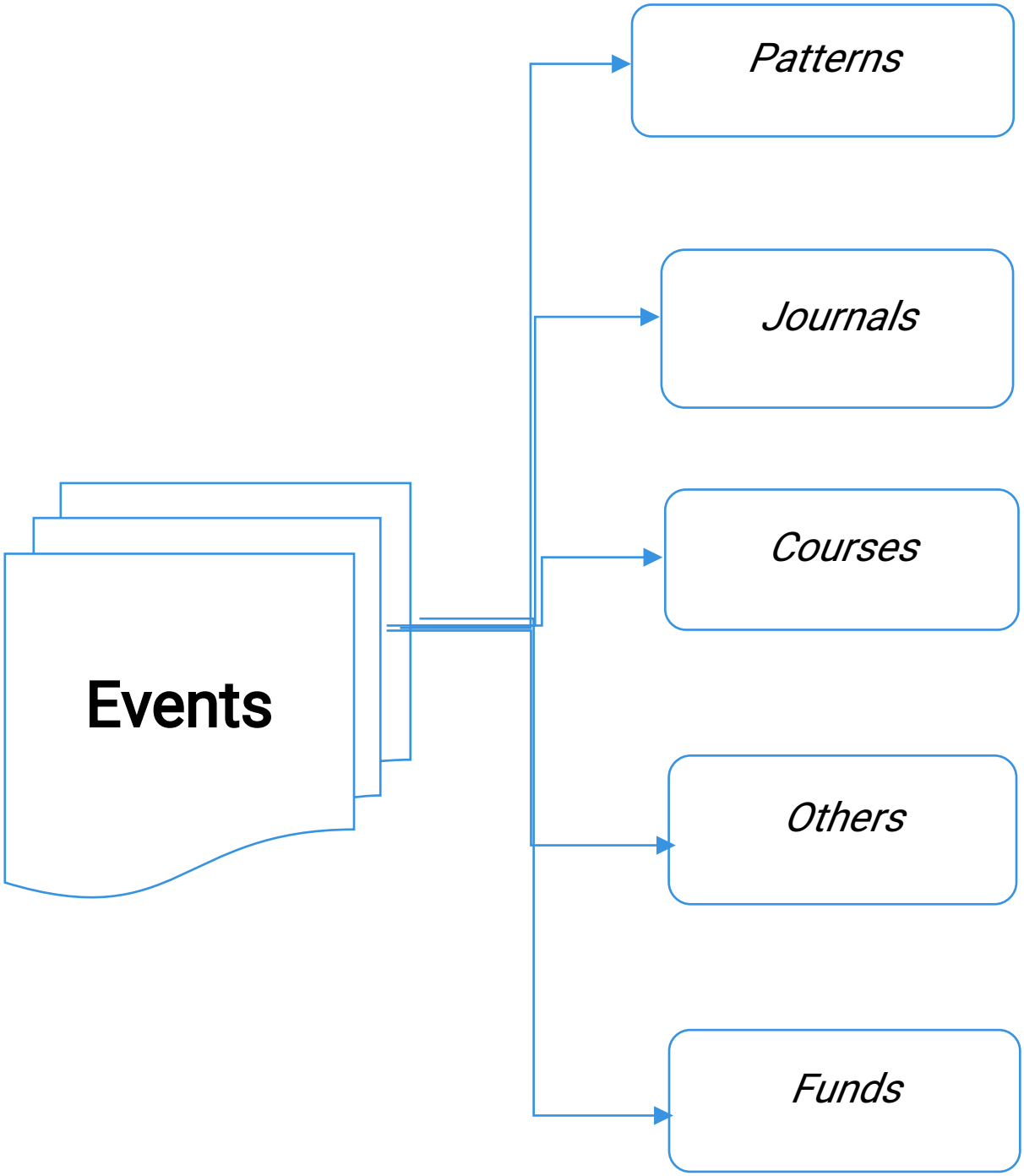


Figure 4.3 Architecture of Event Management System

4.3 DATABASE DESIGN



4.3.1 ER-Diagram

An Entity Relationship Diagram (ERD) is a graphical tool to express the overall structure of a database. It is based on a perception of a real world which consists of a set of basic objects. An entity is a person, place, thong or event of interest to the organization and about which data are captured, stored or processed. The attributes are various kinds of data that describes an entity. An association of several entities in an Entity-Relationship model is called relationship.

Attribute(s):

Attributes are the properties which define the entity type. For example, Roll_No, Name, DOB, Age, Address, Mobile_No are the attributes which defines entity type Student. In ER diagram, attribute is represented by an oval.

Key Attribute –

The attribute which uniquely identifies each entity in the entity set is called key attribute.For example, Roll_No will be unique for each student. In ER diagram, key attribute is represented by an oval with underlying lines.

Composite Attribute –

An attribute composed of many other attribute is called as composite attribute. For example, Address attribute of student Entity type consists of Street, City, State, and Country. In ER diagram, composite attribute is represented by an oval comprising of ovals.

Multivalued Attribute –

An attribute consisting more than one value for a given entity. For example, Phone_No

(can be more than one for a given student). In ER diagram, multivalued attribute is represented by double oval.

Derived Attribute –

An attribute which can be derived from other attributes of the entity type is known as derived attribute. e.g.; Age (can be derived from DOB). In ER diagram, derived attribute is represented by dashed oval.

The complete entity type Student with its attributes can be represented as:An Entity Relationship Diagram (ERD) is a graphical tool to express the overall structure of a database. It is based on a perception of a real world which consists of a set of basic objects. An entity is a person, place, thong or event of interest to the organization and about which data are captured, stored or processed. The attributes are various kinds of data that describes an entity. An association of several entities in an Entity-Relationship model is called relationship.

Participation Constraint:

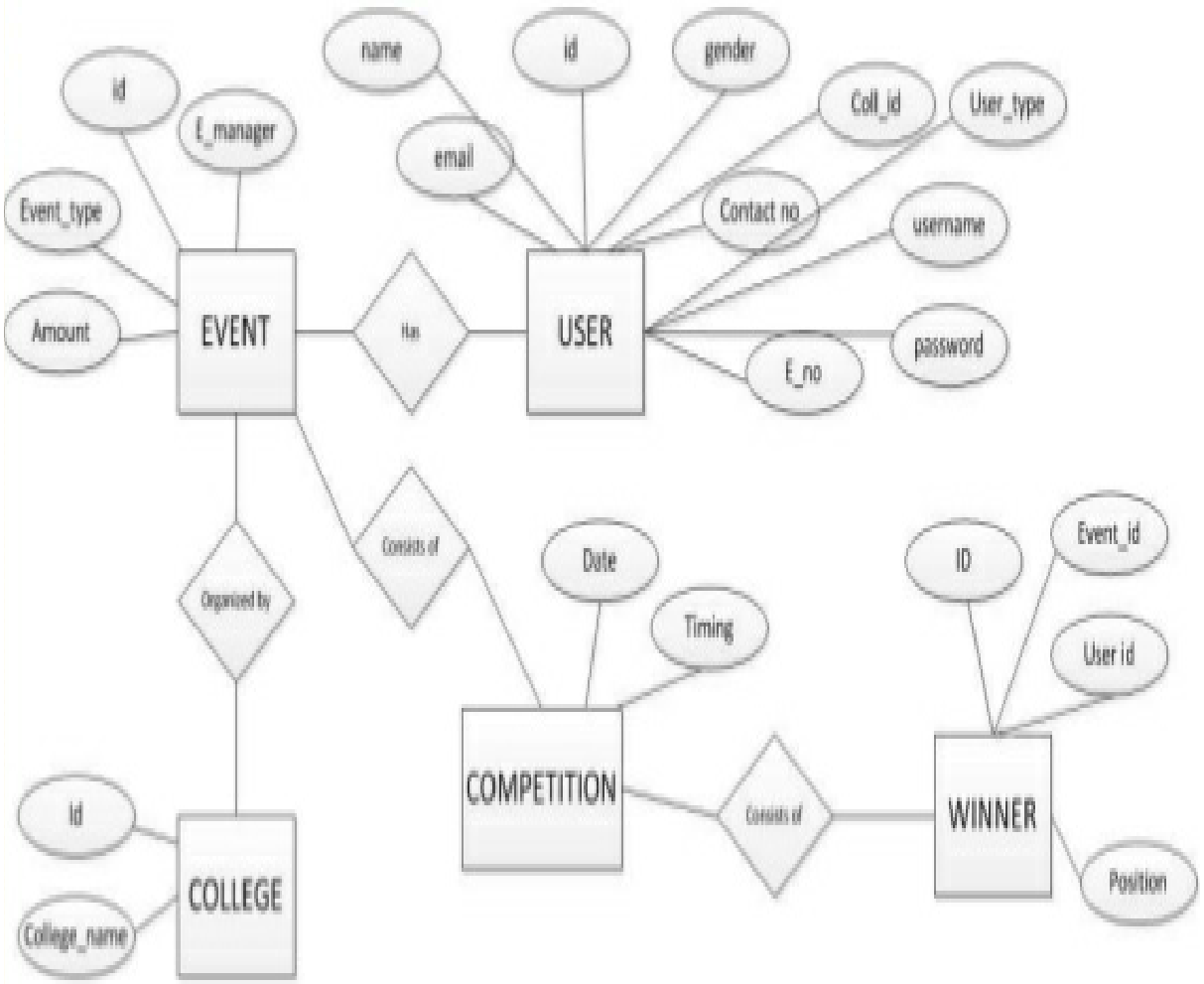
Participation Constraint is applied on the entity participating in the relationship set.

Total Participation – Each entity in the entity set must participate in the relationship. If each student must enroll in a course, the participation of student will be total. Total participation is shown by double line in ER diagram.

Partial Participation – The entity in the entity set may or may NOT participate in the relationship. If some courses are not enrolled by any of the student, the participation of course will be partial.

The diagram depicts the ‘Enrolled in’ relationship set with Student Entity set having total participation and Course Entity set having partial participation.

ER - Diagram


name
id
gender
id
E_manager
email
Coll_id
User_type
Event_type
Contact no
username
Amount
EVENT
Has
USER
password
E_no
Consists of
Date
ID
Event_id
Organized by
Timing
User id
COMPETITION
Consists of
WINNER
Id
COLLEGE
Position
College_name

4.4 UML diagram

A UML diagram is a diagram based on the UML (Unified Modeling Language) with the purpose of visually representing a system along with its main actors, roles, actions, artifacts or classes, in order to better understand, alter, maintain, or document information about the system.

UML is an acronym that stands for Unified Modeling Language. Simply put, UML is a modern approach to modeling and documenting software. In fact, it's one of the most popular business process modeling techniques.

It is based on diagrammatic representations of software components. As the old proverb says: "a picture is worth a thousand words". By usingvisual representations, we are able to better understand possible flaws or errors in software or business processes.

UML was created as a result of the chaos revolving around software development and documentation. In the 1990s, there were several different ways to represent and document software systems. The need arose for a more unified way to visually represent those systems and as a result, in 1994-1996, the UML was developed by three software engineers working at Rational Software. It was later adopted as the standard in 1997 and has remained the standard ever since, receiving only a few updates.

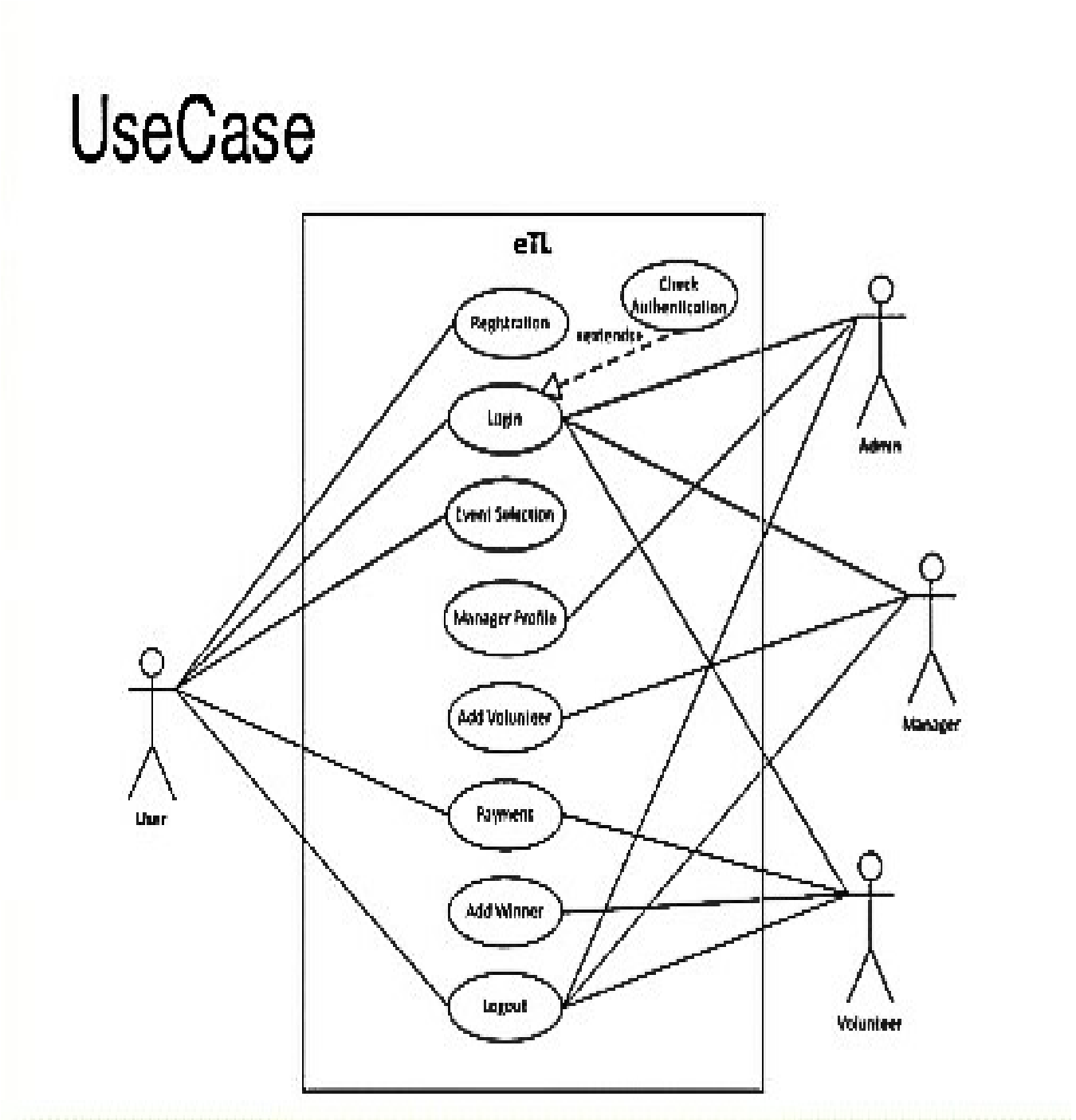
What is the use of UML?

Mainly, UML has been used as a general-purpose modeling language in the field of software engineering. However, it has now found its way into the documentation of several business processes or workflows. For example, activity diagrams, a type of UML diagram, can be used as a replacement for flowcharts. They provide both a more standardized way of modeling workflows as well as a wider range of features to improve readability and efficacy.

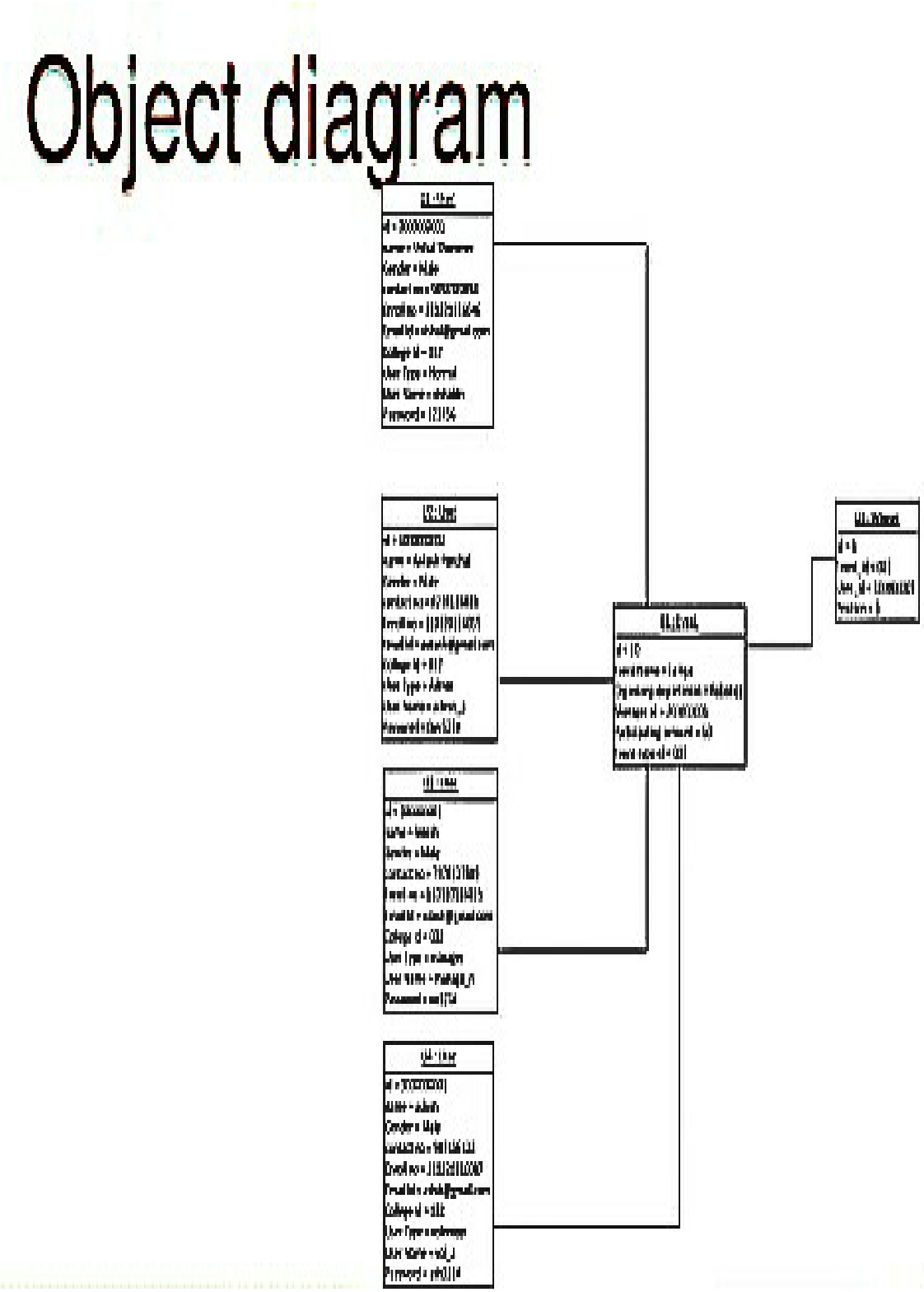
Types

- ✓ Use case diagram
- ✓ Object diagram
- ✓ Activity diagram
- ✓ Class diagram
- ✓ Sequence diagram

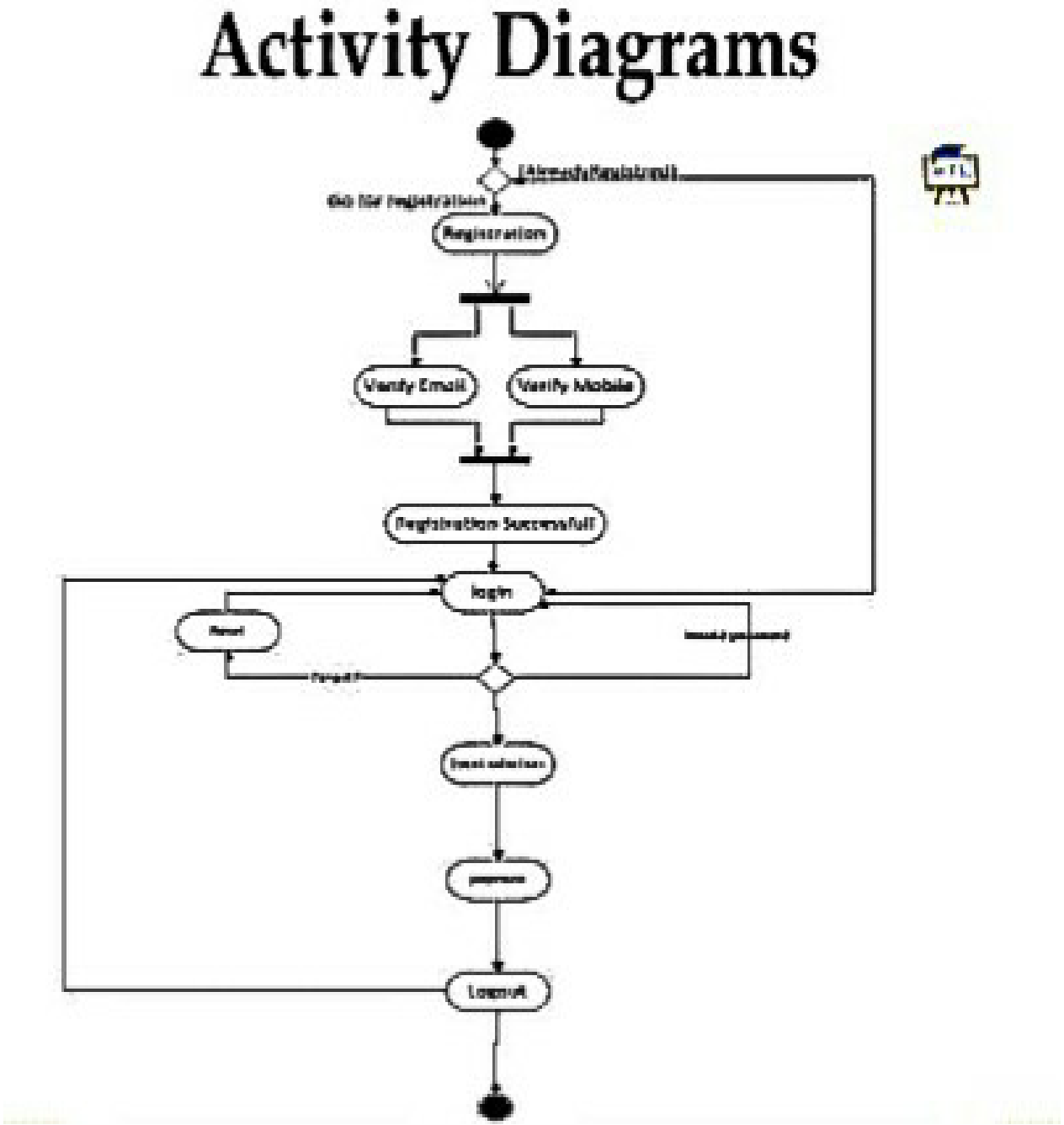
4.4.1 Use Case Diagram



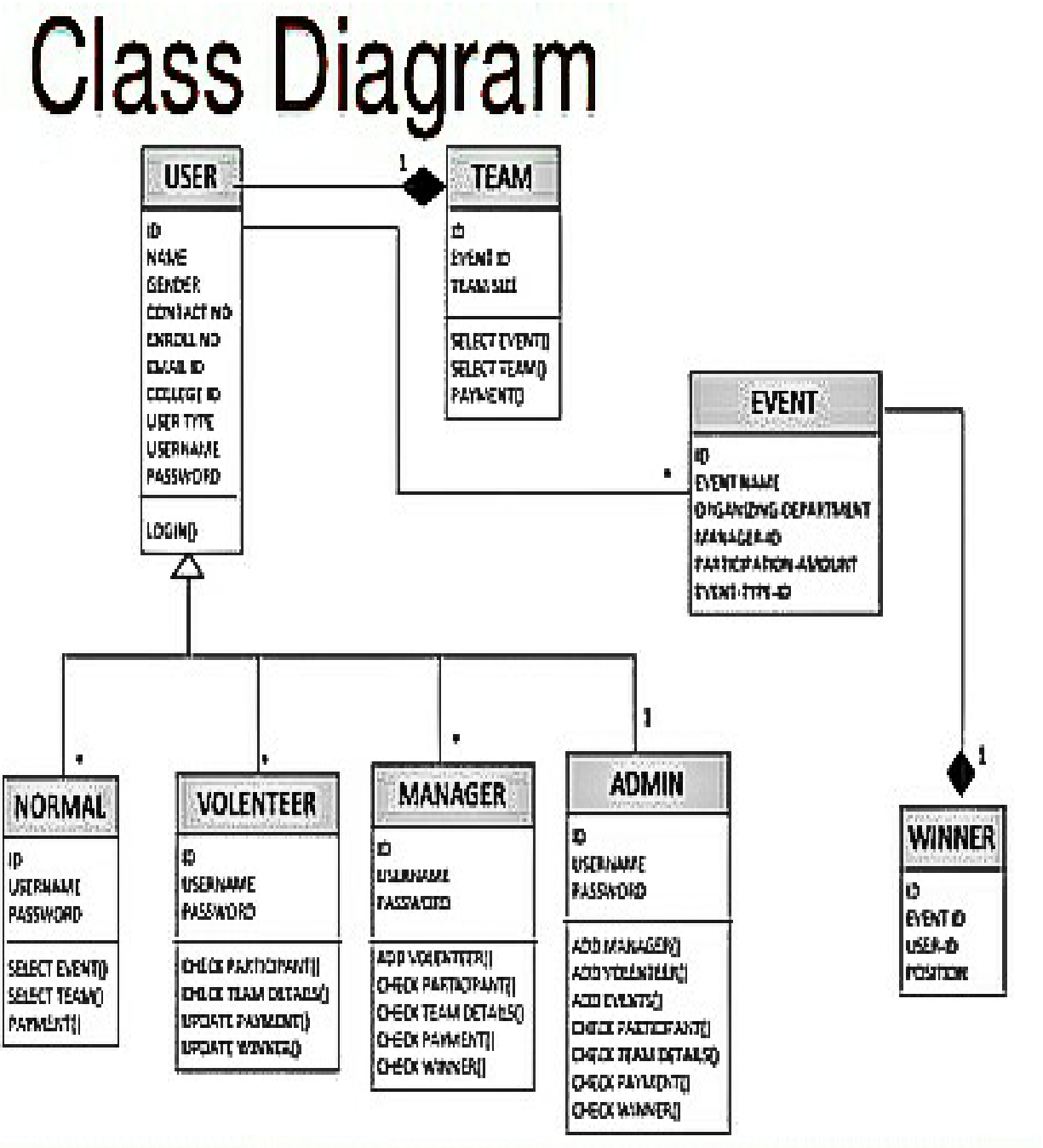
4.4.2 Object Diagram



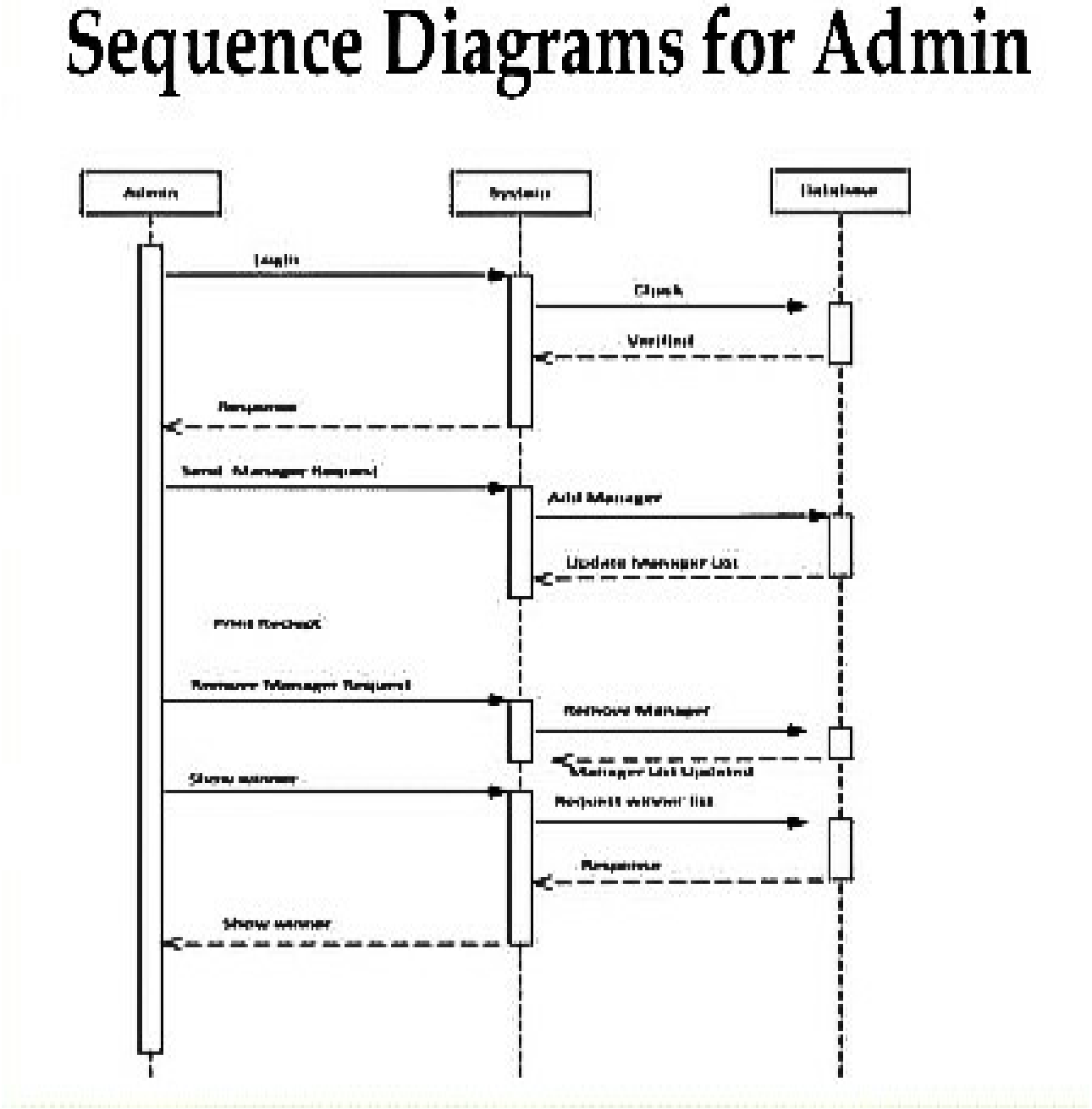
4.4.3 Activity Diagram



4.4.4 Class Diagram



4.4.5 Sequence Diagram



CHAPTER 5

DEVELOPMENT AND CODING

5.1 TECHNOLOGY DESCRIPTION

5.1.1.Php

PHP is a server-side scripting language designed primarily for web development but also used as a general-purpose programming language. Originally created by Rasmus Lerdorf in 1994,the PHP reference implementation is now produced by The PHP Development Team. PHP originally stood for personal home page but it now stands for the recursive acronym PHP: Hypertext Preprocessor.

PHP code may be embedded into HTML or HTML5 code, or it can be used in combination with various web template systems, web content management systems and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.

The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

The PHP language evolved without a written formal specification or standard until 2014, leaving the canonical PHP interpreter as a de facto standard. Since 2014 work has gone on to create a formal PHP specification.PHP development began in 1995

when Rasmus Lerdorf wrote several Common Gateway Interface (CGI) programs in C, which he used to maintain his personal homepage. He extended them to work with web forms and to communicate with databases, and called this implementation "Personal Home Page/Forms Interpreter" or PHP/FI.

PHP/FI could help to build simple, dynamic web applications. To accelerate bug reporting and to improve the code, Lerdorf initially announced the release of PHP/FI as "Hypertext pre processor Tools(PHPTools)version1.0"onthe Usenet discussiongroup comp.infosystems.www.authoring.cgi on June 8, 1995. This release already had the basic functionality that PHP has as of 2013. This included Perl-like variables, form handling, and the ability to embed HTML. The syntax resembled that of Perl but was simpler, more limited and less consistent.

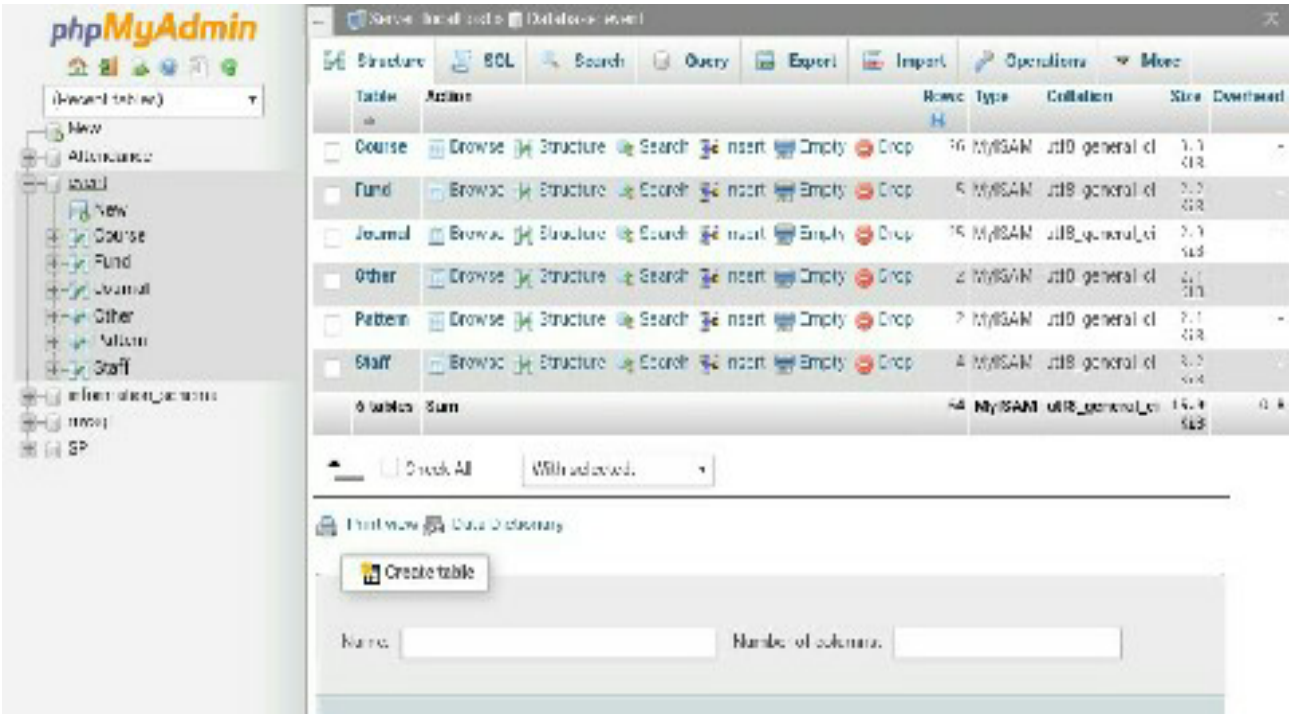


Figure 5.1.1 Phpmyadmin

Lerdorf did not intend the early PHP to become a new programming language, but it grew organically, with Lerdorf noting in retrospect. A development team began to form and, after months of work and beta testing, officially released PHP/FI 2 in November 1997.

The fact that PHP lacked an original overall design but instead developed organically has led to inconsistent naming of functions and inconsistent ordering of their parameters. In some cases, the function names were chosen to match the lower-level libraries which PHP was "wrapping",while in some very early versions of PHP the length of the function names was used internally as a hash function, so names were chosen to improve the distribution of hash values.

5.1.2 MySQL

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius' daughter, and "SQL", the abbreviation for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.

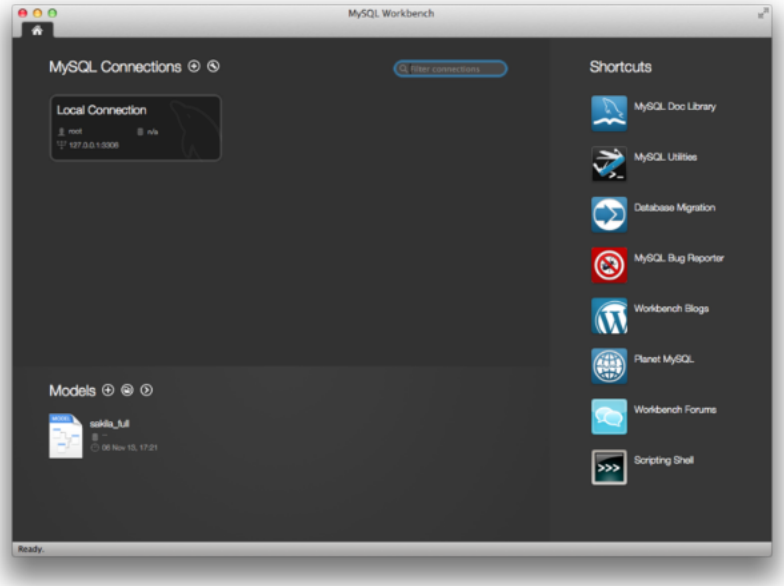


Figure 5.1.3 MySql Workbench running on OS X

For proprietary use, several paid editions are available, and offer additional functionality.MySQL is a central component of the LAMP open-source web application software stack (and other "AMP" stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python". Applications that use the MySQL database include: TYPO3, MODx, Joomla, WordPress, phpBB, MyBB, and Drupal. MySQL is also used in many high-profile, large-scale websites, including Google (though not for searches), Facebook, Twitter, Flickr, and YouTube.

5.1.3 Html

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a web server or from local storage and render them into multimedia web pages. HTML describes the structure of a web page semantically

and originally included cues for the appearance of the document.HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects, such as interactive forms, may be embedded into the rendered page. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> introduce content into the page directly. Others such as ... surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.HTML can embed programs written in a scripting language such as JavaScript which affect the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), maintainer of both the HTML and the CSS standards, has encouraged the use of CSS over explicit presentational HTML.

5.1.4 CSS

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts.[3] This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.[4]

The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/css is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents.[5]

In addition to HTML, other markup languages support the use of CSS including XHTML, plain XML, SVG, and XUL.

Page reformatting

Main article: Progressive enhancement

With a simple change of one line, a different style sheet can be used for the same page. This has advantages for accessibility, as well as providing the ability to tailor a page or site to different target devices. Furthermore, devices not able to understand the styling still display the content.

Accessibility

Without CSS, web designers must typically lay out their pages with techniques such as HTML tables that hinder accessibility for vision-impaired users (see Tableless web design#Accessibility).

5.1.5 Bootstrap - The most widely used free and open-source CSS framework

Bootstrap is one of the most popular CSS Frameworks. The current version of this framework is Bootstrap 4, which was released in 2018. Many significant features were

introduced in this release, such as new color schemes, new modifiers, new utility classes, etc.

1) Powerful responsive design

Bootstrap powers responsive design using its grid system. It's easy to use, and you can quickly create a responsive grid that will work well in all browsers. Your design will look great on all screens and resolutions.

2) Built-in libraries of resources

Bootstrap provides large libraries for front-end developers, for example, website layouts, website templates, Bootstrap themes, admin panels, and a massive collection of UI components.

The components include buttons, forms, cards, progress bars, alerts. Those are pre-build components that can save product design teams a lot of time.

3) Low learning curve

Bootstrap framework is good for web beginners. Using this tool, you can join the front-end development field. There are lots of helpful documentation and tutorials that you can rely on whenever you have questions.

4) Quickly build prototypes

Using ready-to-use components is one of the fastest ways to mock or prototype a solution. With the variables and mixins, responsive grid system, rich components, and many other powerful tools, you can prototype with ease.

More features of Bootstrap：

- Uses Flexbox
- Good documentation
- Includes HTML and JavaScript components

5.1.6 Materialize CSS - A modern responsive front-end framework based on Material Design

Materialize CSS is a responsive front-end framework created by Google in 2014. It's the right solution for anyone who wants to design websites or Android web apps because it comes with ready-to-use classes and components. You can quickly get started using its starter templates.

There are two reasons you might want to use Materialize as one of your design languages.

1) Material Design

We all know that Material Design is an essential ingredient of Google products. That is the reason why Materialize CSS became one of the most popular design languages.

So whether you are a starter or interested in Material design, Materialize CSS is the one you shouldn't miss out.

2)You know how Bootstrap works

Materialize CSS uses Bootstrap's 12-column grid format so you can quickly create responsive page layouts. You will get started even faster with the basic knowledge of a Bootstrap project.

More features of Materialize CSS:

- Mobile menus
- Compatible with Sass

5.1.7 JavaScript

JavaScript has often been dubbed the language of the web, and this is an appropriate, though not entirely accurate description. JavaScript is increasingly becoming a full-fledged programming language capable of manipulating data from the web within a web browser, changing its appearance, content, responding to user input and validating user input to forms, to name but a few of its functionalities.

With the advent of powerful JavaScript environments such as Node.js, data may also be manipulated on a web server (something formerly impossible), but further discussion of this relatively new and powerful technology are beyond the scope of this answer.

5.1.8 Jquery

jQuery is a lightweight, "write less, do more", JavaScript library.

The purpose of jQuery is to make it much easier to use JavaScript on your website.

jQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code.

jQuery also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation.

The jQuery library contains the following features:

- HTML/DOM manipulation
- CSS manipulation
- HTML event methods

- Effects and animations
- AJAX
- Utilities

There are lots of other JavaScript frameworks out there, but jQuery seems to be the most popular, and also the most extendable.

Many of the biggest companies on the Web use jQuery, such as:

- Google
- Microsoft
- IBM
- Netflix

5.1.9 Ajax

Ajax is actually a family of technologies that have been available for years. The means to make requests to the server using only JavaScript were built into Internet Explorer 5.5, but the possibilities of the technology were overlooked. It was only in 2005 that the techniques were rediscovered and used, notably to excellent effect in Google's » GMail web application.

The term Ajax, which stands for "Asynchronous JavaScript and XML", was first coined by Jesse James Garrett in his somewhat infamous article, » Ajax: A New Approach to Web Applications.

So let’s take each of those parts in isolation. Ajax is:

Asynchronous

This means that when you send a request, you wait for the response to come back, but are free to do other things while you wait. The response probably won’t come back immediately, so you set up a function that will wait for the response to be sent back by the server, and react to it once that happens.

JavaScript

JavaScript is used to make a request to the server. Once the response is returned by the server, you will generally use some more JavaScript to modify the current page’s document object model in some way to show the user that the submission went through successfully.

XML

The data that you receive back from the server will often be packaged up as a snippet of XML, so that it can be easily processed with JavaScript. This data can be anything you want, and as long as you want.

There’s nothing really new about what is happening here. We’re requesting a file (which will often be a server-side script, coded in something like PHP), and receiving a page as the response. This is how the web works already — the only difference is that now we can make these requests from JavaScript.

facilitates publication of content in multiple presentation formats based on nominal parameters. Nominal parameters include explicit user preferences, different web browsers, the type of device being used to view the content (a desktop computer or mobile device), the geographic location of the user and many other variables.

CHAPTER 6

TESTING AND IMPLEMENTATION

6.1 THE TESTING SPECTRUM

The term implementation has different meanings ranging from the conversation of a basic application to a complete replacement of a computer system. The procedures however, are virtually the same. Implementation includes all those activities that take place to convert from old system to new. The new system may be totally new replacing an existing manual or automated system or it may be major modification to an existing system. The method of implementation and time scale to be adopted is found out initially. Proper implementation is essential to provide a reliable system to meet organization requirement.

6.1.1 Unit Testing

In computer programming, unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming, a unit could be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process. It forms the basis for component testing. Ideally, each test case is independent from the others. Substitutes such as method stubs, mock objects, fakes, and test harnesses can be used to assist testing a module in isolation. Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended.

6.1.1.1 Benefits

The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. A unit test provides a strict, written contract that the piece of code must satisfy. As a result, it affords several benefits.

- **Find problems early**

  Unit testing finds problems early in the development cycle. In test-driven development (TDD), which is frequently used in both extreme programming and scrum, unit tests are created before the code itself is written. When the tests pass, that code is considered complete. The same unit tests are run against that function frequently as the larger code base is developed either as the code is changed or via an automated process with the build. If the unit tests fail, it is considered to be a bug either in the changed code or the tests themselves. The unit tests then allow the location of the fault or failure to be easily traced. Since the unit tests alert the development team of the problem before handing the code off to testers or clients, it is still early in the development process.

- **Facilitates Change**

  Unit testing allows the programmer to refactor code or upgrade system libraries at a later date, and make sure the module still works correctly (e.g., in regression testing). The procedure is to write test cases for all functions and methods so that whenever a change causes a fault, it can be quickly identified. Unit tests detect changes which may break a design contract.

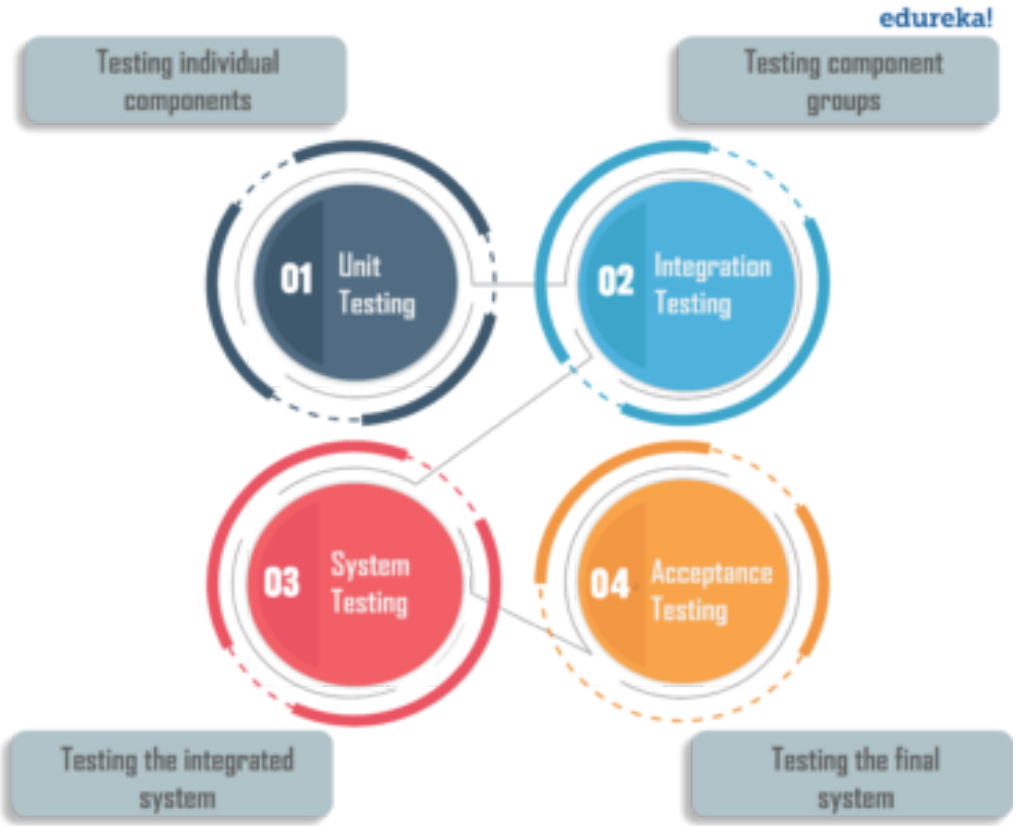
- **Simplifies Integration**

  Unit testing may reduce uncertainty in the units themselves and can be used in

a bottom-up testing style approach. By testing the parts of a program first and then testing the sum of its parts, integration testing becomes much easier.

- **Documentation**

  Unit testing provides a sort of living documentation of the system. Developers looking to learn what functionality is provided by a unit, and how to use it, can look at the unit tests to gain a basic understanding of the unit's interface (API).Unit test cases embody characteristics that are critical to the success of the unit. These characteristics can indicate appropriate/inappropriate use of a unit as well as negative behaviors that are to be trapped by the unit. A unit test case, in and of itself, documents these critical characteristics, although many software development environments do not rely solely upon code to document the product in development.



6.1.2 Integration Testing

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

- **Purpose**

  The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items. These "design items", i.e., assemblages (or groups of units), are exercised through their interfaces using black-box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter-process communication is tested and individual subsystems are exercised through their input interface. Test cases are constructed to test whether all the components within assemblages interact correctly, for example across procedure calls or process activations, and this is done after testing individual modules, i.e., unit testing. The overall idea is a "building block" approach, in which verified assemblages are added to a verified base which is then used to support the integration testing of further assemblages.Software integration testing is performed according to the software development life cycle (SDLC) after module and functional tests. The cross-dependencies for software integration testing are: schedule for integration testing, strategy and selection of the tools used for integration, define the cyclomatical complexity of the software and software architecture, reusability of modules and life-cycle and versioning management.Some different types of integration testing are

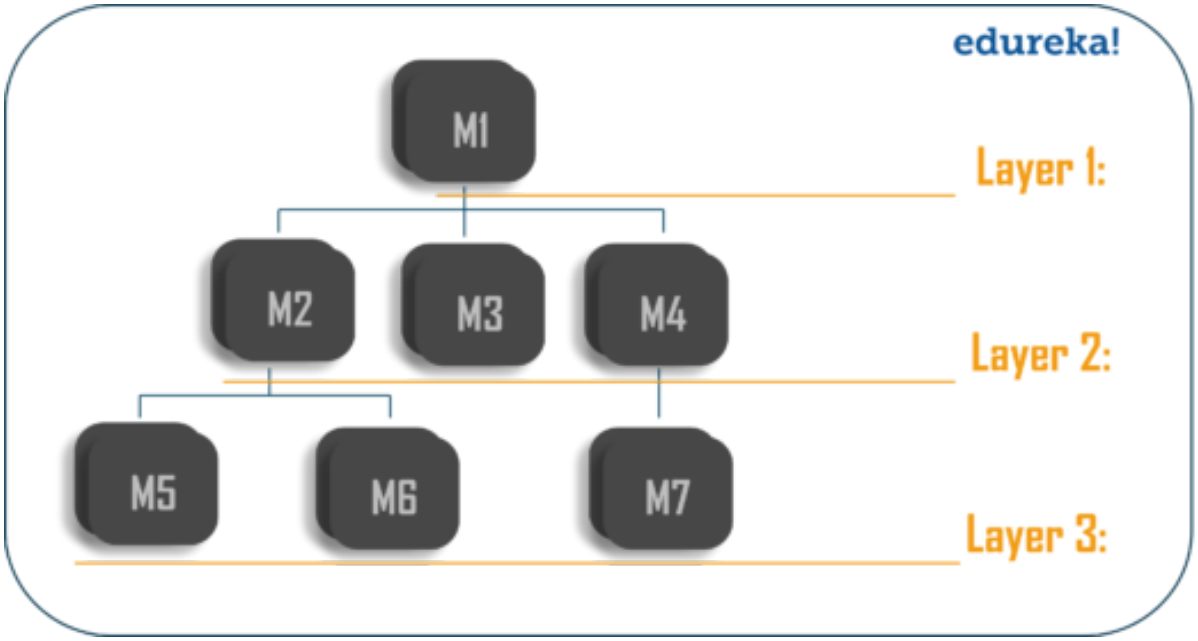
big-bang, top-down, and bottom-up, mixed (sandwich) and risky-hardest. Other Integration Patterns[2] are: collaboration integration, backbone integration, layer integration, client-server integration, distributed services integration and high-frequency integration.

- **Big Bang**

  In the big-bang approach, most of the developed modules are coupled together to form a complete software system or major part of the system and then used for integration testing. This method is very effective for saving time in the integration testing process. However, if the test cases and their results are not recorded properly, the entire integration process will be more complicated and may prevent the testing team from achieving the goal of integration testing.A type of big-bang integration testing is called "usage model testing" which can be used in both software and hardware integration testing. The basis behind this type of integration testing is to run user-like workloads in integrated user-like environments. In doing the testing in this manner, the environment is proofed, while the individual components are proofed indirectly through their use. Usage Model testing takes an optimistic approach to testing, because it expects to have few problems with the individual components. The strategy relies heavily on the component developers to do the isolated unit testing for their product. The goal of the strategy is to avoid redoing the testing done by the developers, and instead flesh-out problems caused by the interaction of the components in the environment. For integration testing, Usage Model testing can be more efficient and provides better test coverage than traditional focused functional integration testing. To be more efficient and accurate, care must be used in defining the user-like workloads for creating realistic scenarios in exercising the environment. This gives confidence that the integrated environment will work as expected for the target customers.

- **Top-down And Bottom-up**

  Bottom-up testing is an approach to integrated testing where the lowest level components are tested first, then used to facilitate the testing of higher level components. The process is repeated until the component at the top of the hierarchy is tested.All the bottom or low-level modules, procedures or functions are integrated and then tested. After the integration testing of lower level integrated modules, the next level of modules will be formed and can be used for integration testing. This approach is helpful only when all or most of the modules of the same development level are ready.



  This method also helps to determine the levels of software developed and makes it easier to report testing progress in the form of a percentage.Top-down testing is an approach to integrated testing where the top integrated modules are tested and the branch of the module is tested step by step until the end of the related module.Sandwich testing is an approach to combine top down testing with bottom up testing.

6.1.3 Software Verification And Validation

In software project management, software testing, and software engineering, verification and validation (V&V) is the process of checking that a software system meets specifications and that it fulfills its intended purpose. It may also be referred to as software quality control. It is normally the responsibility of software testers as part of the software development lifecycle. Validation checks that the product design satisfies or fits the intended use (high-level checking), i.e., the software meets the user requirements.This is done through dynamic testing and other forms of review.Verification and validation are not the same thing, although they are often confused. Boehm succinctly expressed the difference between

- Validation: Are we building the right product?
- Verification: Are we building the product right?

According to the Capability Maturity Model (CMMI-SW v1.1)

- Software Verification: The process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase.
- Software Validation: The process of evaluating software during or at the end of the development process to determine whether it satisfies specified requirements.

In other words, software verification is ensuring that the product has been built according to the requirements and design specifications, while software validation ensures that the product meets the user's needs, and that the specifications were correct in the first place. Software verification ensures that "you built it right". Software validation ensures that "you built the right thing". Software validation confirms that the product, as provided, will fulfill its intended use.

From testing perspective:

- Fault – wrong or missing function in the code.
- Failure – the manifestation of a fault during execution.
- Malfunction – according to its specification the system does not meet its specified functionality.

Both verification and validation are related to the concepts of quality and of software quality assurance. By themselves, verification and validation do not guarantee software quality; planning, traceability, configuration management and other aspects of software engineering are required.Within the modeling and simulation (M&S) community, the definitions of verification, validation and accreditation are similar:

- M&S Verification is the process of determining that a computer model, simulation, or federation of models and simulations implementations and their associated data accurately represent the developer's conceptual description and specifications.
- M&S Validation is the process of determining the degree to which a model, simulation, or federation of models and simulations, and their associated data are accurate representations of the real world from the perspective of the intended use(s).
- Accreditation is the formal certification that a model or simulation is acceptable to be used for a specific purpose.

The definition of M&S validation focuses on the accuracy with which the M&S represents the real-world intended use(s). Determining the degree of M&S accuracy is required because all M&S are approximations of reality, and it is usually critical to determine if the degree of approximation is acceptable for the intended use(s). This stands in contrast to software validation.

- **Classification of Methods**

  In mission-critical software systems, where flawless performance is absolutely necessary, formal methods may be used to ensure the correct operation of a system. However, often for non-mission-critical software systems, formal methods prove to be very costly and an alternative method of software V&V must be sought out. In such cases, syntactic methods are often used.

- **Test Cases**

  A test case is a tool used in the process. Test cases may be prepared for software verification and software validation to determine if the product was built according to the requirements of the user. Other methods, such as reviews, may be used early in the life cycle to provide for software validation.

6.1.4 Black-Box Testing

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing: unit, integration, system and acceptance. It typically comprises most if not all higher level testing, but can also dominate unit testing as well.

- **Test Procedures**

  Specific knowledge of the application's code/internal structure and programming knowledge in general is not required. The tester is aware of what the software is supposed to do but is not aware of how it does it. For instance, the tester is aware that a particular input returns a certain, invariable output but is not aware of how the software produces the output in the first place.

- **Test Cases**

  Test cases are built around specifications and requirements, i.e., what the application is supposed to do. Test cases are generally derived from external descriptions of the software, including specifications, requirements and design parameters. Although the tests used are primarily functional in nature, non-functional tests may also be used. The test designer selects both valid and invalid inputs and determines the correct output, often with the help of an oracle or a previous result that is known to be good, without any knowledge of the test object's internal structure.

- **Test Design Techniques**

  Typical black-box test design techniques include:

  - Decision table testing
  - All-pairs testing
  - Equivalence partitioning
  - Boundary value analysis
  - Cause–effect graph
  - State transition testing
  - Use case testing
  - Domain analysis
  - Combining technique

6.1.5 White-Box Testing

White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing). In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit, e.g. in-circuit testing (ICT). White-box testing can be applied at the unit, integration and system levels of the software testing process. Although traditional testers tended to think of white-box testing as being done at the unit level, it is used for integration and system testing more frequently today. It can test paths within a unit, paths between units during integration, and between subsystems during a system–level test. Though this method of test design can uncover many errors or problems, it has the potential to miss unimplemented parts of the specification or missing requirements.

White-box test design techniques include the following code coverage criteria:

- Control flow testing
- Data flow testing
- Branch testing
- Statement coverage
- Decision coverage
- Modified condition/decision coverage
- Prime path testing
- Path testing

White-box testing is a method of testing the application at the level of the source code. These test cases are derived through the use of the design techniques mentioned above: control flow testing, data flow testing, branch testing, path testing, statement coverage and decision coverage as well as modified condition/decision coverage. White-box testing is the use of these techniques as guidelines to create an error free environment by examining any fragile code. These White-box testing techniques are the building blocks of white-box testing, whose essence is the careful testing of the application at the source code level to prevent any hidden errors later on.[1] These different techniques exercise every visible path of the source code to minimize errors and create an error-free environment. The whole point of white-box testing is the ability to know which line of the code is being executed and being able to identify what the correct output should be.

6.1.5.1 Levels

1. Unit testing. White-box testing is done during unit testing to ensure that the code is working as intended, before any integration happens with previously tested code. White-box testing during unit testing catches any defects early on and aids in any defects that happen later on after the code is integrated with the rest of the application and therefore prevents any type of errors later on.
2. Integration testing. White-box testing at this level are written to test the interactions of each interface with each other. The Unit level testing made sure that each code was tested and working accordingly in an isolated environment and integration examines the correctness of the behaviour in an open environment through the use of white-box testing for any interactions of interfaces that are known to the programmer.

3. Regression testing. White-box testing during regression testing is the use of recycled white-box test cases at the unit and integration testing levels.

6.1.5.2 Basic Procedures

White-box testing's basic procedures involves the tester having a deep level of understanding of the source code being tested. The programmer must have a deep understanding of the application to know what kinds of test cases to create so that every visible path is exercised for testing. Once the source code is understood then the source code can be analyzed for test cases to be created. These are the three basic steps that white-box testing takes in order to create test cases:

1. Input involves different types of requirements, functional specifications, detailed designing of documents, proper source code, security specifications. This is the preparation stage of white-box testing to layout all of the basic information.
2. Processing involves performing risk analysis to guide whole testing process, proper test plan, execute test cases and communicate results. This is the phase of building test cases to make sure they thoroughly test the application the given results are recorded accordingly.
3. Output involves preparing final report that encompasses all of the above preparations and results.

6.1.5.3 Advantages

White-box testing is one of the two biggest testing methodologies used today. It has several major advantages:

1. Side effects of having the knowledge of the source code is beneficial to thorough testing.

2. Optimization of code by revealing hidden errors and being able to remove these possible defects.
3. Gives the programmer introspection because developers carefully describe any new implementation.
4. Provides traceability of tests from the source, allowing future changes to the software to be easily captured in changes to the tests.
5. White box tests are easy to automate.
6. White box testing give clear, engineering-based, rules for when to stop testing.

6.1.5.4 Disadvantages

Although white-box testing has great advantages, it is not perfect and contains some disadvantages:

1. White-box testing brings complexity to testing because the tester must have knowledge of the program, including being a programmer. White-box testing requires a programmer with a high level of knowledge due to the complexity of the level of testing that needs to be done.
2. On some occasions, it is not realistic to be able to test every single existing condition of the application and some conditions will be untested.
3. The tests focus on the software as it exists, and missing functionality may not be discovered.

6.1.6 System Testing

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black-box testing, and as such, should require no

knowledge of the inner design of the code or logic. As a rule, system testing takes, as its input, all of the "integrated" software components that have passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called assemblages) or between any of the assemblages and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

- **Testing The Whole System**

  System testing is performed on the entire system in the context of a Functional Requirement Specification(s) (FRS) and/or a System Requirement Specification (SRS). System testing tests not only the design, but also the behavior and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specification(s).

Step 5: Performance Testing

Once you know your app is functional and responsive, you need to check its performance under a heavy load. Performance testing includes testing under different internet speeds as well as under normal and peak loads.This article gives a good insight into how a basic speed test can be performed with free tools, and how to use the data for preparing an optimization plan. It’s not actually a proper in-depth performance and stress testing, but a good starting point. Stress testing is useful to determine the breaking point of the app; it involves putting the app under increasing amounts of stress until it stops functioning. After all, you need to discover your app’s breaking point before the users do.

CHAPTER 7

CONCLUSION

Social media has a massive impact on the events industry due to details being updated and published for events to increase individuals awareness of events running therefore attracting a vast amount of potential customers to increase profit to the business. Tickets can now be brought via Facebook, Twitter and websites this impacts the events easy access to buy tickets online which increases individuals to attend due to saving time to buy a ticket. Technology has made a massive change to the events industry. Apps for phones are designed so the public can get details on upcoming and on going events Greenwell, Danzey-Bussell and Shonk (2013) suggests Twitter,Youtube and Foursquare are now used to promote and market many events. As new technologies emerge, marketers will continue to reach out to new consumers through these new mediums.

The Event Management Industry has a a high potential for growth in the coming years. There is a wide scope of learning in the field of Event Management. It helps you build a strong professional network as you work with people across different domains.Event Management System is user friendly and cost effective system, it is customized with activities related to event management life-cycle. It provides a new edge to management industry. SolutionDot always keep your objectives and goals on top priority while developing any plan of work.