AI BASED COLLEGE ENQUIRY CHATBOT SYSTEM

A Major Project Report

Submitted in partial fulfillment of the

Requirements for the award of the Degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

GONAPATI SIVABHASKAR REDDY (15001A0550)

AKETI VENKATA SAI SUMANTH (15001A0557)

Under the Esteemed Guidance of

Dr. S. VASUNDRA M.Tech.,Ph.D.

Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

COLLEGE OF ENGINEERING

(Autonomous)

ANANTHAPURAMU-515002

ANDHRA PRADESH

2019

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

COLLEGE OF ENGINEERING

(Autonomous)

ANANTHAPURAMU-515002
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION BY THE CANDIDATE

We, **G. SIVABHASKAR REDDY** bearing the Adm.No.**15001A0550** and **A.V. SAI SUMANTH** bearing the Adm.No.**15001A0557** hereby declare that Major Project Report entitled **"AI BASED COLLEGE ENQUIRY CHATBOT SYSTEM"** under the guidance **Dr. S.VASUNDRA** Professor and N.S.S Coordinator, JNTUA is submitted in partial fulfillment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING.**

This is a record of bonafide work carried out by us and results embodied in this project have not been reproduced or copied from any source. The results embodied in this project report have not been submitted to other university or institute for the award of any other degree or diploma.

G. SIVABHASKAR REDDY	**A.V. SAI SUMANTH**
(15001A0550)	**(15001A0557)**

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

COLLEGE OF ENGINEERING

(Autonomous)

ANANTHAPURAMU-515002

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled **"AI BASED COLLEGE ENQUIRY CHATBOT SYSTEM "** is bonafide work of **G.SIVABHASKAR REDDY** bearing Adm.No:**15001A0550** and **A.V. SAI SUMANTH** bearing Adm.No: **15001A0557** is submitted to the faculty of Computer Science and Engineering, in partial fulfillment of the requirements for the award of degree of Bachelor Of Technology in Computer Science And Engineering from Jawaharlal Nehru Technological University Anantapur, College Of Engineering(*Autonomous*), Ananthapuramu.

Signature of the Guide
Dr. S. VASUNDRA M.Tech., Ph.D.
Professor
Department of Computer
Science and Engineering
JNTUA College of Engineering
Ananthapuramu.

Signature of H.O.D
Dr. A.SURESH BABU M.Tech.,Ph.D
Professor and H.O.D
Department of Computer
Science and Engineering
JNTUA College of Engineering
Ananthapuramu.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that I have now the opportunity to express my gratitude for all of them.

The first and foremost **Dr. S. VASUNDRA**, Professor of Computer Science and Engineering Department, JNTUA CEA and N.S.S Coordinator, JNTUA who has extended her support for the success of this project. Her wide knowledge and logical way of thinking have made a deep impression on us. Her understanding, encouragement, guidance has provided the basis for this thesis. Her source of inspiration of innovative ideas and her support is well to all his students and colleagues.

We wish to thank **Prof. A. SURESH BABU,** Professor and Head of Computer Science And Engineering Department, JNTUA College of Engineering (*Autonomous*), Ananthapuramu. His wide support, knowledge, enthusiastic encouragement impressed me to better involvement into my project thesis and technical design also his ethical morals helped me to develop my personal and technical skills to deploy my project in success.

We wish to thank **Prof. K. RAMANAIDU**, Principal of JNTUA College of Engineering (*Autonomous*), Ananthapuramu who has extended his support for the success of this project.

We wish to thank **Prof. E. KESHAVA REDDY**, Vice Principal of JNTUA College of Engineering (*Autonomous*), Ananthapuramu who has extended his support for the success of this project.

We express our sincere thanks to project committee members, faculty and staff of Computer Science and Engineering Department, JNTUA College of Engineering (*Autonomous*), Ananthapuramu for their valuable guidance and technical support.

We also thank family members and friends for their moral support and constant encouragement. We are very much thankful to one and all who helped us for the successful completion of the project.

With gratitude

G. SIVABHASKAR REDDY
(15001A0550)

A.V.SAI SUMANTH
(15001A0557)

ABSTRACT

Chat bots typically provide a text-based user interface, allowing the user to type commands and receive text as well as text to speech response. Chat bots are usually a stateful services, remembering previous commands (and perhaps even conversation) in order to provide functionality. When chat bot technology is integrated with popular web services it can be utilized securely by an even larger audience.

This Chat Bot project is built using artificial algorithms that analyzes user's queries and understand user's message. This System is a web application built using HTML&CSS for front-end. For storing data, we use MySQL and also, we make use of JAVA. The System uses built-in artificial intelligence to answer the query. The answers are appropriate what the user queries. If the answer found to be invalid, user just need to select the invalid answer button which will notify the admin about the incorrect answer. Admin can view and rectify invalid answer through portal via login.

The user does not have to personally go to the college for enquiry. The system replies as if a real person is talking to the user. The user can query about the college related activities through online with the help of this web application and get updated about the college activities. This chatbot system contains Admin login to access various helping pages, Bot chat where user can interact with the system, Text to Speech where the bot speaks out. It can be used in many colleges and various firms.

CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 PROBLEM DEFINITION

AI Chatbot, college management system, college administration software or student administration system is a management information system for education establishments to manage student data. Student information systems provide capabilities for registering students in courses, documenting grading, transcripts, results of student tests and other. A chatbot (also known as a talkbot, chatterbot, Bot, IM bot, interactive agent, or Artificial Conversational Entity) is a computer program which conducts a conversation via auditory or textual methods. Such programs are often designed to convincingly simulate how a human would behave as a conversational partner, thereby passing the Turing test.

User interfaces for software applications can come in a variety of formats, ranging from command-line, graphical, web application, and even voice. While the most popular user interfaces include graphical and web-based applications, occasionally the need arises for an alternative interface. Whether due to multi-threaded complexity, concurrent connectivity, or details surrounding execution of the service, a chat bot-based interface may suit the need.

Chat bots typically provide a text-based user interface, allowing the user to type commands and receive text as well as text to speech response. Chat bots are usually a stateful services, remembering previous commands (and perhaps even conversation) in order to provide functionality. When chat bot technology is integrated with popular web services it can be utilized securely by an even larger audience.

1.2 EXISTING SYSTEM

In the existing system student chatbot project analyzes user's queries and understand user's message. Students can chat using specific format that specific format the user has to follow. The System uses built in normally to answer the query. The answers are appropriate what the user queries. If the answer found to invalid, user can't do anything. In this some problems are present those are solved in proposed system. There are many applications that are incorporating a human appearance and intending to simulate human dialog, but in most of the cases the knowledge of the conversational bot is stored in a database created by a human expert. However, very few researches have investigated the idea of creating a chat-bot with an artificial character and personality starting from web pages or plain text about a certain person. This paper describes an approach to the idea of identifying the most important facts in texts describing the life (including the personality) of an historical figure for building a conversational agent that could be used in middle-school CSCL scenarios.

1.3 PROPOSED SYSTEM

A Student chatbot project is built using artificial algorithms that analyzes user's queries and understand user's message. This System is a web application which provides answer to the query of the student. Students just have to query through the bot which is used for chatting. Students can chat using any format there is no specific format the user has to follow. The System uses built in artificial intelligence to answer the query. The answers are appropriate what the user queries. If the answer found to invalid, user just need to select the invalid answer button which will notify the admin about the incorrect answer. Admin can view invalid answer through portal via login. System allows admin to delete the invalid answer or to add a specific answer of that equivalent question. The User can query any college related activities through the system. The user does not have to personally go to the college for enquiry. The System analyzes the question and then answers to the user. The system answers to the query as if it is answered by the person. With the help of artificial intelligence, the system answers the query asked by the students. The system replies using an effective Graphical user interface which implies that as if a real person is talking to the user. The user can query about the college related activities through online with the help of this web application. This system helps the student to be updated about the college activities

- A Student Information Chat Bot project is built using artificial algorithms that analyzes user's queries and understand user's message.
- This System is a web application which provides answer to the query of the student very effectively.
- Students just have to query through the bot which is used for chatting.
- Students can chat using any format, as there is no specific format that the user has to follow.
- The System uses built in artificial intelligence to answer the query.
- The answers are appropriate what the user queries.

- If the answer found to be invalid, user just need to select the invalid answer button which will notify the admin about the incorrect answer.
- Admin can view invalid answer through portal via login
- System allows admin to delete the invalid answer or to add a specific answer of that equivalent question.
- The user does not have to personally go to the college for enquiry.
- The system analyzes the question and then answers to the user.
- The system replies using an effective Graphical user interface which implies that as if a real person is talking to the user.
- The user can query about the college related activities through online with the help of this web application.
- This system helps the student to be updated about the college activities.

1.4 SOFTWARE MODEL

Software Development Life Cycle (SDLC) is a process used by the software industry to design, develop and test high quality software. The SDLC aims to produce high-quality software that meets or exceeds customer expectations, reaches completion within times and cost estimates.

- SDLC is the acronym of Software Development Life Cycle.
- It is also called as Software Development Process.
- SDLC is a framework defining tasks performed at each step in the software development process.
- ISO/IEC 12207 is an international standard for software life-cycle processes. It aims to be the standard that defines all the tasks required for developing and maintaining software.

What is SDLC?

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software as shown in Fig 1.1. The life cycle defines a methodology for improving the quality of software and the overall development process.

The following figure is a graphical representation of the various stages of a typical SDLC.

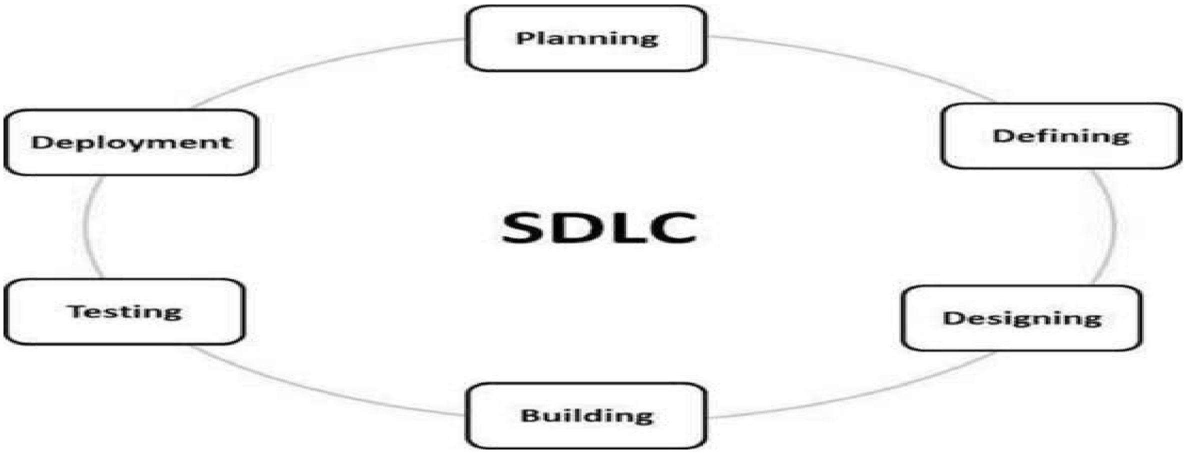


Fig 1.1 SDLC

A typical Software Development Life Cycle consists of the following stages −

Stage 1: Planning and Requirement Analysis

Requirement analysis is the most important and fundamental stage in SDLC. It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product feasibility study in the economical, operational and technical areas.

Planning for the quality assurance requirements and identification of the risks associated with the project is also done in the planning stage. The outcome of the technical feasibility study is to define the various technical approaches that can be followed to implement the project successfully with minimum risks.

Stage 2: Defining Requirements

Once the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved from the customer or the market analysts. This is done through an **SRS (Software Requirement Specification)** document which consists of all the product requirements to be designed and developed during the project life cycle.

Stage 3: Designing the Product Architecture

SRS is the reference for product architects to come out with the best architecture for the product to be developed. Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a DDS - Design Document Specification.

This DDS is reviewed by all the important stakeholders and based on various parameters as risk assessment, product robustness, design modularity, budget and time constraints, the best design approach is selected for the product.

A design approach clearly defines all the architectural modules of the product along with its communication and data flow representation with the external and third party modules (if any). The internal design of all the modules of the proposed architecture should be clearly defined with the minutest of the details in DDS.

Stage 4: Building or Developing the Product

In this stage of SDLC the actual development starts and the product is built. The programming code is generated as per DDS during this stage. If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle.

Developers must follow the coding guidelines defined by their organization and programming tools like compilers, interpreters, debuggers, etc. are used to generate the code. Different high level programming languages such as C, C++, Pascal, Java and PHP are used for coding. The programming language is chosen with respect to the type of software being developed.

Stage 5: Testing the Product

This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC. However, this stage refers to the testing only stage of the product where product defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.

Stage 6: Deployment in the Market and Maintenance

Once the product is tested and ready to be deployed it is released formally in the appropriate market. Sometimes product deployment happens in stages as per the business strategy of that organization. The product may first be released in a limited segment and tested in the real business environment (UAT- User acceptance testing).

Then based on the feedback, the product may be released as it is or with suggested enhancements in the targeting market segment. After the product is released in the market, its maintenance is done for the existing customer base.

SDLC Models

There are various software development life cycle models defined and designed which are followed during the software development process. These models are also referred as Software Development Process Models". Each process model follows a Series of steps unique to its type to ensure success in the process of software development.

Following are the most important and popular SDLC models followed in the industry & miuns;

- Waterfall Model
- Iterative Model
- Spiral Model
- V-Model
- Big Bang Mode

SDLC - Waterfall Model

The Waterfall Model was the first Process Model to be introduced. It is also referred to as a **linear-sequential life cycle model**. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

The Waterfall model is the earliest SDLC approach that was used for software development. The waterfall Model as shown in Fig.1.2 illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap.

Waterfall Model - Design

Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.

The following illustration is a representation of the different phases of the Waterfall Model.

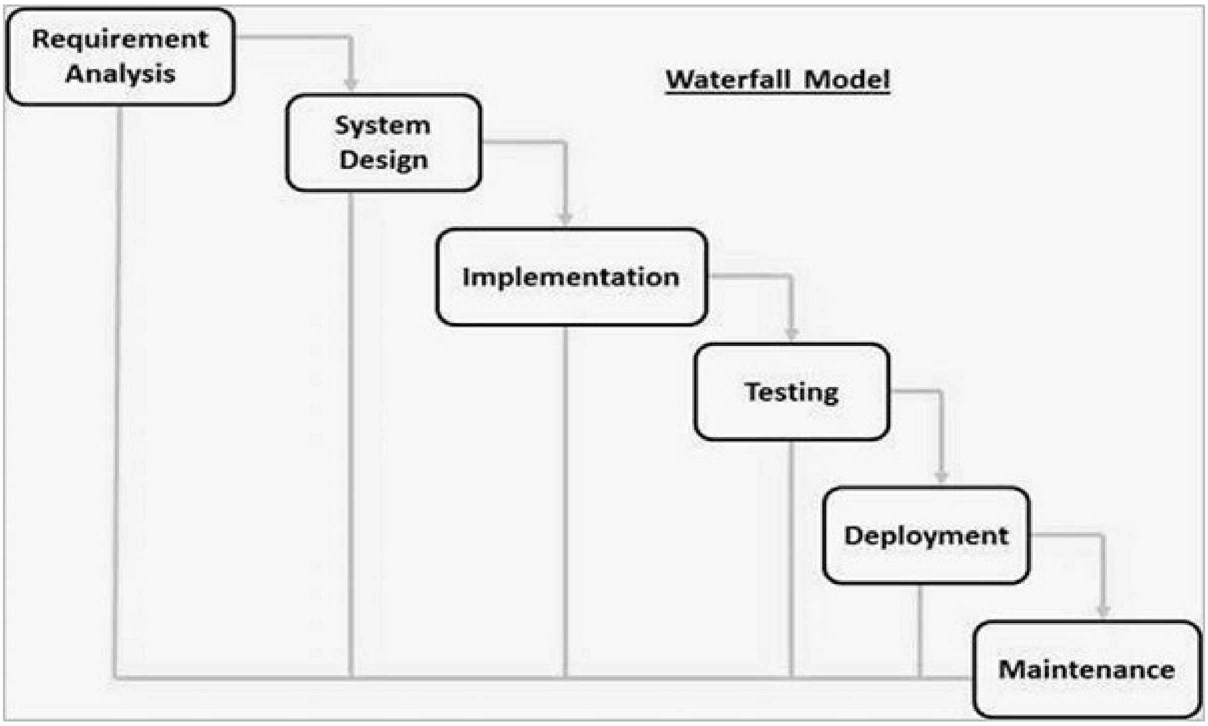


Fig 1.2 WaterFall model

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 document.

- **System Design** − The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.

- **Implementation** − With inputs from the 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.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases as shown in the Fig 1.2. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model, phases do not overlap.

Waterfall Model - Application

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are −

- Requirements are very well documented, clear and fixed.

- Product definition is stable.

- Technology is understood and is not dynamic.

- There are no ambiguous requirements.

- Ample resources with required expertise are available to support the product.
- The project is short.

Waterfall Model - Advantages

The advantages of waterfall development are that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one.

Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order.

Some of the major advantages of the Waterfall Model are as follows −

- Simple and easy to understand and use
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
- Phases are processed and completed one at a time.
- Works well for smaller projects where requirements are very well understood.
- Clearly defined stages.
- Well understood milestones.
- Easy to arrange tasks.
- Process and results are well documented.

Waterfall Model - Disadvantages

The disadvantage of waterfall development is that it does not allow much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-documented or thought upon in the concept stage.

The major disadvantages of the Waterfall Model are as follows −

- No working software is produced until late during the life cycle.
- High amounts of risk and uncertainty.
- Not a good model for complex and object-oriented projects.
- Poor model for long and ongoing projects.
- Not suitable for the projects where requirements are at a moderate to high risk of changing. So, risk and uncertainty is high with this process model.
- It is difficult to measure progress within stages.
- Cannot accommodate changing requirements.

CHAPTER 2

ANALYSIS

2.1 INTRODUCTION

The analysis phase defines the requirements of the system, independent of how these requirements will be accomplished. This phase defines the problem that the customer is trying to solve. The deliverable result at the end of this phase is a requirement document. Ideally, this document states in a clear and precise fashion what is to be built. This analysis represents the ``what" phase. The requirement document tries to capture the requirements from the customer's perspective by defining goals and interactions at a level removed from the implementation details.

2.2 REQUIREMENT SPECIFICATION

OUTPUT DESIGN

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provides a permanent copy of the results for later consultation. The various types of outputs in general are:

- External Outputs, whose destination is outside the organization.
- Internal Outputs whose destination is with in organization and they are the
- User’s main interface with the computer.
- Operational outputs whose use is purely with in the computer department.
- Interface outputs, which involve the user in communicating directly with

OUTPUT DEFINITION

The outputs should be defined in terms of the following points:

- Type of the output
- Content of the output

- Format of the output
- Location of the output
- Frequency of the output
- Volume of the output
- Sequence of the output

It is not always desirable to print or display data as it is held on a computer. It should be decided as which form of the output is the most suitable.

For Example

- Will decimal points need to be inserted
- Should leading zeros be suppressed.

Output Media:

In the next stage it is to be decided that which medium is the most appropriate for the output. The main considerations when deciding about the output media are:

- The suitability for the device to the particular application.
- The need for a hard copy.
- The response time required.
- The location of the users
- The software and hardware available.

Keeping in view the above description the project is to have outputs mainly coming under the category of internal outputs. The main outputs desired according to the requirement specification are:

The outputs were needed to be generated as a hot copy and as well as queries to be viewed on the screen. Keeping in view these outputs, the format for the output is taken from the outputs, which are currently being obtained after manual processing. The standard printer is to be used as output media for hard copies.

INPUT DESIGN

Input design is a part of overall system design. The main objective during the input design is as given below:

- To produce a cost-effective method of input.
- To achieve the highest possible level of accuracy.
- To ensure that the input is acceptable and understood by the user.

INPUT STAGES:

The main input stages can be listed as below:

- Data recording
- Data transcription
- Data conversion
- Data verification
- Data control
- Data transmission
- Data validation
- Data correction

INPUT TYPES:

It is necessary to determine the various types of inputs. Inputs can be categorized as follows:

- External inputs, which are prime inputs for the system.
- Internal inputs, which are user communications with the system.
- Operational, which are computer department's communications to the system?
- Interactive, which are inputs entered during a dialogue.

INPUT MEDIA

At this stage choice has to be made about the input media. To conclude about the input media consideration has to be given to;

- Type of input
- Flexibility of format
- Speed
- Accuracy
- Verification methods
- Rejection rates
- Ease of correction
- Storage and handling requirements
- Security
- Easy to use
- Portability

Keeping in view the above description of the input types and input media, it can be said that most of the inputs are of the form of internal and interactive. As

Input data is to be the directly keyed in by the user, the keyboard can be considered to be the most suitable input device.

ERROR AVOIDANCE

At this stage care is to be taken to ensure that input data remains accurate form the stage at which it is recorded up to the stage in which the data is accepted by the system. This can be achieved only by means of careful control each time the data is handled.

ERROR DETECTION

Even though every effort is made to avoid the occurrence of errors, still a small proportion of errors is always likely to occur, these types of errors can be discovered by using validations to check the input data.

DATA VALIDATION

Procedures are designed to detect errors in data at a lower level of detail. Data validations have been included in the system in almost every area where there is a possibility for the user to commit errors. The system will not accept invalid data. Whenever an invalid data is keyed in, the system immediately prompts the user and the user has to again key in the data and the system will accept the data only if the data is correct. Validations have been included where necessary.

The system is designed to be a user friendly one. In other words, the system has been designed to communicate effectively with the user. The system has been designed with pop-up menus.

USER INTERFACE DESIGN

It is essential to consult the system users and discuss their needs while designing the user interface:

USER INTERFACE SYSTEMS CAN BE BROADLY CLASIFIED AS:

1. User initiated interface the user is in charge, controlling the progress of the user/computer dialogue. In the computer-initiated interface, the computer selects the next stage in the interaction.
2. Computer initiated interfaces

In the computer-initiated interfaces the computer guides the progress of the user/computer dialogue. Information is displayed and the user response of the computer takes action or displays further information.

USER_INITIATED INTERGFACES

User initiated interfaces fall into two approximate classes:

1. Command driven interfaces: In this type of interface the user inputs commands or queries which are interpreted by the computer.
2. Forms oriented interface: The user calls up an image of the form to his/her screen and fills in the form. The forms-oriented interface is chosen because it is the best choice.

COMPUTER-INITIATED INTERFACES

The following computer – initiated interfaces were used:

1. The menu system for the user is presented with a list of alternatives and the user chooses one; of alternatives.
2. Questions – answer type dialog system where the computer asks question and takes action based on the basis of the users reply.

Right from the start the system is going to be menu driven, the opening menu displays the available options. Choosing one option gives another popup menu with more options. In this way every option leads the users to data entry form where the user can key in the data.

2.3 FEASIBILITY STUDY

The feasibility study is performed on the basis of various criteria and parameters that decides to make a decision whether or not to provide with the project by considering the results of the study .

Generally the feasibility is divided into three categories

1. Technical Feasibility
2. Operational Feasibility
3. Economical Feasibility

Technical feasibility refers to the ability of the process to take advantages of the current state of the technology in pursuing further improvement.

Operational feasibility refers how far the system is understandable by the users.

Economic feasibility of proposed projects to generates economic benefit.

2.3.1 TECHNICAL FEASIBILITY

Project is built using JSP, MySQL and Tomcat 7 the technical considerations of the used technologies are as follows.

JSP as Server-Side Technology

JSP Vs Active Server Pages (ASP): ASP is a similar technology from Microsoft. The advantages of JSP is twofold. First, the dynamic part is written in java, not visual Basic or other MS-specific language, so it is more powerful and easier to use. Second, it is portable to other operating systems and non-Microsoft Web servers.

JSP Vs Pure Servlets: Jsp doesn't give you anything that you couldn't in principle do with a servlets. But it is more convenient to write (and too modified!) Regular HTML than to have a zillion println statements that generate the HTML. Plus, by separating the look from the content you can put different people and different tasks: Your web page design experts can build the HTML, leaving places for your design programmers to insert the dynamic content.

JSP Vs Server-Side Includes (SSI): SSI is a widely-supported technology for including externally-defined pieces into a static Web page. JSP is better because it lets you use servlets instead of a separate program to generate that dynamic part. Besides, SSI is really only intended for simple inclusions, not for "real" programs that use form data, make database connections, and the like.

JSP Vs JavaScript: JavaScript can generate HTML dynamically on the client. This is a useful capability, but only handles situations where the dynamic information is based on the client's environment. With the exception of cookies, HTTP and form submission data is not available to JavaScript. And, since it runs on the client, JavaScript can't access server-side resources like databases, catalogues, pricing information, and the like.

JSP Static HTML: Regular HTML, of course, cannot contain dynamic information; JSP is so easy and convenient that it is quite feasible to augment HTML pages that only benefit marginally by the insertion of small amounts of dynamic data. Previously, the cost of using dynamic data would preclude its use in all but the most valuable instances.

Web server: Tomcat.

Apache tomcat implements a new servlets container (called Catalina) that is based on completely new architecture. The 5.x releases implement the Servlets 2.3 and JSP 2 specifications.

Apache Tomcat. Apache Tomcat 5.5.is a refactoring of Apache Tomcat 5.5 and contains significant enhancements, including:

- JMX based administration features
- JSP and struts-based administration web application
- New Coyote connector (HTTP/1.1/AJP1.3 and JNI support)
- Rewritten jasper JSP page compiler
- Performance and memory efficiency improvements
- Enhanced manager application support for integration with development tools
- Custom Ant tasks to interact with the manager application directly from build .xml scripts

What is MySQL?

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by MySQL AB. MySQL AB is a commercial company, founded by the MySQL developers. It is a second-generation Open Source company that unites Open Source values and methodology with a successful business model.

- MySQL is a database management system.

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

- MySQL is a relational database management system.

A relational database stores data in separate tables rather than putting all the data in one big storeroom. This adds speed and flexibility. The SQL part of "MySQL" stands for "Structured Query Language." SQL is the most common standardized language used to access databases and is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, "SQL-92" refers to the standard released in 1992, "SQL:1999" refers to the standard released in 1999, and "SQL:2003" refers to the current version of the standard. We use the phrase "the SQL standard" to mean the current version of the SQL Standard at any time.

- MySQL software is Open Source.

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), http://www.fsf.org/licenses/, to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed

MySQL code into a commercial application, you can buy a commercially licensed version from us. See the MySQL Licensing Overview for more information (http://www.mysql.com/company/legal/licensing/).

2.3.2 OPERATIONAL FEASIBILITY

The system is operationally feasible because it doesn’t need any sophisticated training. It is organized into menus so the user of the system can easily interact with the system. The sight map allows the user for searching information easily.

2.3.3 ECONOMICAL FEASIBILITY

Our application is economic feasible because the cost of the production of the application is very low. Because Tomcat and java are free distributions and windows and access are available lower cost remaining databases.

The application reduces efforts, communications cost and the time to be spend by the user to communicate with the staff of the university. So, our project is purely feasible for various users.

2.4 SOFTWARE REQUIREMENTS

Server:

Operating system	Windows xp/ any server
Network layer	TCP/IP
Web server layer	Tomcat
Database	MySQL

Designer's system:

Operating system	Windows xp/ any server
Languages	Java2 with JDK
Database configuration tool	MySQL Connector
Development tool	eclipse
Web server	Tomcat
Browser	Internet explorer 6 or compatible browsers

System to run application:

Operating system	Windows xp/ any server
Internet connection	Minimum of 56 kbps dial up network
Browser	Internet explorer 6 or compatible browsers

Hard ware requirements:

Server:

- Pentium with 2 GHz or better and a motherboard with built-in network card of 100 mbps.
- 256 MB RAM minimum
- 800 * 600 resolution; 16-bit (1024 * 768, 24- bit is recommended)
- 15” monitor (17” monitor recommended for best results)
- Minimum of 1 GB hard disk space is required.
- CD - Drive 52x.

Designer’s system:

- Pentium with 2 GHz or better and a motherboard with built-in network card of 100 mbps.
- 1GB RAM minimum.
- 800 * 600 resolution; 16-bit (1024 * 768, 24- bit is recommended)
- 15” monitor (17” monitor recommended for best results)
- Minimum of 2 GB hard disk space is required.
- CD - Drive 52x.

System to run application:

- Pentium with 100MHz or better
- 64 MB RAM minimum
- 800 * 600 resolution; 16-bit (1024 * 768, 24- bit is recommended)
- Modem of 56 Kbps
- Minimum of 2 GB hard disk space is recommended

CHAPTER 3

SYSTEM DESIGN

3.1 INTRODUCTION

Design is the first step in the development in the phase for any engineering product or System. Software design is the main step of the software engineering process and is applied regardless of the software process model that is used. Once the software require, design is the first step in the development phase for any engineering product or elements has been analyzed and specified software design is the first step of activities.

Software requirements, manifested by the data, functional and behavioural models feed the design step products a data design, an architectural design, an interface design and a procedural design.

The data design transforms the information domain model created during analysis into the data structures that will be required to implement the software. The data objects and relationships defined in the E-R diagram and the data content available in data dictionary provide the basis for data design. The architectural design defines the relationship among major structural elements of the program. This design representation can be derived from the analysis model and the interaction of the subsystem defined with in the analysis model.

The interface design describes how the software communicates within itself, to systems that interoperate with it and with human who use it. Design is the place where quality is emphasized in software development. Design is the only way that we ca accurately translate customer's requirements into a finished software product or system. Software design serves as a foundation for all software maintain steps that follow.

3.2 DESIGN PROCESS

Following are the three characteristics that serve as a guide for the evolution of the good design.

- The design must implement all of the explicit requirements contained in the analysis model, and it must accommodate all of the explicit requirements desired by the customer.

- The design must be readable, understandable guide for those who generate code and for those who test and subsequently maintain the software.

- The design should provide a complete picture of the software, addressing the data, functional and behavioural domains from an implementation perspective.

3.3 MODULE DESCRIPTION

1. **Admin Login:**
   - User has to login to the system to access various helping pages through which user can ask queries to the system with the help of bot.

2. **Bot Chat:**
   - User can chat with the bot it implies as if enquiring to the college person about college related activities.

3. **Response:**
   - The system gives the corresponding response.

3.4 NORMALIZATION

It is a process of converting a relation to a standard form. The process is used to handle the problems that can arise due to data redundancy i.e. repetition of data in the database, maintain data integrity as well as handling problems that can arise due to insertion, updating, deletion anomalies.

Insertion anomaly: Inability to add data to the database due to absence of other data.

Deletion anomaly: Unintended loss of data due to deletion of other data.

Update anomaly: Data inconsistency resulting from data redundancy and partial update

Normal Forms: These are the rules for structuring relations that eliminate anomalies.

FIRST NORMAL FORM:

A relation is said to be in first normal form if the values in the relation are atomic for every attribute in the relation. By this we mean simply that no attribute value can be a set of values or, as it is sometimes expressed, a repeating group.

SECOND NORMAL FORM: A relation is said to be in second Normal form is it is in first normal form and it should satisfy any one of the following rules.

1) Primary key is a not a composite primary key
2) No non key attributes are present
3) Every non key attribute is fully functionally dependent on full set of primary key.

THIRD NORMAL FORM:

A relation is said to be in third normal form if their exits no transitive dependencies.

Transitive Dependency: If two non-key attributes depend on each other as well as on the primary key then they are said to be transitively dependent.

The above normalization principles were applied to decompose the data in multiple tables thereby making the data to be maintained in a consistent state as shown in Fig. 3.1

3.5 DATA DICTIONARY

A data dictionary is a collection of descriptions of the data objects or items in a data model for the benefit of programmers and others who need to refer to them.

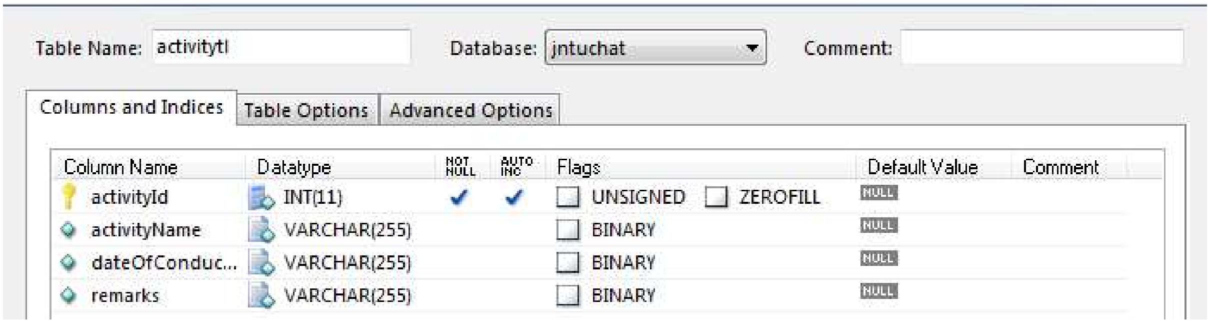


Table Name: activitytl   Database: jntuchat   Comment:

Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
activityId	INT(11)	✓	✓	□ UNSIGNED □ ZEROFILL	NULL	
activityName	VARCHAR(255)			□ BINARY	NULL	
dateOfConduc...	VARCHAR(255)			□ BINARY	NULL	
remarks	VARCHAR(255)			□ BINARY	NULL	

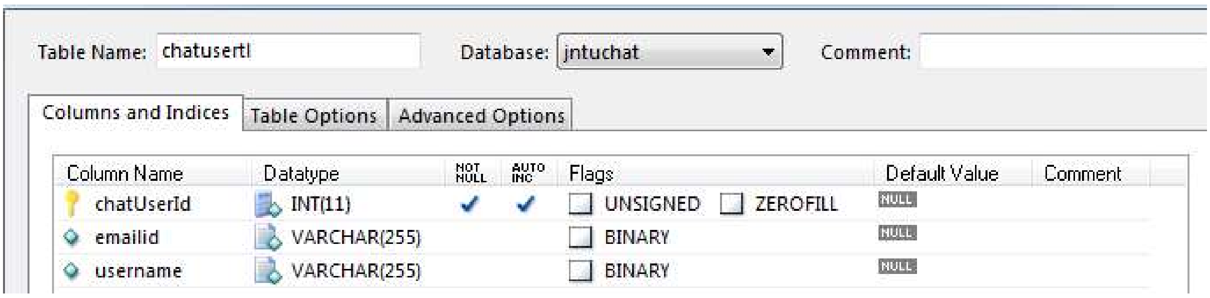


Table Name: chatusertl   Database: jntuchat   Comment:

Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
chatUserId	INT(11)	✓	✓	☐ UNSIGNED ☐ ZEROFILL	NULL	
emailid	VARCHAR(255)			☐ BINARY	NULL	
username	VARCHAR(255)			☐ BINARY	NULL	

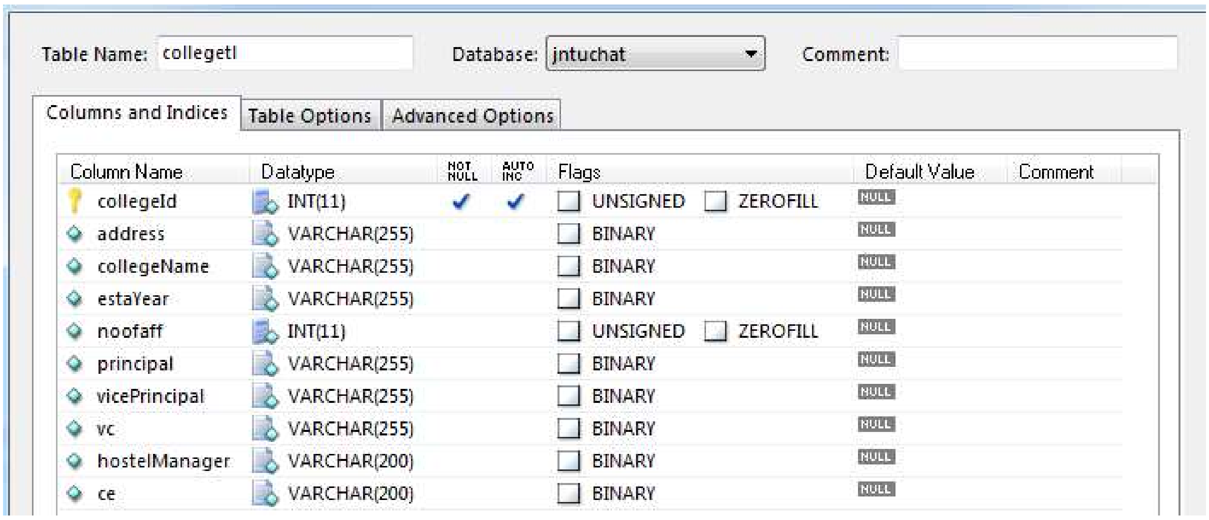


Table Name: collegetl   Database: jntuchat   Comment:

Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
collegeId	INT(11)	✓	✓	☐ UNSIGNED ☐ ZEROFILL	NULL	
address	VARCHAR(255)			☐ BINARY	NULL	
collegeName	VARCHAR(255)			☐ BINARY	NULL	
estaYear	VARCHAR(255)			☐ BINARY	NULL	
noofaff	INT(11)			☐ UNSIGNED ☐ ZEROFILL	NULL	
principal	VARCHAR(255)			☐ BINARY	NULL	
vicePrincipal	VARCHAR(255)			☐ BINARY	NULL	
vc	VARCHAR(255)			☐ BINARY	NULL	
hostelManager	VARCHAR(200)			☐ BINARY	NULL	
ce	VARCHAR(200)			☐ BINARY	NULL	

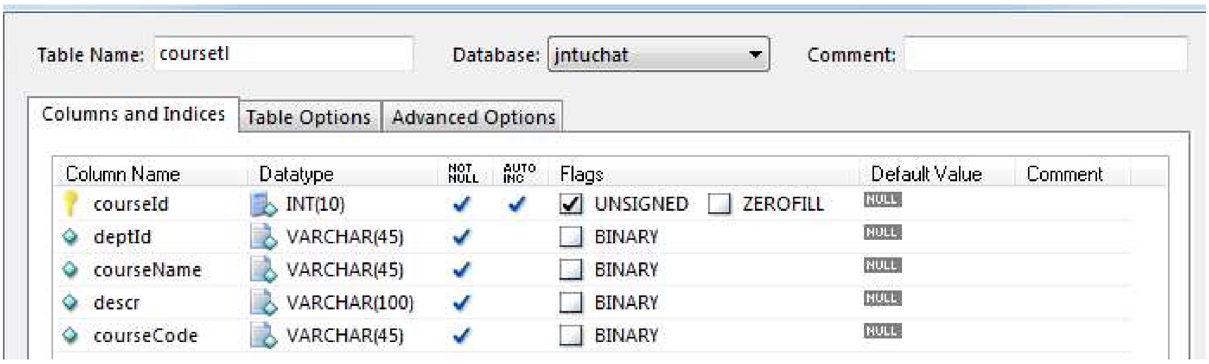


Table Name: coursetl   Database: jntuchat   Comment:

Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
courseId	INT(10)	✓	✓	☑ UNSIGNED ☐ ZEROFILL	NULL	
deptId	VARCHAR(45)	✓		☐ BINARY	NULL	
courseName	VARCHAR(45)	✓		☐ BINARY	NULL	
descr	VARCHAR(100)	✓		☐ BINARY	NULL	
courseCode	VARCHAR(45)	✓		☐ BINARY	NULL	

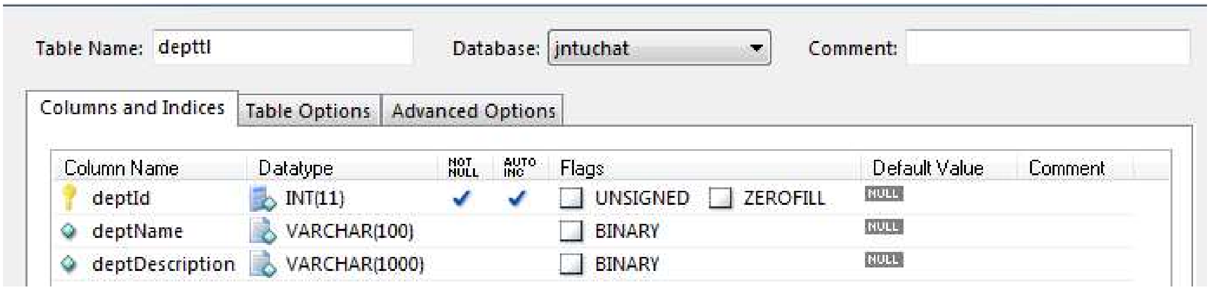


Table Name: depttl   Database: jntuchat   Comment:

Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
deptId	INT(11)	✓	✓	☐ UNSIGNED ☐ ZEROFILL	NULL	
deptName	VARCHAR(100)			☐ BINARY	NULL	
deptDescription	VARCHAR(1000)			☐ BINARY	NULL	

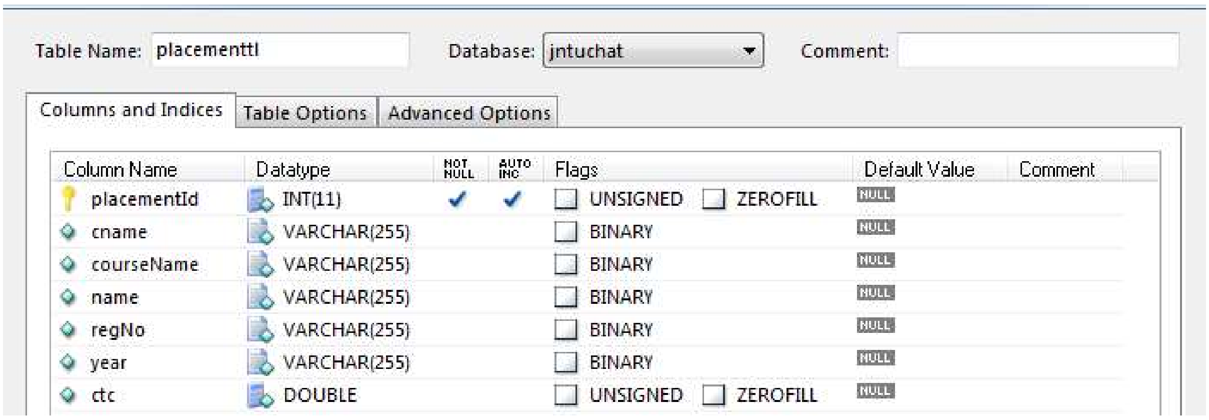


Table Name: placementtl Database: jntuchat Comment:

Columns and Indices | Table Options | Advanced Options

Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
placementId	INT(11)	✓	✓	□ UNSIGNED □ ZEROFILL	NULL	
cname	VARCHAR(255)			□ BINARY	NULL	
courseName	VARCHAR(255)			□ BINARY	NULL	
name	VARCHAR(255)			□ BINARY	NULL	
regNo	VARCHAR(255)			□ BINARY	NULL	
year	VARCHAR(255)			□ BINARY	NULL	
ctc	DOUBLE			□ UNSIGNED □ ZEROFILL	NULL	

Fig. 3.1 Data Dictionaries

3.6 E - R DIAGRAM

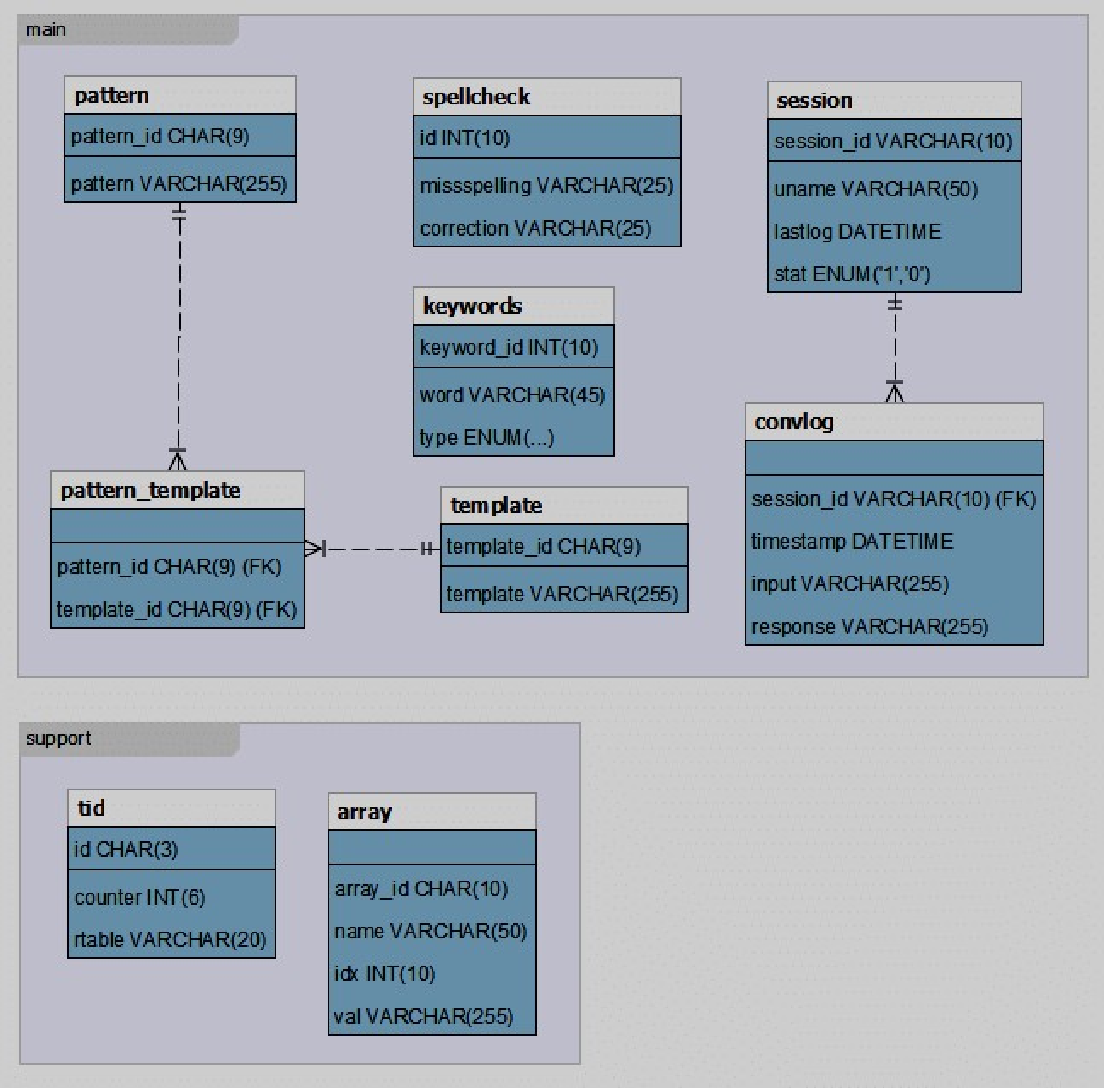


Fig. 3.2 ER Diagram

An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database. Let’s have a look at a simple ER diagram to understand this concept.

3.7 DATA FLOW DIAGRAMS

A data flow diagram is graphical tool used to describe and analyze movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams as shown in Fig.3.3 shows the actual implements and movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams. Using two familiar notations Yourdon, Gane and Sarson notation develops the data flow diagrams. Each component in a DFD is labeled with a descriptive name. Process is further identified with a number that will be used for identification purpose. The development of DFD’S is done in several levels. Each process in lower level diagrams can be broken down into a more detailed DFD in the next level. The lop-level diagram is often called context diagram. It consists a single process bit, which plays vital role in studying the current system. The process in the context level diagram is exploded into other process at the first level DFD.

The idea behind the explosion of a process into more process is that understanding at one level of detail is exploded into greater detail at the next level. This is done until further explosion is necessary and an adequate amount of detail is described for analyst to understand the process.

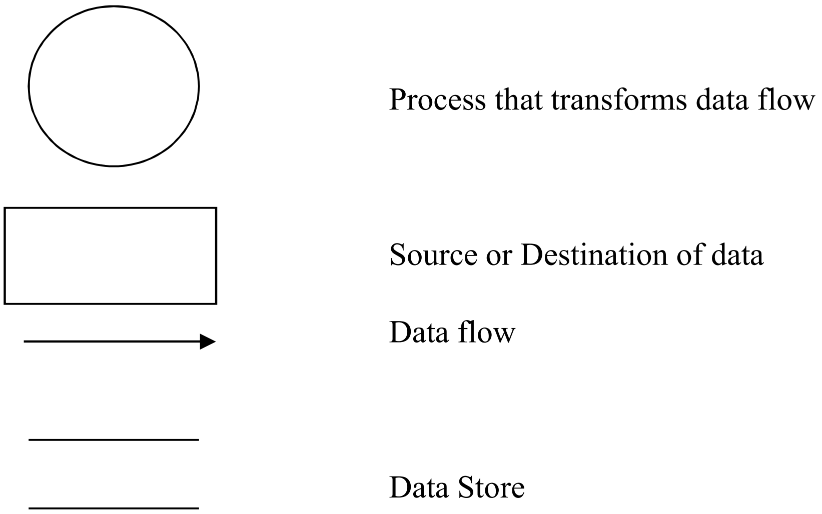
Larry Constantine first developed the DFD as a way of expressing system requirements in a graphical from, this lead to the modular design.

A DFD is also known as a “bubble Chart” has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

DFD SYMBOLS:

In the DFD, there are four symbols

- ✓ A square defines a source(originator) or destination of system data
- ✓ An arrow identifies data flow. It is the pipeline through which the information flows
- ✓ A circle or a bubble represents a process that transforms incoming data flow into outgoing data flows.
- ✓ An open rectangle is a data store, data at rest or a temporary repository of data


Process that transforms data flow
Source or Destination of data
Data flow
Data Store

RULES GOVERNING THE DFD’S

PROCESS

- ✓ No process can have only outputs.
- ✓ No process can have only inputs. If an object has only inputs than it must be a sink.
- ✓ A process has a verb phrase label.

DATA STORE

- ✓ Data cannot move directly from one data store to another data store, a process must move data.
- ✓ Data cannot move directly from an outside source to a data store, a process, which receives, must move data from the source and place the data into data store
- ✓ A data store has a noun phrase label.

SOURCE OR SINK

The origin and /or destination of data.

- ✓ Data cannot move direly from a source to sink it must be moved by a process
- ✓ A source and /or sink has a noun phrase land

DATA FLOW

- ✓ A Data Flow has only one direction of flow between symbols. It may flow in both directions between a process and a data store to show a read before an update. The later is usually indicated however by two separate arrows since these happen at different type.
- ✓ A join in DFD means that exactly the same data comes from any of two or more different processes data store or sink to a common location.
- ✓ A data flow cannot go directly back to the same process it leads. There must be at least one other process that handles the data flow produce some other data flow returns the original data into the beginning process.

✓ A Data flow to a data store means update (delete or change).

✓ A data Flow from a data store means retrieve or use.

✓ A data flow has a noun phrase label more than one data flow noun phrase can appear on a single arrow as long as all of the flows on the same arrow move together as one package.

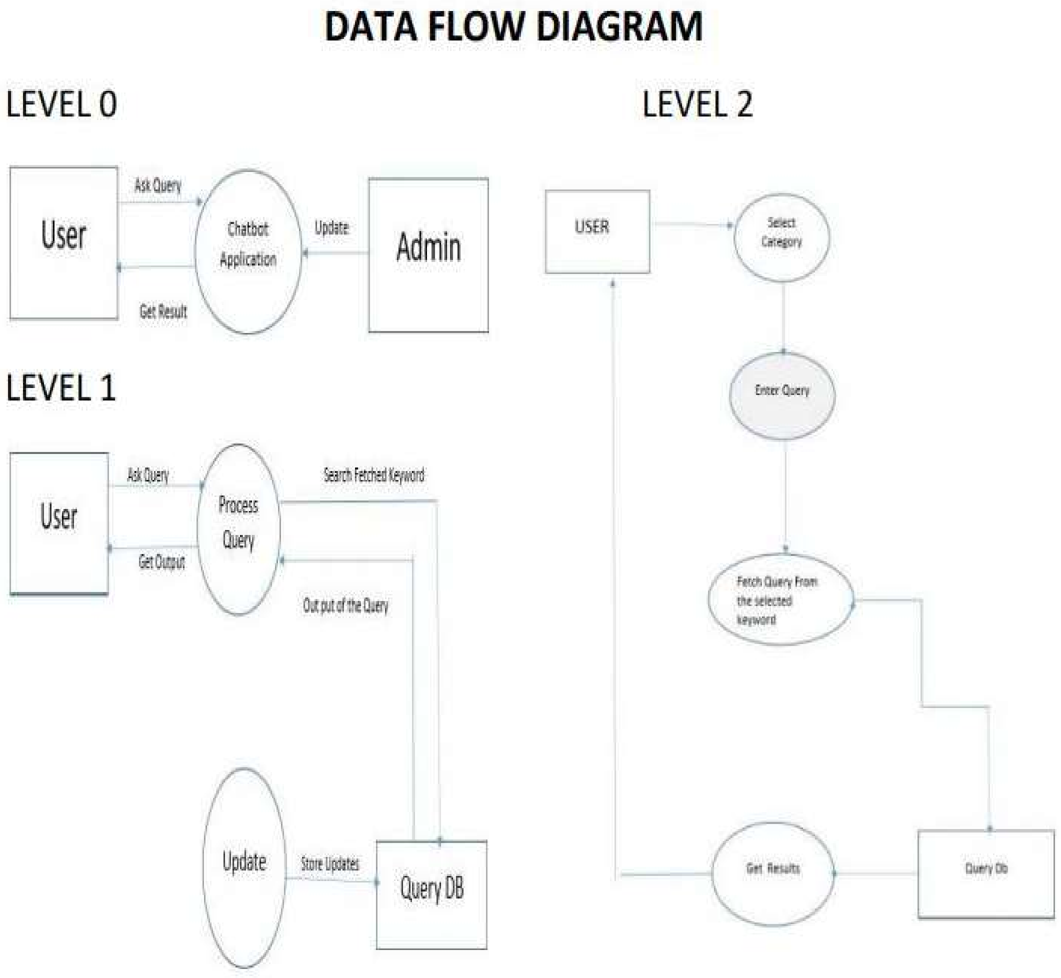


Fig. 3.3 Data Flow Diagram

3.8 UML DIAGRAMS

3.8.1 SEQUENCE DIAGRAM

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

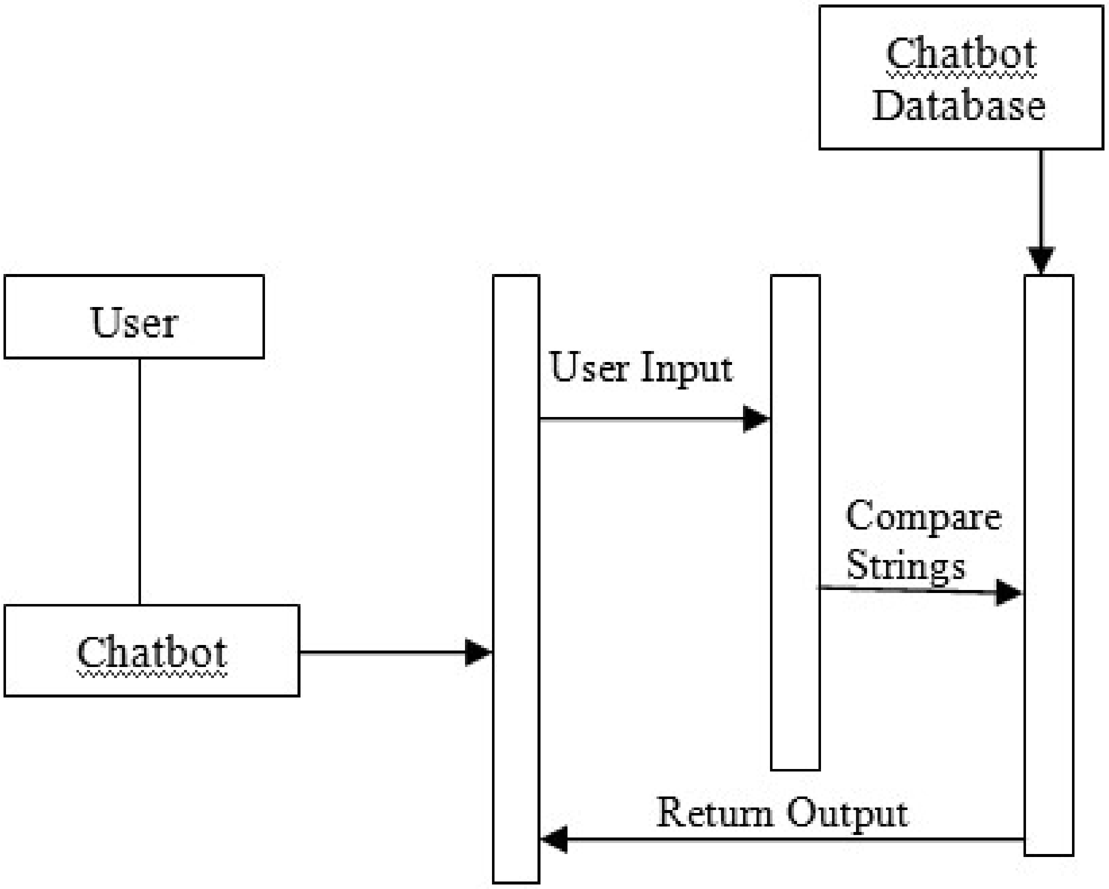


Fig.3.4 Sequence Diagram

A shown in the Fig. 3.4 represents the interactions between different entities of the chatbot system.

3.8.2 ACTIVITY DIAGRAM

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart as shown in Fig 3.5 to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc

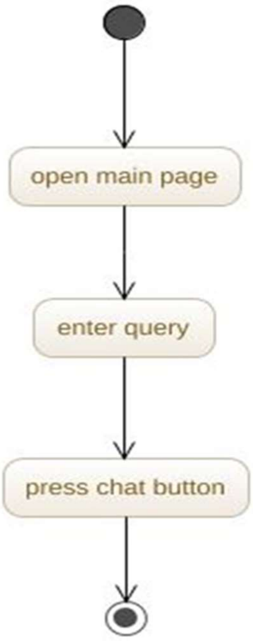


Fig 3.5 Activity Diagram

CHAPTER 4
IMPLEMENTATION

4.1 INTRODUCTION

This section describes about the implementation of the Safe application and the details of how to access this control from any application.

Implementation is the process of assuring that the information system is operational and then allowing users take over its operation for use and evaluation. Implementation includes the following activities:

Obtaining and installing the system hardware.

Installing the system and making it run on its intended hardware.

Providing user access to the System.

Creating and updating the Data Base.

Documenting the System for its users and for those who will be responsible for maintaining it in the future work.

Making arrangements to support the users as the system is used.

Transferring ongoing responsibility for the system from its developers to the operations or maintenance part.

Evaluating the operation and use of the system.

The most visible component of implementation process is the system conversion.

4.2 METHODS OF IMPLEMENTATION

The four basic methods of implementation are:

1. Parallel system method.

2. Direct cut over method.

3. Pilot system method.

4. Phase in method.

Parallel system method:

In this method the old system is operated with the new one. It offers great security. The old system can take over if errors are found in new system or if some usage problems occur.

Direct cut-over method:

In this system, the new one replaces the old system. This makes organization to fully rely on new system.

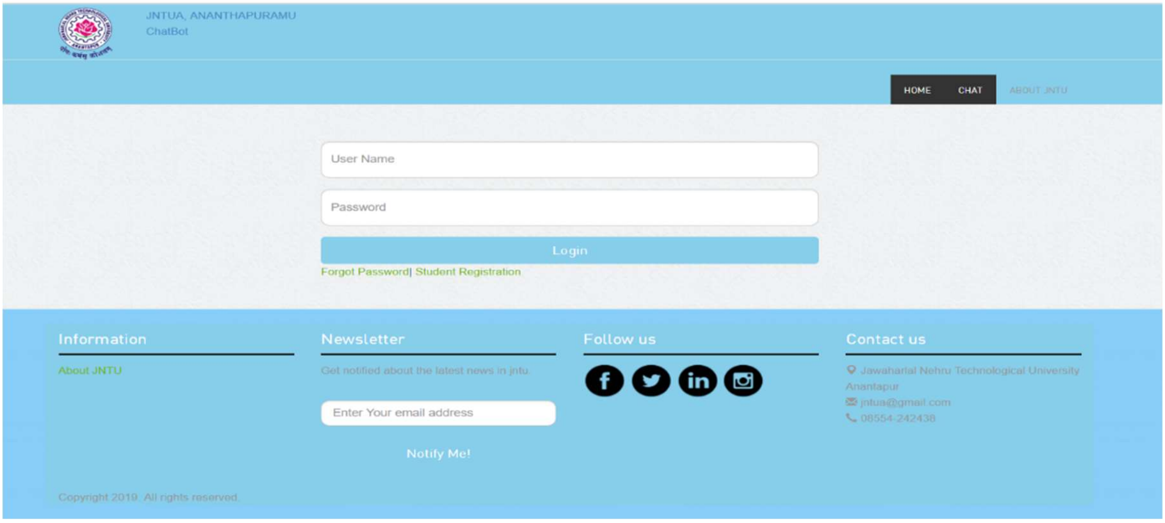
Pilot system method:

This means that introducing new system to a small part of organization, expanding its use once once it is know to operate properly there. This method provides experience and live test and live test before implementation.

Phase-in method:

In this, the system is implemented gradually across all users. It allows some users to take advantage of the system each and allow in these without necessary use of resources. After completing the testing, the system is installed in organization using any of the above methods.

4.3 SCREEN SHOTS



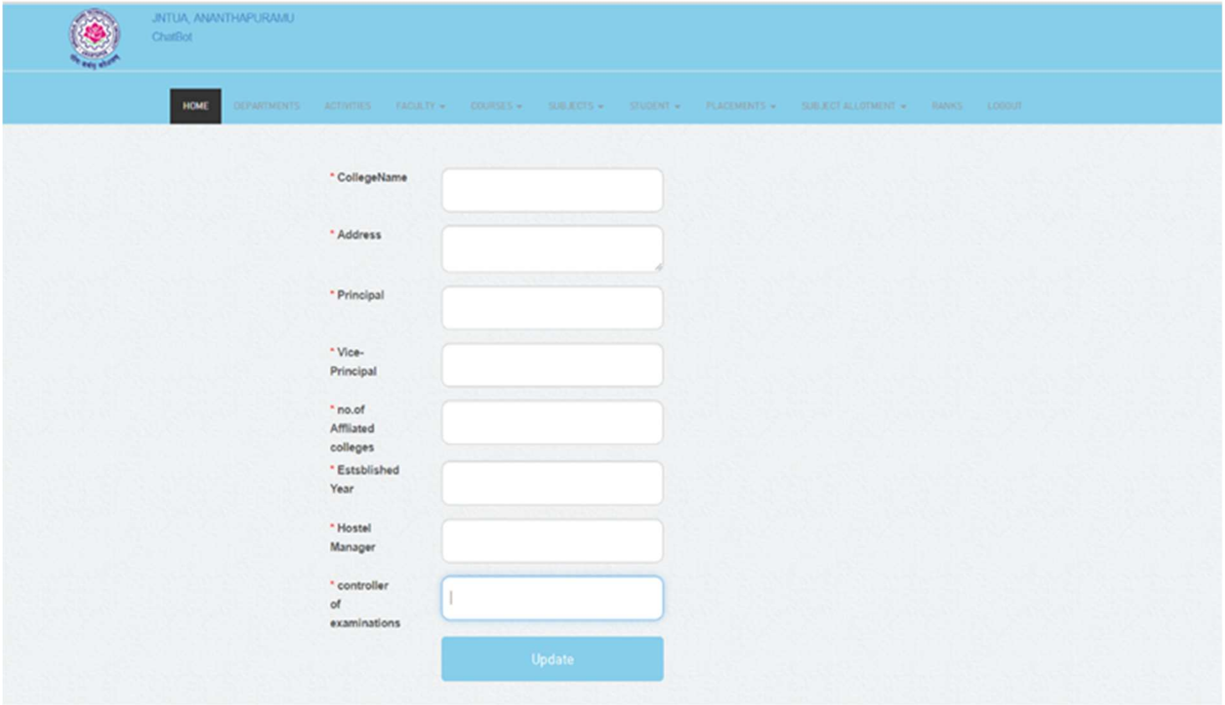
Get notified about the latest news in jntu.

Jawaharlal Nehru Technological University Anantapur
jntua@gmail.com
08554-242438

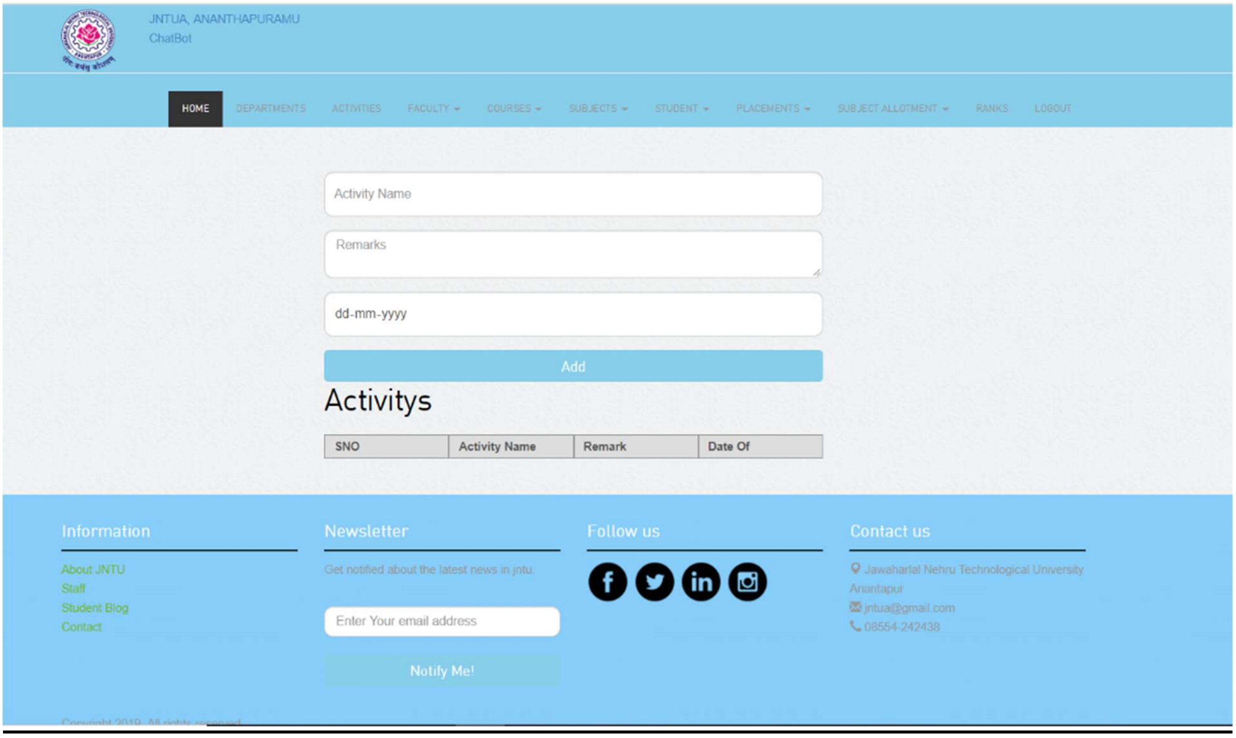
Copyright 2019. All rights reserved.

This page is for admin login

The following are the pages for the admin to enter the data:



This page is used to enter the college details



Activitys

SNO	Activity Name	Remark	Date Of

Information

About JNTU
Staff
Student Blog
Contact

Newsletter

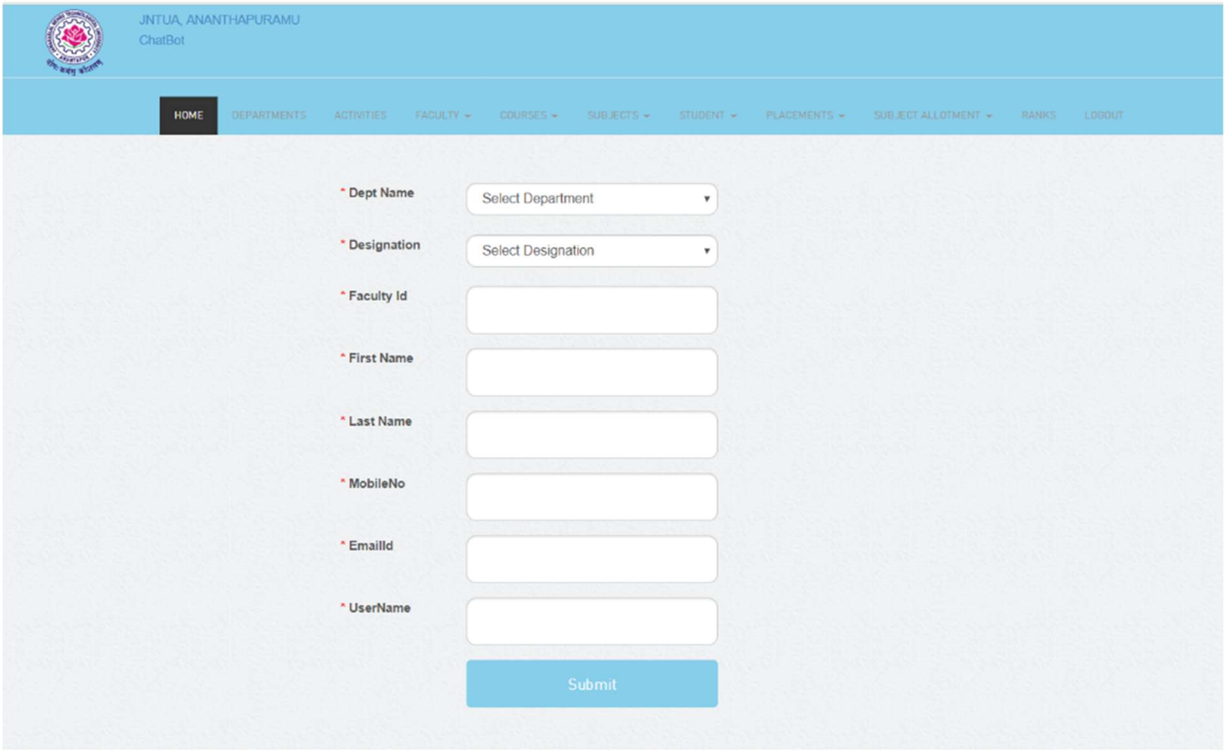
Get notified about the latest news in jntu.

Follow us

Contact us

Jawaharlal Nehru Technological University Anantapur
jntua@gmail.com
08554-242438

This page is required to enter any activity details



* Dept Name: Select Department

* Designation: Select Designation

* Faculty Id

* First Name

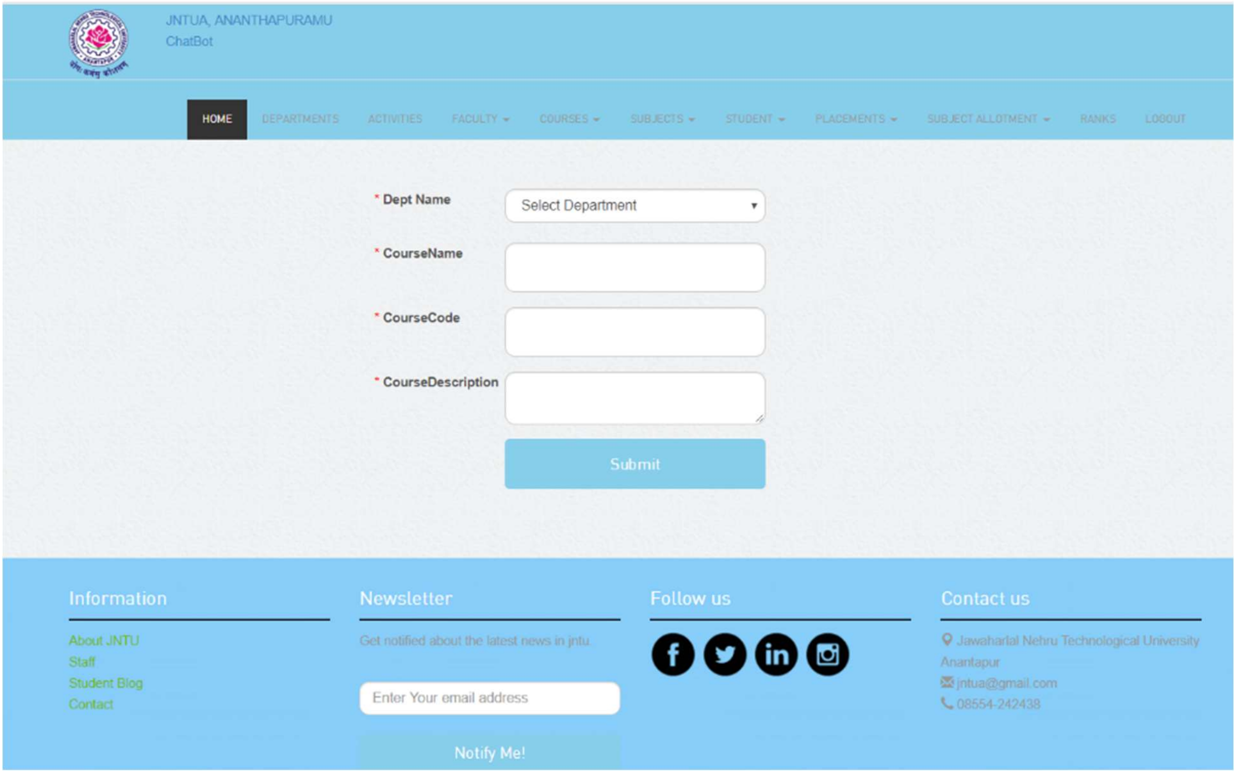
* Last Name

* MobileNo

* EmailId

* UserName

This page is used to enter the faculty details



* Dept Name: Select Department

* CourseName

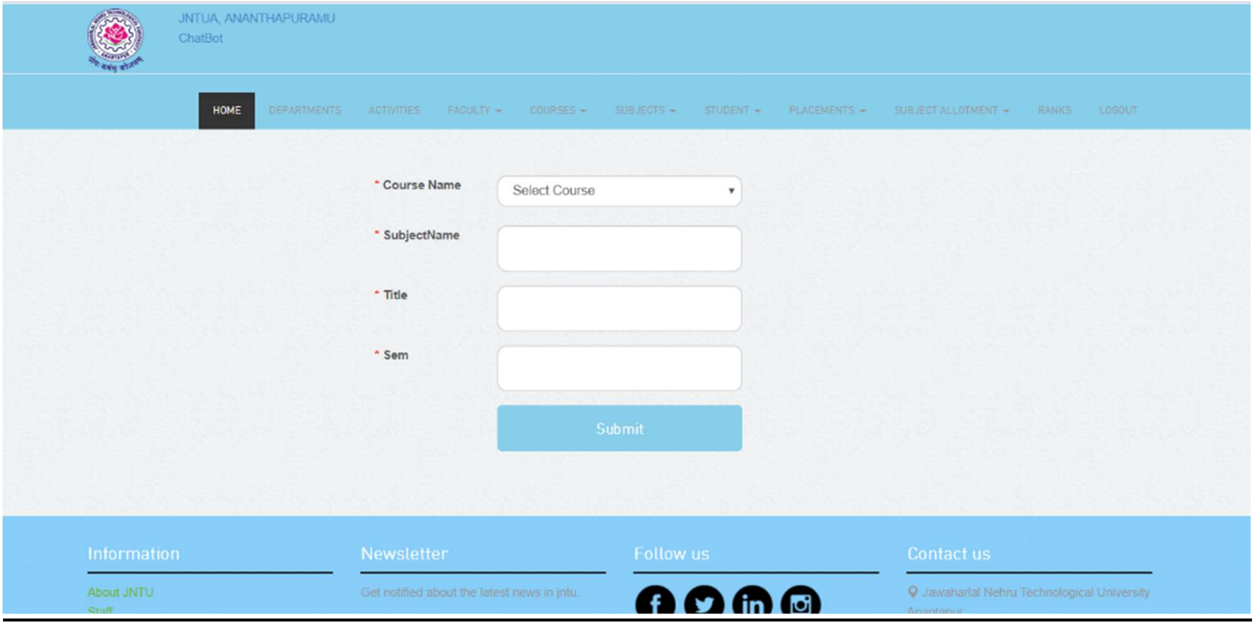
* CourseCode

* CourseDescription

Get notified about the latest news in jntu.

Jawaharlal Nehru Technological University
Anantapur
jntua@gmail.com
08554-242438

This page is used to enter Department details



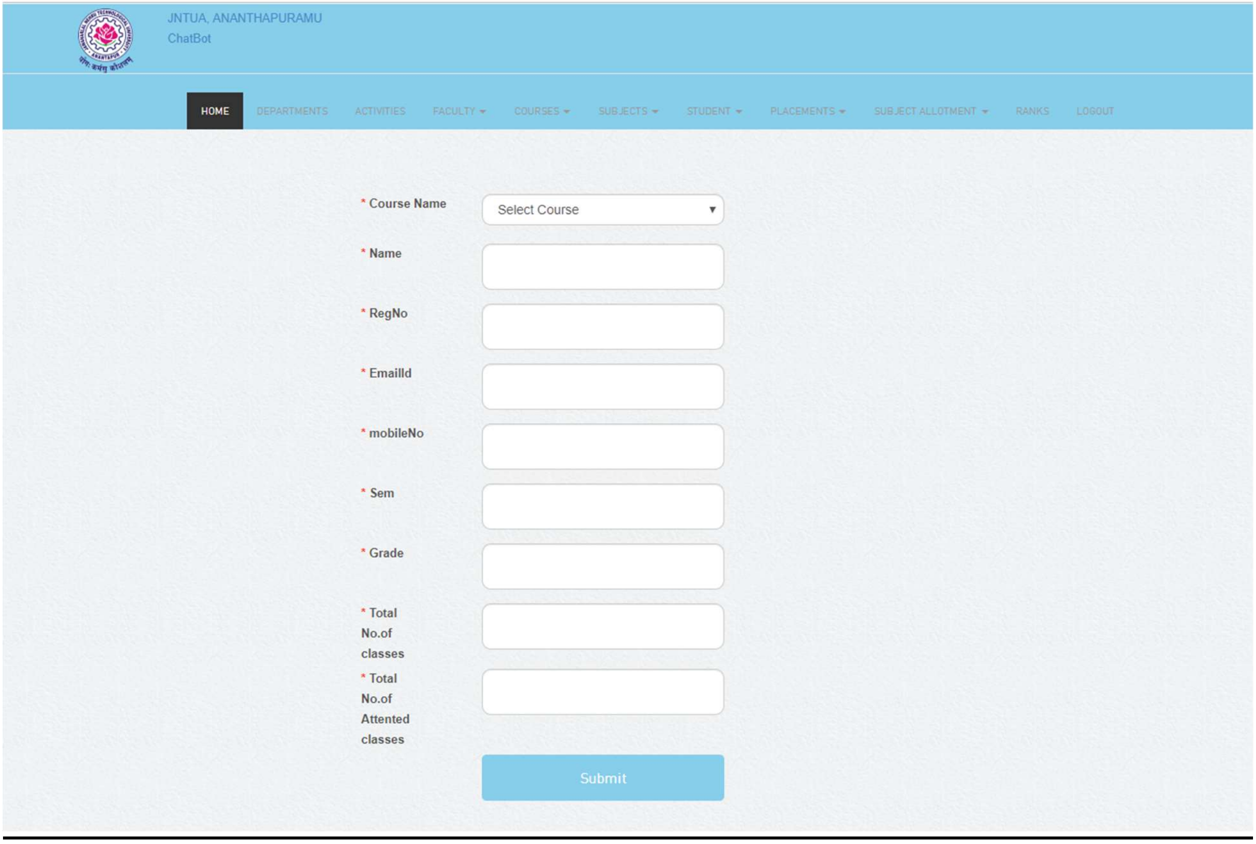
* Course Name: Select Course

* SubjectName

* Title

* Sem

This page is used to enter Subject details



* Course Name: Select Course

* Name

* RegNo

* EmailId

* mobileNo

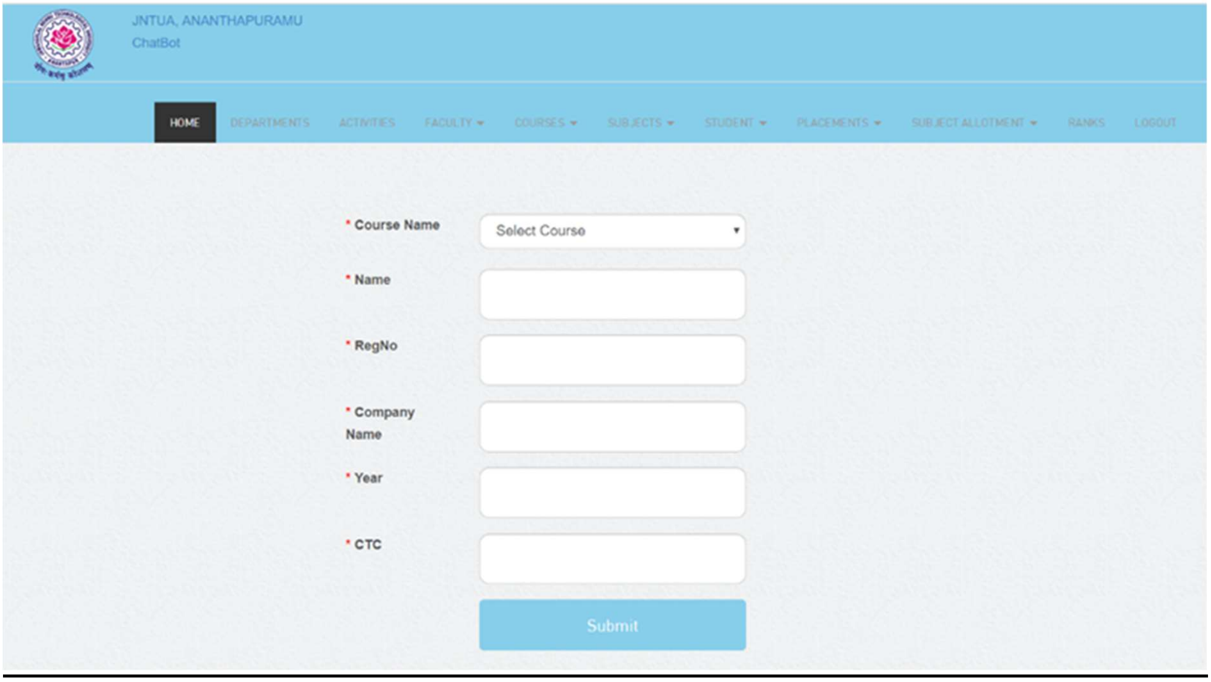
* Sem

* Grade

* Total No.of classes

* Total No.of Attented classes

This page is used to enter Student details



* Course Name: Select Course ▾

* Name:

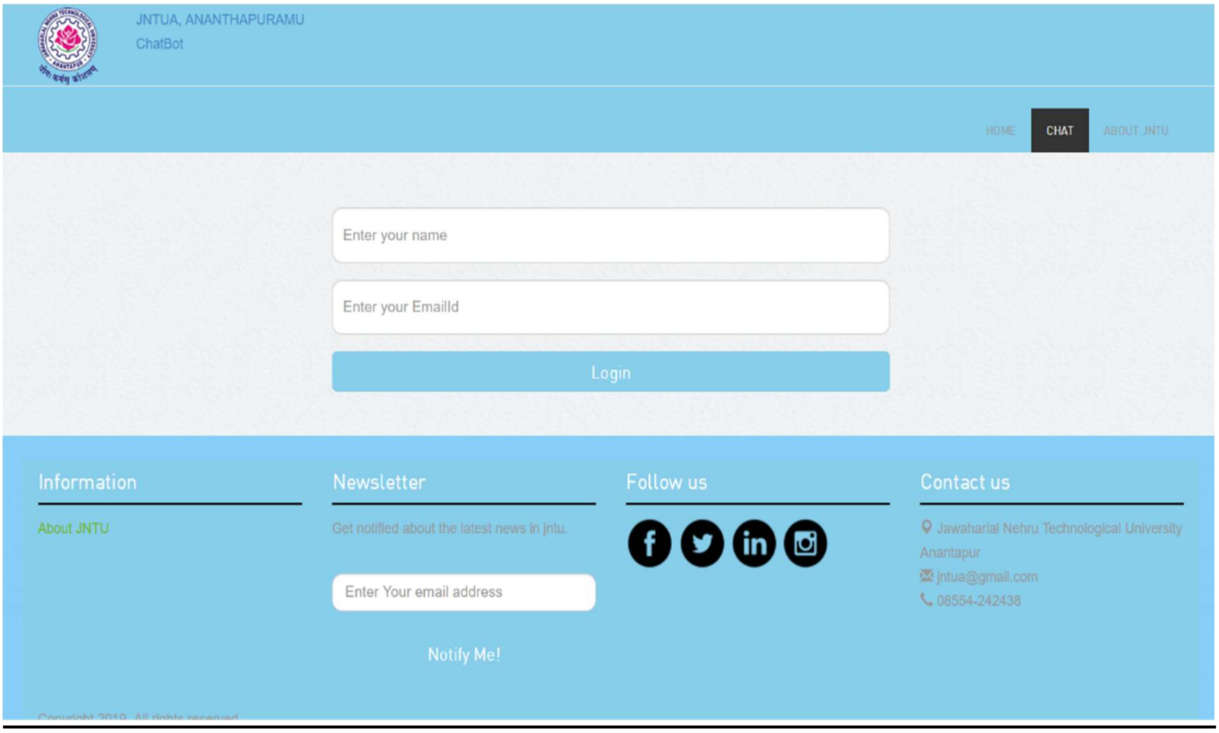
* RegNo:

* Company Name:

* Year:

* CTC:

This page is used to enter Placements details



Enter your name

Enter your EmailId

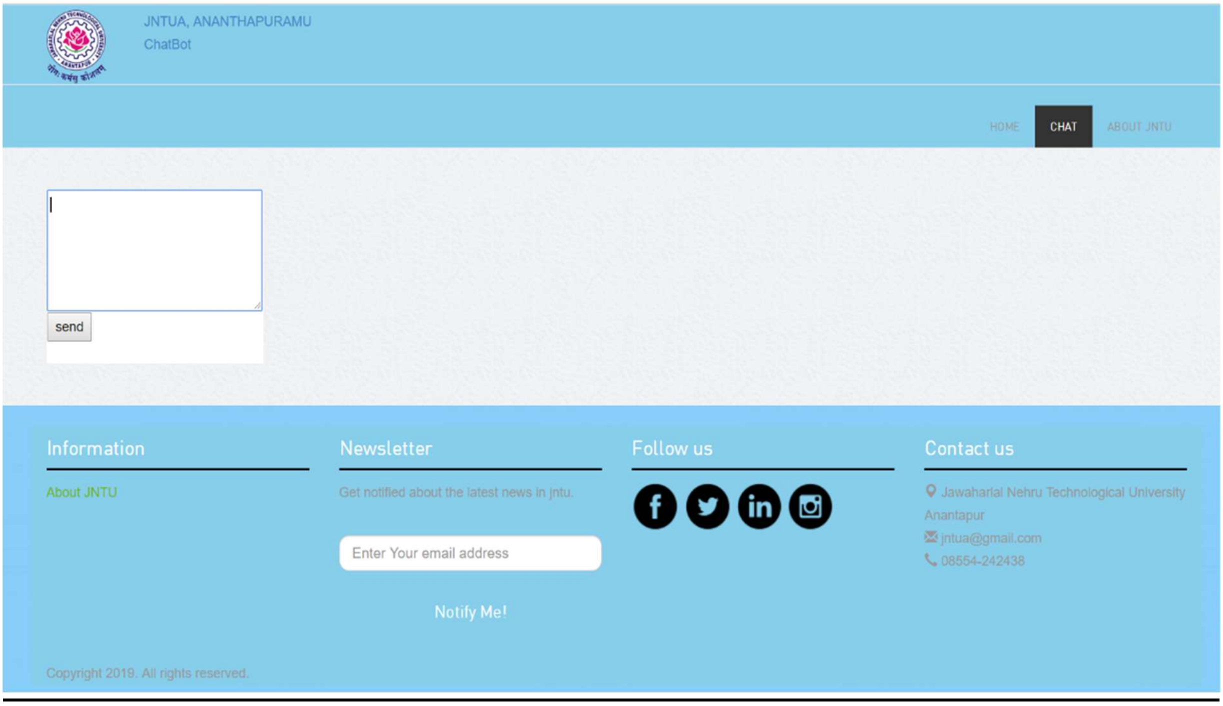
Get notified about the latest news in jntu.

Enter Your email address

Jawaharlal Nehru Technological University
Anantapur
jntua@gmail.com
08554-242438

Copyright 2019. All rights reserved

This page shows the user entering their details

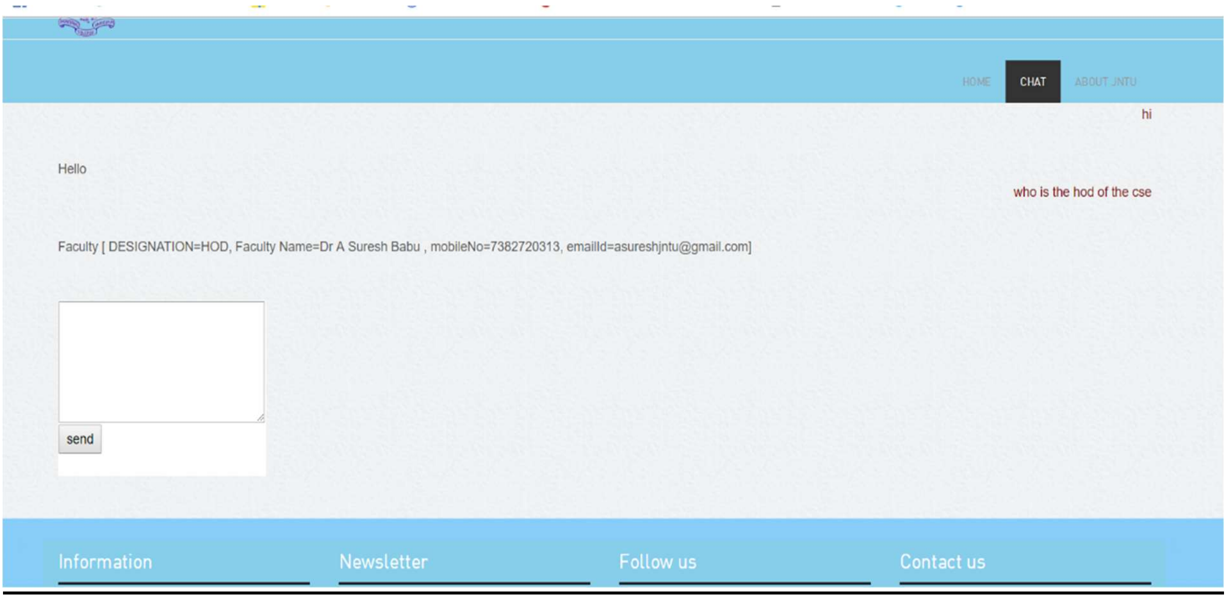


Get notified about the latest news in jntu.

Jawaharlal Nehru Technological University
Anantapur
jntua@gmail.com
08554-242438

Copyright 2019. All rights reserved.

This page displays the text area for chatting



hi

Hello

who is the hod of the cse

Faculty [DESIGNATION=HOD, Faculty Name=Dr A Suresh Babu , mobileNo=7382720313, emailId=asureshjntu@gmail.com]

This page shows how the system generates the result

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

The software engineering process can be viewed as a spiral. Initially system engineering defines the role of software and leads to software requirement analysis where the information domain, functions, behavior, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to design and finally to coding.

A strategy for software testing may also be viewed in the context of the spiral. Unit testing begins at the vertex of the spiral and concentrates on each unit of the software as implemented in source code. Testing progress by moving outward along the spiral to integration testing, where the focus is on the design and the construction of the software architecture. Talking another turn on outward on the spiral we encounter validation testing where requirements established as part of software requirements analysis are validated against the software that has been constructed. Finally, we arrive at system testing, where the software and other system elements are tested as a whole.

5.2 TYPES OF TESTING

UNIT TESTING

Unit testing focuses verification effort on the smallest unit of software design, the module. The unit testing, we have is white box oriented and some modules the steps are conducted in parallel. Unit Testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class. (Some treat a module of an application as a unit. This is to be discouraged as there will probably be many individual units within that module.) Unit testing frameworks, drivers, stubs, and mock/ fake objects are used to assist in unit testing.

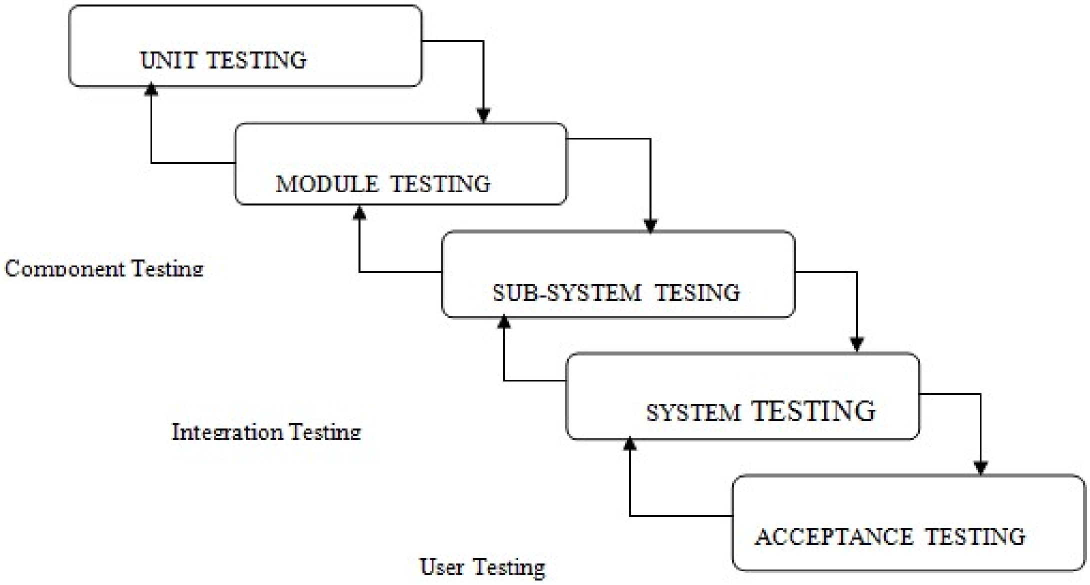


Fig. 5.1 Testing Strategy

INTEGRATION TESTING

Integration Testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.

FUNCTIONAL TESTING

Functional Testing is a type of software testing whereby the system is tested against the functional requirements/specifications.

Functions (or features) are tested by feeding them input and examining the output. Functional testing ensures that the requirements are properly satisfied by the application. This type of testing is not concerned with how processing occurs, but rather, with the results of processing. It simulates actual system usage but does not make any system structure assumptions.

During functional testing, Black box testing technique is used in which the internal logic of the system being tested is not known to the tester.

Functional testing is normally performed during the levels of System Testing and Accept Testing.

Typically, functional testing involves the following steps:

- Identify functions that the software is expected to perform.
- Create input data based on the function's specifications.
- Determine the output based on the function's specifications.
- Execute the test case.
- Compare the actual and expected outputs.

Functional testing is more effective when the test conditions are created directly from user/business requirements. When test conditions are created from the system documentation (system requirements/ design documents), the defects in that documentation will not be detected through testing and this may be the cause of end-user's wrath when they finally use the software.

SYSTEM TESTING

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

WHITE BOX TESTING

This type of testing ensures that

- All independent paths have been exercised at least once
- All logical decisions have been exercised on their true and false sides
- All loops are executed at their boundaries and within their operational bounds
- All internal data structures have been exercised to assure their validity.

To follow the concept of white box testing we have tested each form. we have created independently to verify that Data flow is correct, All conditions are exercised to check their validity, All loops are executed on their boundaries.

BLACKBOX TESTING

BLACK BOX TESTING, also known as Behavioral Testing, is a software testing method in which the internal structure/design/implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional. This method is named so because the software program, in the eyes of the tester, is like a black box; inside which one cannot see.

5.3. UNIT TESTING

Unit testing is usually conducted as a part of combined code and unit test phase of software lifecycle, although it is not uncommon for coding and unit testing as two distinct phases.

TEST STRATEGY AND APPROACH

Field testing will be performed manually and functional test will be written in detail.

TEST OBJECTIVES

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

FEATURES TO BE TESTED

- Verify that the entries are of the correct format.
- No duplicate entries should be allowed.
- All links should take the user to the correct page.

5.4. ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

CONCLUSION

The main objectives of the project were to develop an algorithm that will be used to identify answers related to user submitted questions. To develop a database were all the related data will be stored and to develop a web interface. The web interface developed had two parts, one for simple users and one for the administrator. A background research took place, which included an overview of the conversation procedure and any relevant chat bots available. A database was developed, which stores information about questions, answers, keywords, logs and feedback messages. A usable system was designed, developed and deployed to the web server on two occasions. An evaluation took place from data collected by potential students of the University. Also, after received feedback from the first deployment, extra requirements were introduced and implemented.

FUTURE WORK

After having a lot of chatting with different users and receiving feedbacks, it will be able to answer quite reasonably. Also, its database will continue to grow in size automatically. Equipping it with face-recognition techniques will give it the ability of recognising its usual users. Moreover, integration of speech-recognition software will give relief to a user from the pain of typing. It can also be integrated into dialogue systems for various practical purposes such as personalized service or information-acquisition. So, it can also be useful apart from being entertaining. Speaking of desktop and mobile apps, the ones whose interface is powered by AI and processed by chatbots, they have the capability of providing flawless delivery and quick access of the required data, at the beck and call of users.

REFERENCES

[1] Emanuela Haller and Traian Rebedea, “Designing a Chat-bot that Simulates an Historical Figure”, IEEE Conference Publications, July 2013.

[2] Maja Pantic, Reinier Zwitserloot, and Robbert Jan Grootjans, “Teaching Introductory Artificial Intelligence Using Asimple Agent Framework”, IEEE Transactions On Education, Vol. 48, No. 3, August 2005.

[3] The Anatomy of A.L.I.C.E.: Dr. Richard S. Wallace, http://www.alicebot.org/anatomy.html

[4] Computing machinery and intelligence, Alan Turing [1950], http://www.abelard.org/turpap/turpap.html

[5] Mauldin, Michael (1994), " ChatterBots, TinyMuds, and the Turing Test: Entering the Loebner Prize Competition", Proceedings of the Eleventh National Conference on Artificial Intelligence, AAAI Press, retrieved 2008-03-05.