

# **STRATEGIC DECISION MAKING AT HIGHER EDUCATIONAL INSTITUTIONS**

Submitted in partial fulfillment of the requirements for  
the award of  
Bachelor of Engineering Degree in Computer Science and Engineering

by

**CHARIT KUMAR REDDY (37110128)**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
SCHOOL OF COMPUTING**

## **SATHYABAMA**

**INSTITUTE OF SCIENCE AND TECHNOLOGY**

**(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,  
CHENNAI – 600119.**

**APRIL 2021**



**SATHYABAMA**  
**INSTITUTE OF SCIENCE AND TECHNOLOGY**  
(DEEMED TO BE UNIVERSITY)  
Accredited with Grade "A" by NAAC  
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,  
Chennai - 600119  
[www.sathyabama.ac.in](http://www.sathyabama.ac.in)



---

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**BONAFIDE CERTIFICATE**

This is to certify that this Project Report is the bonafide work of **CHARIT KUMAR REDDY(37110128)** who carried out the project entitled "**STRATEGIC DECISION MAKING AT HIGHER EDUCATIONAL INSTITUTIONS**" under our supervision from November 2020 to April 2021.

Internal Guide  
**Dr.B.ANKAYARKANNI,M.E.,Phd.,**  
Head of the Department

---

Submitted for Viva voce Examination held on \_\_\_\_\_

Internal Examiner

External Examiner

## **DECLARATION**

I, **CHARIT KUMAR REDDY (37110128)** hereby declare that the Project Report entitled **-STRATEGIC DECISION MAKING AT HIGHER EDUCATIONAL INSTITUTIONS** done by me under the guidance of **Dr.B,Ankayarkanni**, at Sathyabama Institute of Science and technology is submitted in partial fulfillment of the requirements of the award of Bachelor of Engineering degree in Computer Science and Engineering.

**DATE:**

**PLACE:**

**SIGNATURE OF THE CANDIDATE**

## ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to Board of management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

We convey our thanks to **Dr. T. SASIKALA M.E., Ph.D., Dean, School of Computing, Dr. S. VIGNESHWARI M.E., Ph.D. and Dr. L. LAKSHMANAN M.E., Ph.D., Heads of the Department**, Department of Computer Science and Engineering for providing us the necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **Dr.B.Ankayarkanni,,M.E.,Phd.,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

We wish to express our thanks to all Teaching and Non-teaching staff members of the Department of **COMPUTER SCIENCE AND ENGINEERING** who were helpful in many ways for the completion of the project.

## **ABSTRACT**

In many organizations, machine learning techniques are used for analyzing large amount of available data's, information's for decision making process. In educational sector, Machine learning is used for wide variety of applications such as suggestion to the students based on 10<sup>th</sup> mark and interest. One of the most important milestones in an individual's life involves self-analysis, critical thinking and finally decision making. In this paper, we present survey results on which academic decisions they concern and the variables involved in them. Using machine learning algorithms, we predicted courses/institutions in a real case study to support decision making. The choice of the career is influenced by views of your parents, friends, relatives, teachers and the media. Today with a wider choice and an ever increasing competition, you need to plan your career wisely and at the earliest. While choosing a stream after 10th, a training course or a career and 12<sup>th</sup> groups you should know your abilities, interests, and personality. Besides these you should gather information regarding different career options, the eligibility criteria, the premier institutions/Schools, and other criteria of selection and the market demands. The present system undertaken by Education Department, on different educational and vocational courses available and the institutions through competitive examination after 10th and 12th aims at giving you the much needed information.

## TABLE OF CONTENTS

| CHAPTER<br>NO | TITLE                                 | PAGE NUMBER |
|---------------|---------------------------------------|-------------|
|               | <b>ABSTRACT</b>                       |             |
|               | <b>LIST OF FIGURES</b>                |             |
| <b>1</b>      | <b>INTRODUCTION</b>                   | <b>1</b>    |
|               | 1.1 OVERVIEW                          | 2           |
|               | 1.2 SCOPE OF THE PROJECT              | 2           |
|               | 1.3 DOMAIN OVERVIEW                   | 3           |
|               | 1.3.1 MACHINE LEARNING                | 3           |
| <b>2</b>      | <b>LITERATURE SURVEY</b>              | <b>8</b>    |
| <b>3</b>      | <b>SYSTEM ANALYSIS</b>                | <b>12</b>   |
|               | 3.1 EXISTING SYSTEM                   | 18          |
|               | 3.2 DRAWBACKS OF EXISTING SYSTEM      | 18          |
|               | 3.3 PROPOSED SYSTEM                   | 19          |
|               | 3.4 ADVANTAGES OF PROPOSED SYSTEM     | 19          |
| <b>4</b>      | <b>REQUIREMENT SPECIFICATION</b>      | <b>20</b>   |
|               | 4.1 INTRODUCTION                      | 20          |
|               | 4.2 HARDWARE & SOFTWARE SPECIFICATION | 20          |
|               | 4.2.1 HARDWARE SPECIFICATION          | 20          |
|               | 4.2.2 SOFTWARE SPECIFICATION          | 21          |
|               | 4.3 SOFTWARE DESCRIPTION              | 21          |
|               | 4.3.1 ABOUT JAVA                      | 22          |
|               | 4.3.2 ABOUT MYSQL                     | 27          |
|               | 4.3.3 ABOUT IDE                       | 30          |
| <b>5</b>      | <b>SYSTEM DESIGN</b>                  | <b>35</b>   |
|               | 5.1 INTRODUCTION                      | 35          |
|               | 5.2 ARCHITECTURE DIAGRAM              | 36          |
|               | 5.3 DATAFLOW DIAGRAM                  | 36          |
|               | 5.4 UML DIAGRAM                       | 37          |
|               | 5.4.1 USE CASE DIAGRAM                | 37          |
|               | 5.4.2 CLASS DIAGRAM                   | 38          |
|               | 5.4.3 ACTIVITY DIAGRAM                | 39          |

|          |                                           |           |
|----------|-------------------------------------------|-----------|
|          | 5.4.4 SEQUENCE DIAGRAM                    | 39        |
| <b>6</b> | <b>SYSTEM IMPLEMENTATION</b>              | <b>40</b> |
|          | LIST OF MODULES                           | 40        |
|          | EXPLORATORY OF DATA EVALUATION            | 41        |
|          | PRE-PROCESSING                            | 42        |
|          | FEATURE ENGINEERING                       | 42        |
|          | PREDUCTION                                | 43        |
| <b>7</b> | <b>SOFTWARE TESTING</b>                   | <b>44</b> |
|          | GENERAL                                   | 49        |
|          | TEST DRIVEN DEVELOPMENT                   | 50        |
|          | UNIT TESTING                              | 51        |
|          | BLACKBOX TESTING                          | 52        |
|          | INTEGRATION TESTING                       | 53        |
|          | SYSTEM TESTING                            | 54        |
|          | SANITY TESTING                            | 56        |
|          | REGRESSION TESTING                        | 57        |
| <b>8</b> | <b>CONCLUSION &amp; FUTURE ENHACEMENT</b> | <b>63</b> |
|          | CONCLUSION                                | 63        |
|          | FUTURE ENHANCEMENT                        | 63        |
|          | <b>APPENDIX-I SCREEN SHOTS</b>            |           |
|          | <b>APPENDIX-II SAMPLE CODING</b>          |           |
|          | <b>REFERENCE</b>                          | <b>64</b> |

## **CHAPTER 1**

### **INTRODUCTION**

Two imperatives for better use of data confront higher education. The first is driven by external factors while the second is driven internally by continuous quality improvement. Steep declines in financial and public support have driven efforts by governments to collect data that support the proposition that institutions are accountable for the revenue they receive. Working from a defensive posture, many colleges and universities have been able to waylay undesirable changes by satisfying external requests for data. At a higher level, however, those institutions that deliberately use data to improve overall performance meet compliance-based requirements while enacting a future that is informed by data. The proposition that higher education's approach to data use has changed very little may be disputed. At the same time, it also is clear that technology has made new conversations possible. New techniques including analytics or predictive analytics provide institutions new opportunities to use data to improve their efficiency while better serving students (see, for example, Bichsel, 2012 and WCET, n.d.). Colleges and universities are entering an era in which strategic information about student learning and success, budgeting, and efficiency can be united under the umbrella of big data. Higher education is now collecting more data than ever before. However, these efforts are most often directed at the first imperative, compliance reporting, rather than the second imperative, improving institutional strategy. Forward thinking institutions will quickly resolve this seeming dichotomy. They will seek opportunities to build capacity, remove constraints to span existing boundaries that determine data use and find ways to bring data and strategy together. The result can advance institutional mission, meeting external policy demands and improving student success. Strategic thinking and the data that serve those strategies come at a price. In this chapter, we review both opportunities and barriers associated with creating and using actionable strategic and operational data. We also identify successful steps for data use based on our experiences in working with higher education institutions to facilitate strategic planning and to create cultures of inquiry and evidence. We also survey emerging

technologies and their promise to help institutions help their students. This chapter is intended to provide practical advice and not to provide a theoretical overview of the tenets of strategic planning. Institutions sufficiently courageous to engage in a data journey require support. Toward that end, this chapter also provides advice drawn from personal experience and new developments in management science to help navigate these new pathways.

## **OVERVIEW**

One of the main objects of the recently developed policy for higher education regards the creation of a more diversified higher education system with flexible and adaptive institutions. In educational sector, machine learning is used for wide variety of applications such as suggestion to the students based on 10<sup>th</sup> mark and interest. Hoping that it would help you in understanding what each course for career entails and help you introspect and analyze the required abilities to be successful, happy and content in the course/career you choose. While choosing a stream after 10<sup>th</sup>, a training course or a career and 12<sup>th</sup> groups you should know your abilities, interests, and personality. Besides these you should gather information regarding different career options, the eligibility criteria, the premier institutions/Schools, and other criteria of selection and the market demands.

## **SCOPE OF THE PROJECT**

The main contributions of this project therefore are:

- Data Analysis
- Dataset preprocessing
- Training the Model
- Testing of Dataset

## **DOMAIN OVERVIEW**

Machine Learning is the most popular technique of predicting the future or classifying information to help people in making necessary decisions. Machine Learning

algorithms are trained over instances or examples through which they learn from past experiences and also analyze the historical data. Therefore, as it trains over the examples, again and again, it is able to identify patterns in order to make predictions about the future.

Data is the core backbone of machine learning algorithms. With the help of the historical data, we are able to create more data by training these machine learning algorithms. For example, Generative Adversarial Networks are an advanced concept of Machine Learning that learns from the historical images through which they are capable of generating more images. This is also applied towards speech and text synthesis. Therefore, Machine Learning has opened up a vast potential for data science applications.

## **MACHINE LEARNING**

Machine Learning combines computer science, mathematics, and statistics. Statistics is essential for drawing inferences from the data. Mathematics is useful for developing machine learning models and finally, computer science is used for implementing algorithms.

However, simply building models is not enough. You must also optimize and tune the model appropriately so that it provides you with accurate results. Optimization techniques involve tuning the hyperparameters to reach an optimum result.

The world today is evolving and so are the needs and requirements of people. Furthermore, we are witnessing a fourth industrial revolution of data. In order to derive meaningful insights from this data and learn from the way in which people and the system interface with the data, we need computational algorithms that can churn the data and provide us with results that would benefit us in various ways. Machine Learning has revolutionized industries like medicine, healthcare, manufacturing, banking, and several other industries. Therefore, Machine Learning has become an essential part of modern industry.

Data is expanding exponentially and in order to harness the power of this data, added by the massive increase in computation power, Machine Learning has added another

dimension to the way we perceive information. Machine Learning is being utilized everywhere. The electronic devices you use, the applications that are part of your everyday life are powered by powerful machine learning algorithms.

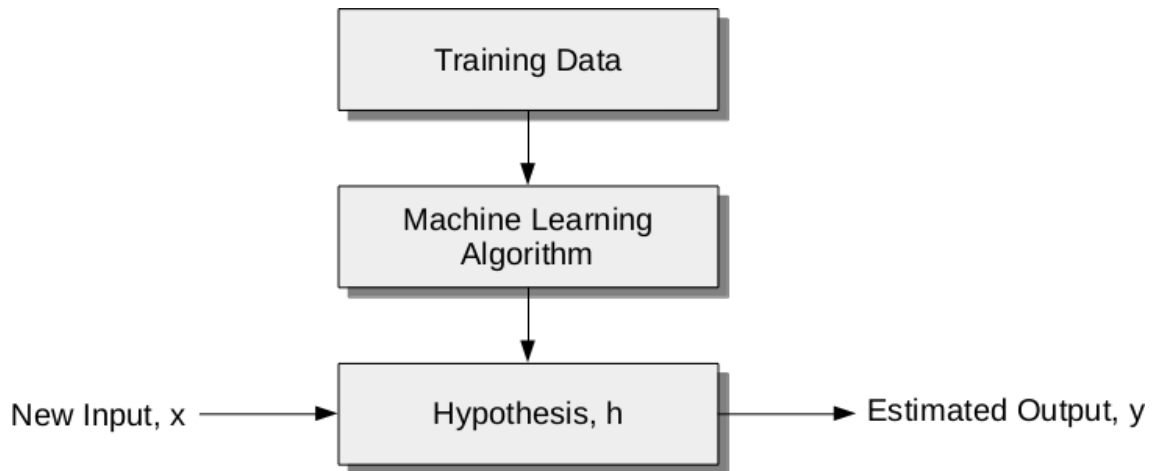
With an exponential increase in data, there is a need for having a system that can handle this massive load of data. Machine Learning models like Deep Learning allow the vast majority of data to be handled with an accurate generation of predictions. Machine Learning has revolutionized the way we perceive information and the various insights we can gain out of it.

These machine learning algorithms use the patterns contained in the training data to perform classification and future predictions. Whenever any new input is introduced to the ML model, it applies its learned patterns over the new data to make future predictions. Based on the final accuracy, one can optimize their models using various standardized approaches. In this way, Machine Learning model learns to adapt to new examples and produce better results.

#### Types of Machine Learning

Machine Learning Algorithms can be classified into 3 types as follows

1. Supervised Learning
2. Unsupervised Learning
3. Reinforcement Learning



## Supervised Learning

In the majority of supervised learning applications, the ultimate goal is to develop a finely tuned predictor function  $h(x)$  (sometimes called the —hypothesis—). —Learning— consists of using sophisticated mathematical algorithms to optimize this function so that, given input data  $x$  about a certain domain (say, square footage of a house), it will accurately predict some interesting value  $h(x)$  (say, market price for said house).

$$h(x_1, x_2, x_3, x_4) = \theta_0 + \theta_1 x_1 + \theta_2 x_3^2 + \theta_3 x_3 x_4 + \theta_4 x_1^3 x_2^2 + \theta_5 x_2 x_3^4 x_4^2$$

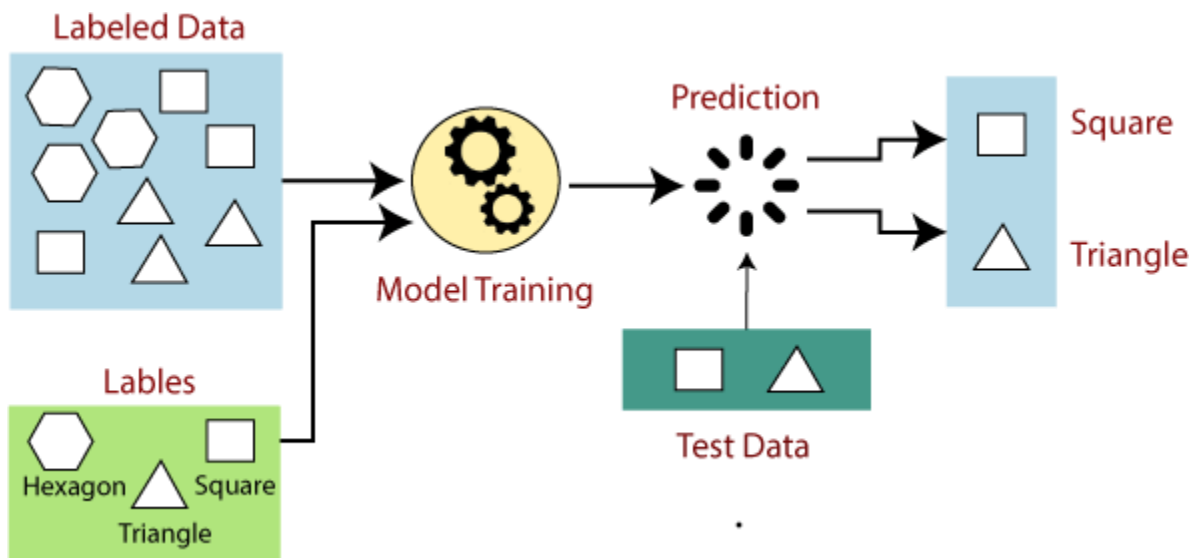
This function takes input in four dimensions and has a variety of polynomial terms. Deriving a normal equation for this function is a significant challenge. Many modern machine learning problems take thousands or even millions of dimensions of data to build predictions using hundreds of coefficients. Predicting how an organism's genome will be expressed, or what the climate will be like in fifty years, are examples of such complex problems.

Under supervised ML, two major subcategories are:

- Regression machine learning systems: Systems where the value being predicted falls somewhere on a continuous spectrum. These systems help us with questions of —How much?— or —How many?—.
- Classification machine learning systems: Systems where we seek a yes-or-no prediction, such as —Is this tumor cancerous?—, —Does this cookie meet our

quality standards?ll, and so on.

In practice,  $x$  almost always represents multiple data points. So, for example, a housing price predictor might take not only square-footage ( $x_1$ ) but also number of bedrooms ( $x_2$ ), number of bathrooms ( $x_3$ ), number of floors ( $x_4$ ), year built ( $x_5$ ), zip code ( $x_6$ ), and so forth. Determining which inputs to use is an important part of ML design. However, for the sake of explanation, it is easiest to assume a single input value is used.



Steps Involved in Supervised Learning:

- First Determine the type of training dataset.
- Collect/Gather the labelled training data.
- Split the training dataset into training dataset, test dataset, and validation dataset.
- Determine the input features of the training dataset, which should have enough knowledge so that the model can accurately predict the output.
- Determine the suitable algorithm for the model, such as support vector machine, decision tree, etc.
- Execute the algorithm on the training dataset. Sometimes we need validation sets as the control parameters, which are the subset of training datasets.

- Evaluate the accuracy of the model by providing the test set. If the model predicts the correct output, which means our model is accurate.

## Regression

Regression algorithms are used if there is a relationship between the input variable and the output variable. It is used for the prediction of continuous variables, such as Weather forecasting, Market Trends, etc. Below are some popular Regression algorithms which come under supervised learning:

1. Linear Regression
2. Regression Trees
3. Non-Linear Regression
4. Bayesian Linear Regression
5. Polynomial Regression

## Classification

Classification algorithms are used when the output variable is categorical, which means there are two classes such as Yes-No, Male-Female, True-false, etc.

- Spam Filtering,
- Random Forest
- Decision Trees
- Logistic Regression
- Support vector Machines

## **CHAPTER 2**

### **LITERATURE SURVEY**

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, then the next step is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration are taken into account for developing the proposed system. The major part of the project development sector considers and fully survey all the required needs for developing the project. For every project Literature survey is the most important sector in software development process. Before developing the tools and the associated designing it is necessary to determine and survey the time factor, resource requirement, man power, economy, and company strength. Once these things are satisfied and fully surveyed, then the next step is to determine about the software specifications in the respective system such as what type of operating system the project would require, and what are all the necessary software are needed to proceed with the next step such as developing the tools, and the associated operations.

#### **Barriers and drivers of innovation in higher education: Case study-based evidence across ten European Universities**

The paper advances current knowledge on factors affecting higher education institutions in their quest for innovation in education. Based on an analysis of ten institutional cases from five European countries, a comprehensive description and classification of barriers and drivers of innovation are provided. Results indicate certain —disengagementll in relation of higher education institutions and education policy makers, business, and students as well as between higher education institutions' managers and their subordinates. Based on the findings, major

innovation-related challenges in the higher education are discussed and related practical recommendations are presented.

### **Supporting academic decision making at higher educational institutions using machine learning-based algorithms**

Decisions made by deans and university managers greatly impact the entire academic community as well as society as a whole. In this paper, we present survey results on which academic decisions they concern and the variables involved in them. Using machine learning algorithms, we predicted graduation rates in a real case study to support decision making. Real data from five undergraduate engineering programs at District University Francisco Jose de Caldas in Colombia illustrate our results. The comparison between support vector machine and artificial neural network is held using the confusion matrix and the receiver operating characteristic curve. The algorithm methods and architecture are presented.

### **The effect of blended learning on student performance at course-level in higher education: A meta-analysis**

The present paper analyzes the impact of blended learning (BL) on the academic achievement of higher education students. A meta-analysis ( $k = 51$  effect sizes) was conducted to perform a statistical synthesis of studies contrasting student performance in BL conditions with traditional classroom instruction. We include disciplines and instructors' end-of-course evaluation method as moderating variables. The results show that BL demonstrates a small summary effect ( $g^+ = 0.385$ ,  $p < 0.001$ ) compared to traditional teaching methods. A significantly higher mean effect size was found in STEM disciplines ( $g^+ = 0.496$ ) compared to that of non-STEM disciplines ( $g^+ = 0.210$ ). Nevertheless, the weighted mean effect sizes reveal no significant differences regarding of end-of-course assessment methods, namely one-moment and multiple-component assessment. The finding confirms that BL is significantly associated with greater learning performance of STEM-disciplined students than with traditional classroom practice. Accordingly, discussion concerning the findings and implications for future research are elaborated.

## **Using system dynamics to develop education for sustainable development in higher education with the emphasis on the sustainability competencies of students**

In response to growing concerns of the community about sustainability challenges and the intensification of the international calls to move towards a sustainable future, higher education should be involved in implementing the programs of education for sustainable development; because of this fact that University graduates are part of this solution as future leaders and inheritors of technology. In this study, the underlying research question is: What mechanisms are needed to develop education for sustainable development in higher education with the emphasis on the sustainability competencies of students? The research method was system dynamics. Therefore, we used a mix method research design. Tools of research included the literature review, questionnaire, interview, and observation. We developed a dynamic model to develop the education for sustainable development in higher education with the emphasis on the sustainability competencies of students. This model describes the research problem and predicts the behavior of model variables by simulating in the next 20 years. This model included 18 reinforcing and six balancing feedback loops. After ensuring the validity of the model, mechanisms were elicited from the model. Finally, we evaluated these mechanisms for finding the impacts on improving the

## **Strategic alignment between sustainability and information systems: A case analysis in Malaysian public higher education Institutions**

Higher Education Institutions have embedded sustainability initiatives into their core activities of curriculum, research, community, and operational to respond the global transformation towards a sustainable future. Numerous studies have been conducted concerning an integrated sustainability into the missions and strategies of the Higher Education Institutions. However, there is a lack of works that highlight the important role of information systems to support the sustainability practices in Higher Education Institutions. This problem motivates the study to underscore the important role of information systems during the sustainability implementation. A preliminary case analysis is performed to observe how the Higher Education Institutions implement the sustainability initiatives and to proof the knowledge gap in real practice. The analysis was conducted during visits via a semi-structured interview with a number of persons from Higher Education Institutions in Malaysia that have been implementing sustainable campus. According to the preliminary case analysis, the Higher Education Institutions do not contemplate the information system strategy at the beginning of their sustainability implementation. There is a lack of information system application to support their sustainability practices. They still use manual methods to collect the sustainability metrics and assess their sustainability performance. As a result, the sustainability decision-making is isolated and they cannot evaluate their sustainable performance. The problems of managing the sustainability data and processes arise. Therefore, a concept of the strategic alignment between sustainability and information system is underscored in the present study. The introduction of this concept through this study would enhance our understanding in term of the significant role of the information systems to support the sustainability practices and to achieve a successfully sustainable campus.

## **CHAPTER 3**

### **SYSTEM ANALYSIS**

One of the primary choices we face for our project implementations is -Which development methodology should we use?|| There are various software development methodologies available in the ICT industry. They are

- Agile Software Development
- Crystal Methods
- Dynamic Systems Development Model (DSDM)
- Extreme Programming (XP)
- Feature Driven Development (FDD)
- Joint Application Development (JAD)
- Lean Development (LD)
- Rapid Application Development (RAD)
- Rational Unified Process (RUP)
- Scrum
- Spiral
- Systems Development Life Cycle (SDLC)
- Waterfall (a.k.a. Traditional)

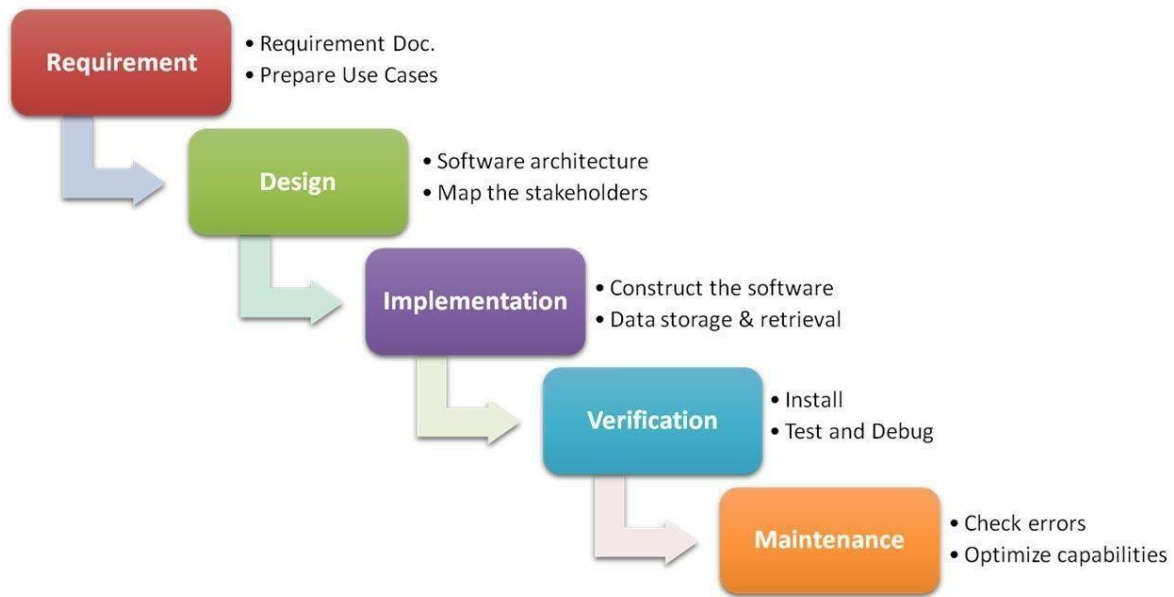
Among all these methodologies, we have decided to compare the two most popular and widely used methodologies:

- Waterfall Model
- RAD Model

#### **Waterfall Model**

It is called as traditional approach. Waterfall is a linear method (sequential model) for any software application development. Here application development is segregated into a sequence of pre -defined phases.

Waterfall Model ( Taken from Google.com)



(Reference : <https://www.lucidchart.com/blog/waterfall-project-management-methodology>)

### 1. Requirement gathering and documentation

In this stage, you should gather comprehensive data approximately what this challenge requires. You may gather this data in a diffusion of ways, from interviews to questionnaires to interactive brainstorming. By means of the stop of this section, the task requirements have to be clean, and also you have to have a necessities file that has been allotted on your team.

### 2. System design

Using the well-known requirements, your team designs the solutions. During this phase, no development will be happening. But the project team starts specification such as programming language or hardware requirements.

### 3. Implementation

During this phase software development coding will be happening. Web application programmers take data from the previous stage and create a functional product. Web application programmers write source code in small pieces, which are integrated at the end of this phase or the beginning of the next.

#### 4. Testing

Once all coding is done, testing of the product can begin. Testers methodically find and report any problems. If serious issues arise, your project may need to return to phase one for reevaluation.

#### 5. Delivery/deployment

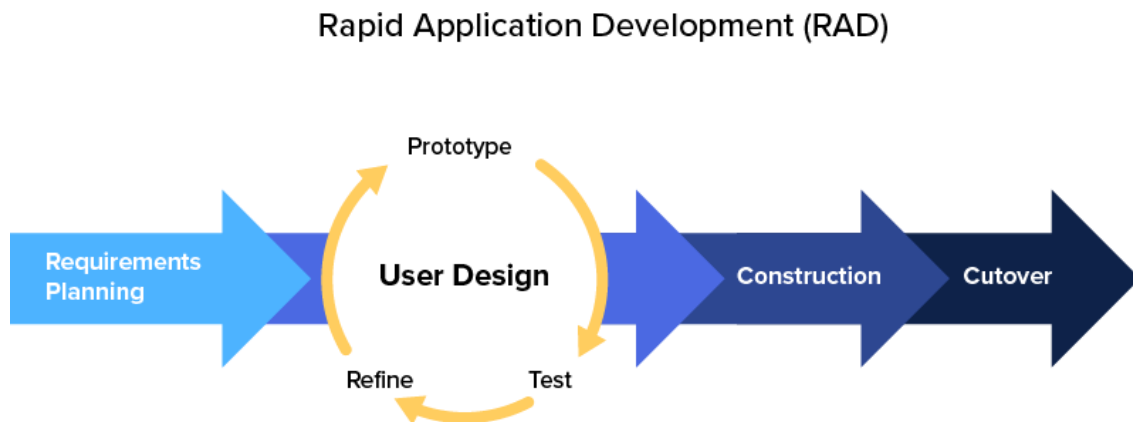
In this phase, the solution is complete, and your project team submits the deliverables to be deployed or released.

#### 6. Maintenance

The final solution has been implemented to the client and is being used. As troubles arise, the project team might also want to create patches and updates might also to deal with them. Again, huge troubles may also necessitate a return to segment one.

## RAD Model

Agile is an iterative, team-based approach to development. This method highlights the speedy delivery of an software application in comprehensive functional components.



RAD Model (Taken from Google.com)

References (<https://www.lucidchart.com/blog/rapid-application-development-methodology>)

### Phase 1: Requirements planning

This phase is equivalent to a project scoping meeting. Although the planning phase is condensed compared to other project management methodologies, this is a critical step for the ultimate success of the project.

During this stage, web application programmers, clients (software users), and team members communicate to determine the goals and expectations for the project as well as current and potential issues that would need to be addressed during the build.

A basic breakdown of this stage involves:

Researching the current problem

Defining the requirements for the project

Finalizing the requirements with each stakeholder's approval

It is important that everyone has the opportunity to evaluate the goals and expectations for the project and weigh in. By getting approval from each key stakeholder and web application programmer, teams can avoid miscommunications and costly change orders down the road.

Phase: User design

Once the project is scoped out, it's time to jump right into development, building out the user design through various prototype iterations.

This is the meat and potatoes of the RAD methodology—and what sets it apart from other project management strategies. During this phase, clients work hand in hand with web application programmers to ensure their needs are being met at every step in the design process. It's almost like customizable software development where the users can test each prototype of the product, at each stage, to ensure it meets their expectations.

All the bugs and kinks are worked out in an iterative process. The web application programmer designs a prototype, the client (user) tests it, and then they come together to communicate on what worked and what didn't.

This method gives web application programmers the opportunity to tweak the model as they go until they reach a satisfactory design.

Both the software web application programmers and the clients learn from the experience to make sure there is no potential for something to slip through the cracks.

Phase 3: Rapid construction

Phase 3 takes the prototypes and beta systems from the design phase and converts them into the working model.

Because the majority of the problems and changes were addressed during the thorough iterative design phase, web application programmers can construct the final working model more quickly than they could by following a traditional project management approach.

The phase breaks down into several smaller steps:

- 1.Preparation for rapid construction
- 2.Program and application development
- 3.Coding Unit, integration, and system testing

The software development team of programmers, coders, testers, and web application programmers work together during this stage to make sure everything is working smoothly and that the end result satisfies the client's expectations and objectives.

This third phase is important because the client still gets to give input throughout the process. They can suggest alterations, changes, or even new ideas that can solve problems as they arise.

#### Phase 4: Cutover

This is the implementation phase where the finished product goes to launch. It includes data conversion, testing, and changeover to the new system, as well as user training.

All final changes are made while the coders and clients continue to look for bugs in the system.

#### Benefits of RAD methodology

RAD is one of the most successful software development programs available today, with numerous benefits for both software development teams as well as their clients.

Here are just a few advantages:

- RAD lets you break the project down into smaller, more manageable tasks.

- The task-oriented structure allows project managers to optimize their team's efficiency by assigning tasks according to members' specialties and experience.
- Clients get a working product delivered in a shorter time frame.
- Regular communication and constant feedback between team members and stakeholders increases the efficiency of the design and build process.

### **EXISTING SYSTEM**

In existing system, three supervised classification algorithms are deployed to predict graduation rates from real data about undergraduate engineering students in South America. The analysis of receiver operating characteristic curve and accuracy are executed as measures of effectiveness to compare and evaluate decision tree, logistic regression, and random forest, where this last one demonstrates the best outcomes.

### **DRAWBACKS OF EXISTING SYSTEM**

- Existing system miss the undesirable data for the students
- It may not check the social data for the student.

## **PROPOSED SYSTEM**

In proposed system, machine learning is used for wide variety of applications such as suggestion to the students based on 10<sup>th</sup> mark and interest. One of the most important milestones in an individual's life involves self-analysis, critical thinking and finally decision making. While choosing a stream after 10<sup>th</sup>, a training course or a career and 12<sup>th</sup> groups you should know your abilities, interests, and personality. Besides these you should gather information regarding different career options, the eligibility criteria, the premier institutions/Schools, and other criteria of selection and the market demands.

## **ADVANTAGES OF PROPOSED SYSTEM**

- Accuracy is high
- High efficiency

## **4. PROPOSED ALGORITHM**

### **K-MEANS CLUSTERING ALGORITHM:**

What is K-means?

1. Partitional clustering approach
2. Each cluster is associated with a centroid (center point)
3. Each point is assigned to the cluster with the closest centroid
- 4 Number of clusters K must be specified

## **CHAPTER 4**

### **REQUIREMENT SPECIFICATION**

#### **INTRODUCTION**

Clearly defined requirements are essential signs on the road that leads to a successful project. They establish a formal agreement between a client and a provider that they are both working to reach the same goal. High-quality, detailed requirements also help mitigate financial risks and keep the project on a schedule. According to the Business Analysis Body of Knowledge definition, requirements are a usable representation of a need.

Creating requirements is a complex task as it includes a set of processes such as elicitation, analysis, specification, validation, and management. In this article, we'll discuss the main types of requirements for software products and provide a number of recommendations for their use.

#### **HARDWARE SPECIFICATION**

##### **HARDWARE REQUIREMENT**

| Hardware   | Minimum Requirement                                |
|------------|----------------------------------------------------|
| Disk Space | 32 GB or more,10 GB or more for Foundation Edition |
| Processor  | 1.4 GHz 64 bit                                     |
| Memory     | 512 MB                                             |
| Display    | (800 × 600) Capable video adapter and monitor      |

## **SOFTWARE SPECIFICATION**

- Operating System - Windows XP
- Coding language - Java
- IDE - Net beans
- Database -MySQL

## **SOFTWARE DESCRIPTION**

### **ABOUT JAVA**

Java is one of the world's most important and widely used computer languages, and it has held this distinction for many years. Unlike some other computer languages whose influence has wearied with passage of time, while Java's has grown.



### **APPLICATION OF JAVA**

Java is widely used in every corner of world and of human life. Java is not only used in softwares but is also widely used in designing hardware controlling software

components. There are more than 930 million JRE downloads each year and 3 billion mobile phones run java.

Following are some other usage of Java:

1. Developing Desktop Applications
2. Web Applications like Linkedin.com, Snapdeal.com etc
3. Mobile Operating System like Android
4. Embedded Systems
5. Robotics and games etc.

## **FEATURES OF JAVA**

The prime reason behind creation of Java was to bring portability and security feature into a computer language. Beside these two major features, there were many other features that played an important role in moulding out the final form of this outstanding language. Those features are;

### **1) Simple**

Java is easy to learn and its syntax is quite simple, clean and easy to understand. The confusing and ambiguous concepts of C++ are either left out in Java or they have been re-implemented in a cleaner way.

*Eg:* Pointers and Operator Overloading are not there in java but were an important part of C++.

### **2) Object Oriented**

In java everything is Object which has some data and behaviour. Java can be easily extended as it is based on Object Model.

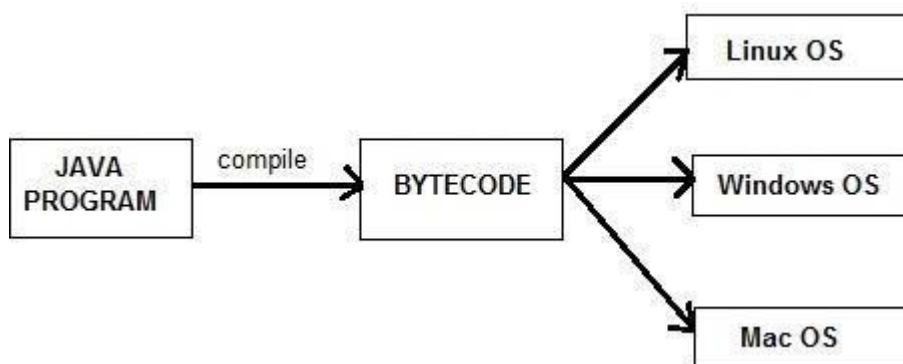
### 3) Robust

Java makes an effort to eliminate error prone codes by emphasizing mainly on compile time error checking and runtime checking. But the main areas which Java improved were Memory Management and mishandled Exceptions by introducing automatic Garbage Collector and Exception Handling.

### 4) Platform Independent

Unlike other programming languages such as C, C++ etc. which are compiled into platform specific machines. Java is guaranteed to be write-once, run-anywhere language.

On compilation Java program is compiled into byte code. This byte code is platform independent and can be run on any machine, plus this byte code format also provide security. Any machine with Java Runtime Environment can run Java Programs.



### 5) Secure

When it comes to security, Java is always the first choice. With java secure features it enable us to develop virus free, temper free system. Java program always

runs in Java runtime environment with almost null interaction with system OS, hence it is more secure.

## **6) Multi-Threading**

Java multithreading feature makes it possible to write program that can do many tasks simultaneously. Benefit of multithreading is that it utilizes same memory and other resources to execute multiple threads at the same time, like While typing, grammatical errors are checked along.

## **7) Architectural Neutral**

Compiler generates byte codes, which have nothing to do with a particular computer architecture, hence a Java program is easy to interpret on any machine.

### **Portable**

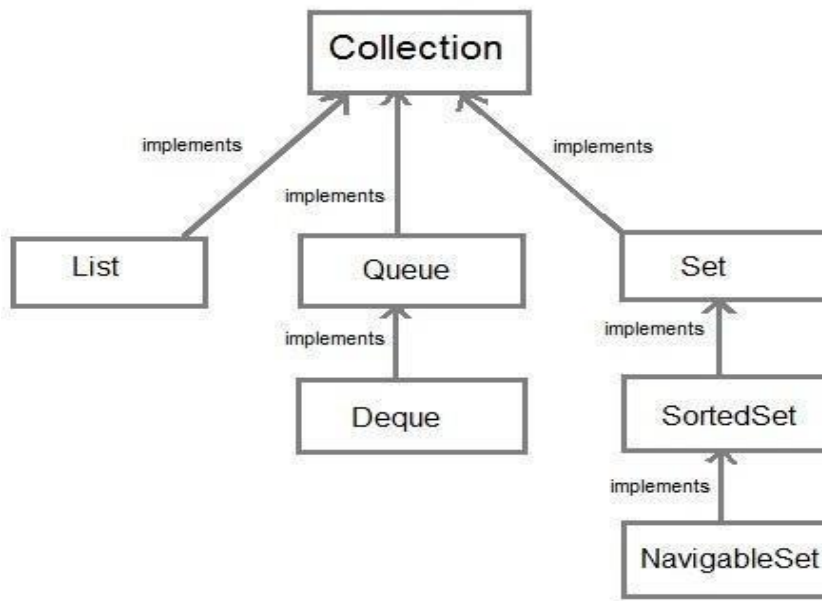
Java Byte code can be carried to any platform. No implementation dependent features. Everything related to storage is predefined, example: size of primitive data types

### **High Performance**

Java is an interpreted language, so it will never be as fast as a compiled language like C or C++. But, Java enables high performance with the use of just-in-time compiler.

## **COLLECTION FRAMEWORK**

Collection framework was not part of original Java release. Collections was added to J2SE 1.2. Prior to Java 2, Java provided adhoc classes such as Dictionary, Vector, Stack and Properties to store and manipulate groups of objects. Collection framework provides many important classes and interfaces to collect and organize group of alike objects.



## MYSQL

MySQL, officially, but also called "My Sequel" is the world's most widely used open-source relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases, though SQLite probably has more total embedded deployments. The SQL phrase stands for Structured Query Language.

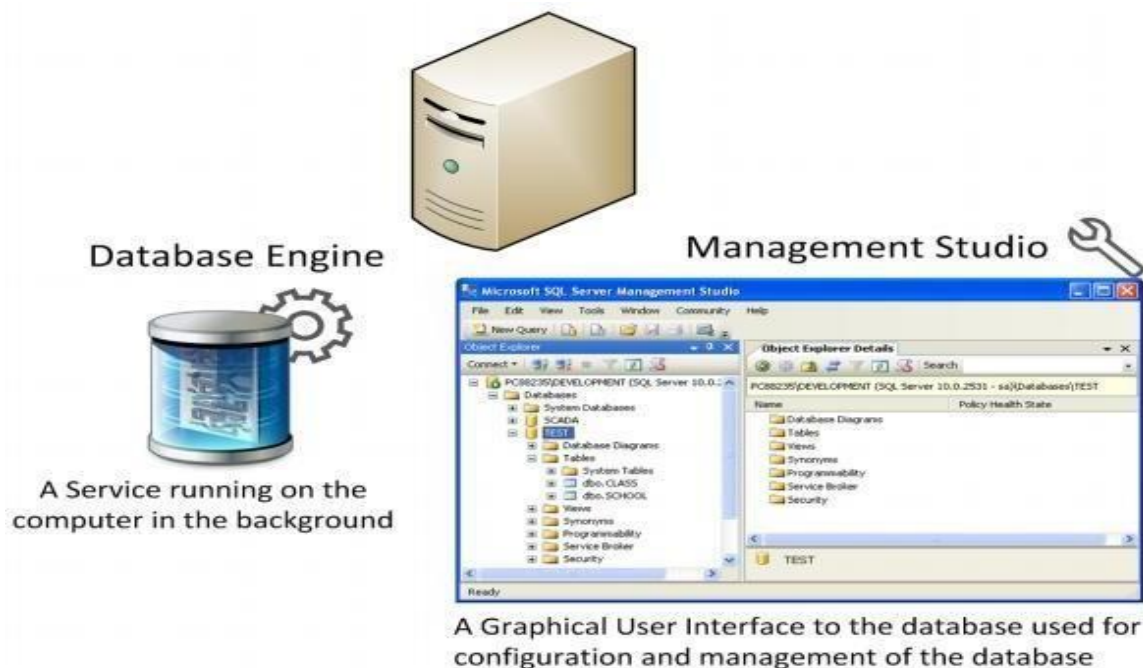
The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack (and other 'AMP' stacks).

LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Freesoftware-open source projects that require a full-featured database management system often use MySQL.

For commercial use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: TYPO3, MODx, Joomla, WordPress, phpBB, MyBB, Drupal and other software. MySQL is also used in many high-profile, large-scale websites, including Wikipedia, Google (though not for searches), Facebook, Twitter, Flickr and YouTube. Microsoft is the vendor of SQL Server. We have different editions of SQL Server, where SQL Server Express is free to download and use. SQL Server uses T-SQL (Transact-SQL). T-SQL is Microsoft's proprietary extension to SQL. TSQL is very similar to standard SQL, but in addition it supports some extra functionality, built in functions, etc.

T-SQL expands on the SQL standard to include procedural programming, local variables, and various support functions for string processing, date processing, mathematics, etc. SQL Server consists of a Database Engine and a Management Studio (and lots of other stuff which we will not mention here). The Database engine has no graphical interface - it is just a service running in the background of your computer (preferable on the server). The Management Studio is graphical tool for configuring and viewing the information in the database. It can be installed on the server or on the client (or both).

SQL Server Management Studio can also be used to create a new database, alter any existing database schema by adding or modifying tables and indexes, or analyze performance. It includes the query windows which provide a GUI based interface to write and execute queries.

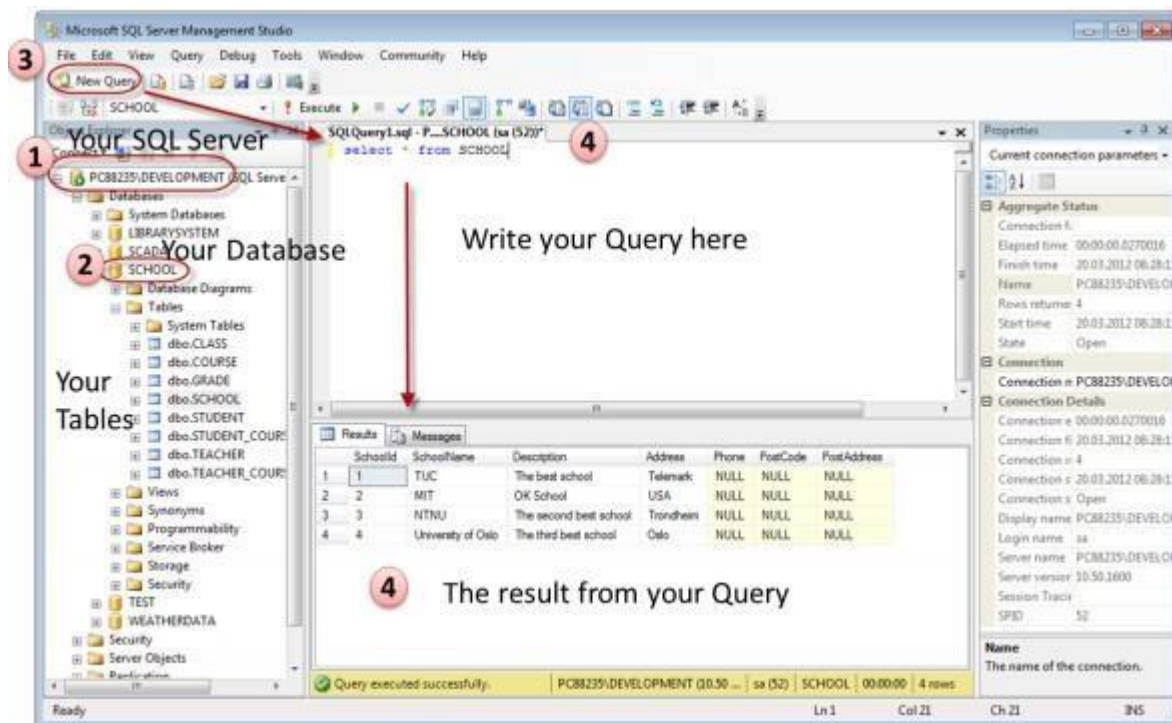


## SQL SERVER MANAGEMENT STUDIO

SQL Server Management Studio is a GUI tool included with SQL Server for configuring, managing, and administering all components within Microsoft SQL Server. The tool includes both script editors and graphical tools that work with objects and features of the server. As mentioned earlier, version of SQL Server Management Studio is also available for SQL Server Express Edition, for which it is known as SQL Server Management Studio Express.

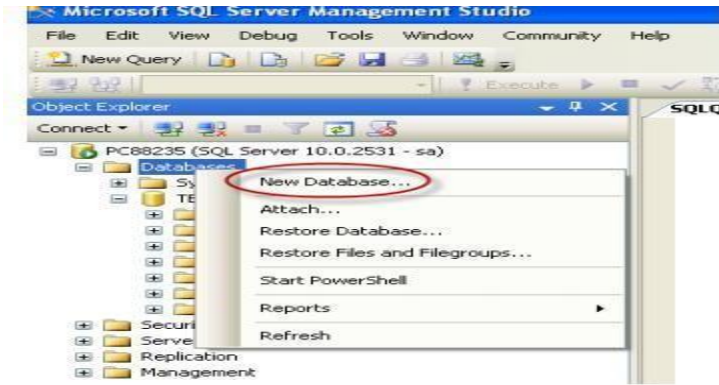
A central feature of SQL Server Management Studio is the Object Explorer, which allows the user to browse, select, and act upon any of the objects within the server. It can be used to visually observe and analyze query plans and optimize the database performance, among others. SQL Server Management Studio can also be used to create a new database, alter any existing database schema by adding or modifying tables and indexes, or analyze performance. It includes the query windows which provide a GUI based interface to write and execute queries.

When creating SQL commands and queries, the —Query Editor (select —New Query from the Toolbar) is used (shown in the figure above). With SQL and the —Query Editor we can do almost everything with code, but sometimes it is also a good idea to use the different Designer tools in SQL to help us do the work without coding (so much).



## Create a new Database

It is quite simple to create a new database in Microsoft SQL Server. Just right-click on the —Databases node and select —New Database...



The OLAP Services feature available in SQL Server version 7.0 is now called SQL Server 2000 Analysis Services. The term OLAP Services has been replaced with the term Analysis Services. Analysis Services also includes a new data mining component. The Repository component available in SQL Server version 7.0 is now called Microsoft SQL Server 2000 Meta Data Services. References to the component now use the term Meta Data Services. The term repository is used only in reference to the repository engine within Meta Data Services

SQL-SERVER database consist of six type of objects,

They are,

1. TABLE
2. QUERY
3. FORM
4. REPORT
5. MACRO

## **Table**

A database is a collection of data about a specific topic.

## **Views of Table**

We can work with a table in two types,

1. Design View
2. Datasheet View

## **Design View**

To build or modify the structure of a table we work in the table design view. We can specify what kind of data will be hold.

## **Datasheet View**

To add, edit or analyses the data itself we work in tables datasheet view mode.

## **Query**

A query is a question that has to be asked the data. Access gathers data that answers the question from one or more table. The data that make up the answer is either dynaset (if you edit it) or a snapshot (it cannot be edited).Each time we run query, we get latest information in the dynaset. Access either displays the dynaset or snapshot for us to view or perform an action on it, such as deleting or updating.

## **ABOUT IDE**

IDE – Integrated Development Environment. It is software suite. It contains different instruments for web application software engineers to compose and test applications.

Programming Web application software engineers utilize wide scope of devices habitually all through application source code creation, building and application testing. Programming Development apparatuses frequently incorporate word processors for source code creation, code libraries for reuse, compilers and test stages.

An Integrated Development Environment (IDE) is a product application that gives a programming domain to streamline creating and troubleshooting programming. Instead of playing out every one of the means required to make an executable program as disconnected individual assignments, it brings every one of the devices required into one application and workspace. Every one of the devices has a familiarity with the earth, and they cooperate to display a consistent improvement set for the designer.

An IDE typically has an editor and a compiler window. There, we can write our code, run it, and compile it. Hence, it becomes much easier to develop applications using an IDE. The major reasons for using an IDE are given below:

- We can easily write our code in its text editor window.
- To store the resource files of a project, we have a project editor window.
- It makes debugging of the code easier.
- An IDE's autosuggest feature helps us in writing efficient code.
- We can import other projects for integrating with the existing one.

Without an IDE, a product web application software engineer must choose, send, incorporate and deal with these instruments independently. An IDE brings huge numbers of those improvement related instruments all together system, application or administration. The incorporated toolset is intended to streamline programming advancement and can distinguish and limit coding missteps and grammatical errors.

### Best Java IDEs

The following are the best Java IDEs that are mostly used in the world:

- Eclipse
- NetBeans
- IntelliJ IDEA
- BlueJ
- JCreator
- JDeveloper

- MyEclipse
- Greenfoot
- DrJava
- Xcode
- Codenvy

## Eclipse

It is a Java-based open-source platform that enables us to create highly customized IDEs from Eclipse member's plug-in components. This platform is also suitable for beginners to create user-friendly and more sophisticated applications. It contains many plugins that enable developers to develop and test code written in different languages. Some of Eclipse's features are as follows:

1. Eclipse provides powerful tools for different software development processes, such as charting, reporting, checking, etc. so that Java developers can build the application as quickly as possible.
2. Eclipse can be used on platforms such as MacOS, Linux, Windows, and Solaris.
3. Eclipse could also make several mathematical documents with LaTeX using both the TeXlipse plug-in and Mathematica software packages.

## FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

The feasibility study investigates the problem and the information needs of the stakeholders. It seeks to determine the resources required to provide an information systems solution, the cost and benefits of such a solution, and the feasibility of such a solution.

The goal of the feasibility study is to consider alternative information systems solutions, evaluate their feasibility, and propose the alternative most suitable to the organization. The feasibility of a proposed solution is evaluated in terms of its components.

### **ECONOMICAL FEASIBILITY**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system is well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### **TECHNICAL FEASIBILITY**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

## **SOCIAL FEASIBILITY**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity.

## **CHAPTER 5**

### **SYSTEM DESIGN**

#### **INTRODUCTION**

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.

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 efficiency.

#### **ARCHITECTURE DIAGRAM**

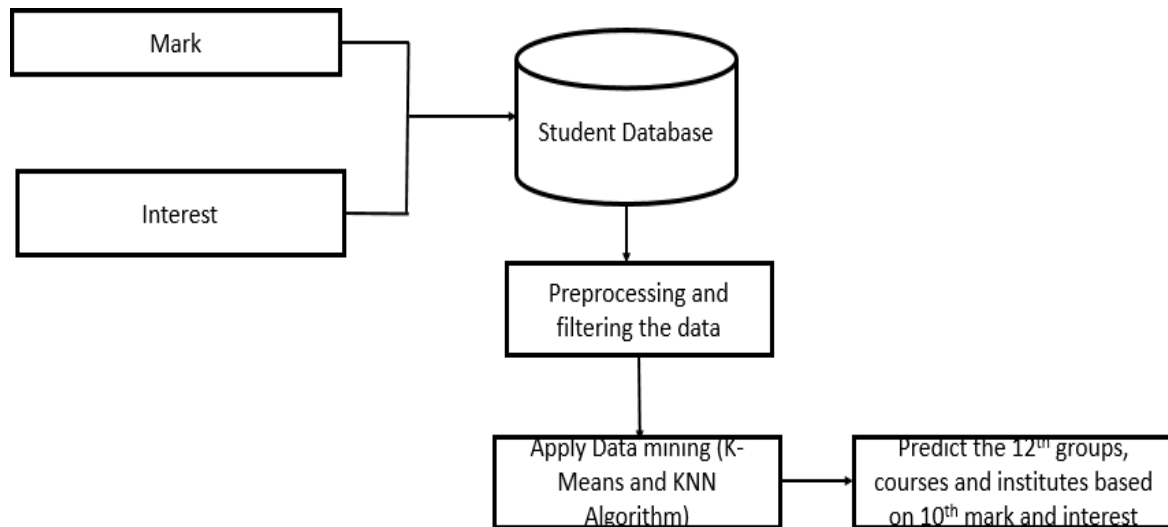
An architecture diagram is a graphical representation of a set of concepts, that are part of an architecture, including their principles, elements and components.

What is a diagram? What are the types of diagrams for architecture? The Dragon1 open EA Method makes it very clear: if a diagram does not show a concept, principle or part of a principle, it is NOT an architecture diagram, because it does not show (a part of) the architecture.

There are many kinds of architecture diagrams, like a software architecture diagram, system architecture diagram, application architecture diagram, security architecture diagram, etc.

For system developers, they need system architecture diagrams to understand, clarify, and communicate ideas about the system structure and the user requirements that the system must support. It's a basic framework can be used at the system

planning phase helping partners understand the architecture, discuss changes, and communicate intentions clearly.



## UML DIAGRAMS

UML is simply another graphical representation of a common semantic model. UML provides a comprehensive notation for the full lifecycle of object-oriented development.

## ADVANTAGES

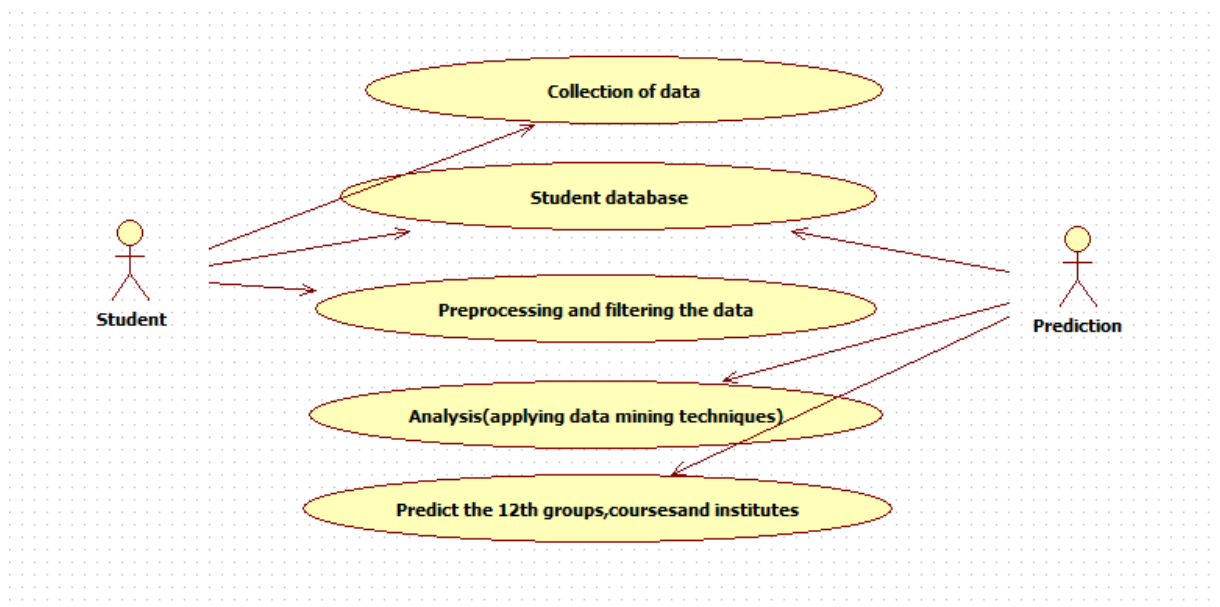
- To represent complete systems (instead of only the software portion) using object oriented concepts
- To establish an explicit coupling between concepts and executable code
- To take into account the scaling factors that are inherent to complex and critical systems
- To creating a modeling language usable by both humans and machines

UML defines several models for representing systems

- The class model captures the static structure
- The state model expresses the dynamic behavior of objects
- The use case model describes the requirements of the user
- The interaction model represents the scenarios and messages flows
- The implementation model shows the work units
- The deployment model provides details that pertain to process allocation

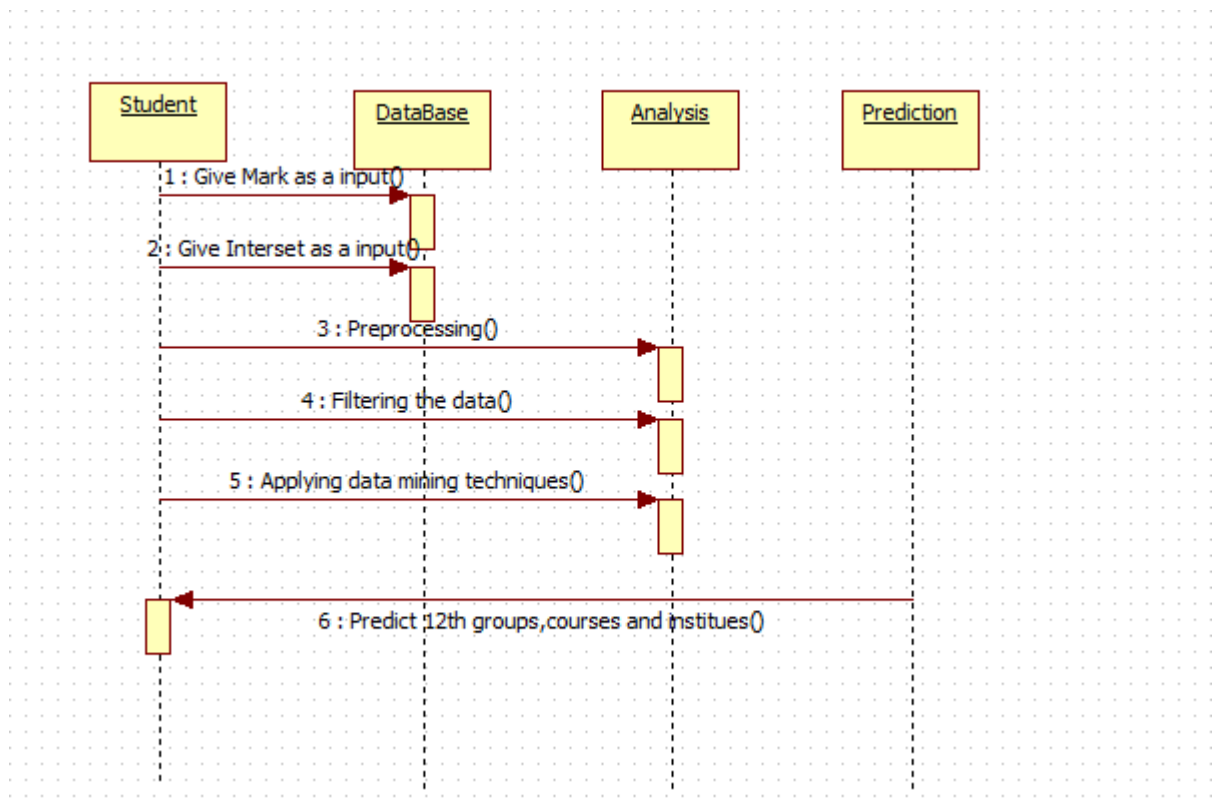
## USECASE DIAGRAM

Use case diagrams overview the usage requirement for system. They are useful for presentations to management and/or project stakeholders, but for actual development you will find that use cases provide significantly more value because they describe —the meantll of the actual requirements. A use case describes a sequence of action that provides something of measurable value to an action and is drawn as a horizontal ellipse.



## SEQUENCE DIAGRAM

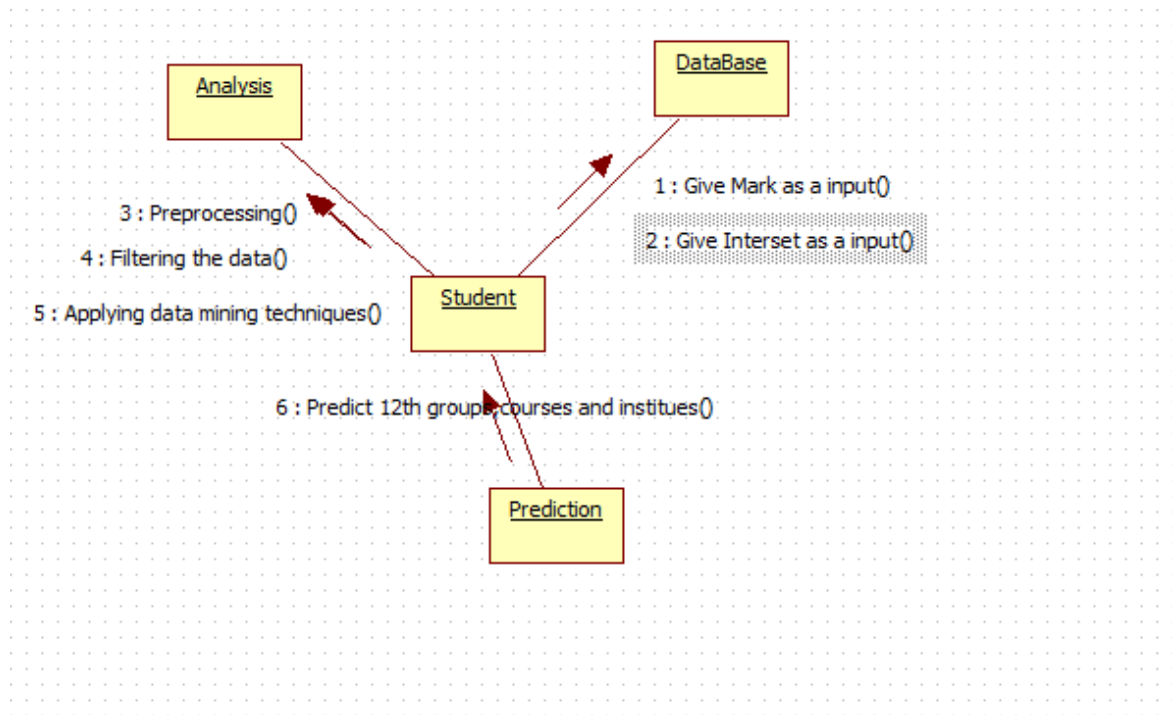
Sequence diagram model the flow of logic within your system in a visual manner, enabling you both to document and validate your logic, and commonly used for both analysis and design purpose. Sequence diagram are the most popular UML artifact for dynamic modeling, which focuses on identifying the behavior within your system.



## COLLABORATION DIAGRAM

Another type of interaction diagram is the collaboration diagram. A collaboration diagram represents a collaboration, which is a set of objects related in

a particular context, and interaction, which is a set of messages exchange among the objects within the collaboration to achieve a desired outcome.



## **CHAPTER 6**

### **SYSTEM IMPLEMENTATION**

#### **EXPLORATORY DATA EVALUATION**

Exploratory Data Analysis (EDA) is the first step in your data analysis process. Here, you make sense of the data you have and then figure out what questions you want to ask and how to frame them, as well as how best to manipulate your available data sources to get the answers you need. You do this by taking a broad look at patterns, trends, outliers, unexpected results and so on in your existing data, using visual and quantitative methods to get a sense of the story this tells.

Exploratory Data Analysis is valuable to data science projects since it allows to get closer to the certainty that the future results will be valid, correctly interpreted, and applicable to the desired business contexts. Such level of certainty can be achieved only after raw data is validated and checked for anomalies, ensuring that the data set was collected without errors. EDA also helps to find insights that were not evident or worth investigating to business stakeholders and data scientists but can be very informative about a particular business.

stakeholders and data scientists but can be very informative about a particular business.

EDA is performed in order to define and refine the selection of feature variables that will be used for machine learning. Once data scientists become familiar with the data set, they often have to return to feature engineering step, since the initial features may turn out not to be serving their intended purpose. Once the EDA stage is complete, data scientists get a firm feature set they need for supervised and unsupervised machine learning.

#### **PRE-PROCESSING**

Sometimes you may find some data are missing in the dataset. We need to be equipped to handle the problem when we come across them. Obviously you could remove the entire line of data but what if you are unknowingly removing crucial

information? Of course we would not want to do that. One of the most common idea to handle the problem is to take a mean of all the values of the same column and have it to replace the missing data.

The library that we are going to use for the task is called Scikit Learn preprocessing. It contains a class called Imputer which will help us take care of the missing data.

Sometimes our data is in qualitative form, that is we have texts as our data. We can find categories in text form. Now it gets complicated for machines to understand texts and process them, rather than numbers, since the models are based on mathematical equations and calculations. Therefore, we have to encode the categorical data.

Now we need to split our dataset into two sets — a Training set and a Test set. We will train our machine learning models on our training set, i.e our machine learning models will try to understand any correlations in our training set and then we will test the models on our test set to check how accurately it can predict. A general rule of the thumb is to allocate 80% of the dataset to training set and the remaining 20% to test set. For this task, we will import `test_train_split` from `model_selection` library of scikit.

## **FEATURE ENGINEERING**

Filter methods are generally used as a preprocessing step. The selection of features is independent of any machine learning algorithms. Instead, features are selected on the basis of their scores in various statistical tests for their correlation with the outcome variable. The correlation is a subjective term here. For basic guidance, you can refer to the following table for defining correlation co-efficients.

**Pearson's Correlation:** It is used as a measure for quantifying linear dependence between two continuous variables X and Y. Its value varies from -1 to +1.

**LDA:** Linear discriminant analysis is used to find a linear combination of features that characterizes or separates two or more classes (or levels) of a categorical variable.

**ANOVA:** ANOVA stands for Analysis of variance. It is similar to LDA except for the fact that it is operated using one or more categorical independent features and one

continuous dependent feature. It provides a statistical test of whether the means of several groups are equal or not.

Chi-Square: It is a statistical test applied to the groups of categorical features to evaluate the likelihood of correlation or association between them using their frequency distribution.

## **PREDICTION**

Once training is complete, it's time to see if the model is any good, using Evaluation. This is where that dataset that we set aside earlier comes into play. Evaluation allows us to test our model against data that has never been used for training. This metric allows us to see how the model might perform against data that it has not yet seen. This is meant to be representative of how the model might perform in the real world.

A good rule of thumb I use for a training-evaluation split somewhere on the order of 80/20 or 70/30. Much of this depends on the size of the original source dataset. If you have a lot of data, perhaps you don't need as big of a fraction for the evaluation dataset.

Once you've done evaluation, it's possible that you want to see if you can further improve your training in any way. We can do this by tuning our parameters. There were a few parameters we implicitly assumed when we did our training, and now is a good time to go back and test those assumptions and try other values.

A tree has many analogies in real life, and turns out that it has influenced a wide area of machine learning, covering both classification and regression. In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. As the name goes, it uses a tree-like model of decisions. A decision tree is drawn upside down with its root at the top. In the image on the left, the bold text in black represents a condition/internal node, based on which the tree splits into branches/ edges. The end of the branch that doesn't split anymore is the decision/leaf, in this case, whether the passenger died or survived, represented as red and green text respectively.

Although, a real dataset will have a lot more features and this will just be a branch in a much bigger tree, but you can't ignore the simplicity of this algorithm. The feature importance is clear and relations can be viewed easily. This methodology is more commonly known as learning decision tree from data and above tree is called Classification tree as the target is to classify passenger as survived or died. Regression trees are represented in the same manner, just they predict continuous values like price of a house. In general, Decision Tree algorithms are referred to as CART or Classification and Regression Trees.

## **CHAPTER 7**

### **SOFTWARE TESTING**

Testing documentation is the documentation of artifacts that are created during or before the testing of a software application. Documentation reflects the importance of processes for the customer, individual and organization. Projects which contain all documents have a high level of maturity. Careful documentation can save the time, efforts and wealth of the organization.

If the testing or development team gets software that is not working correctly and developed by someone else, so to find the error, the team will first need a document. Now, if the documents are available then the team will quickly find out the cause of the error by examining documentation. But, if the documents are not available then the tester need to do black box and white box testing again, which will waste the time and money of the organization. More than that, Lack of documentation becomes a problem for acceptance.

#### **Benefits of using Documentation**

- Documentation clarifies the quality of methods and objectives.
- It ensures internal coordination when a customer uses software application.
- It ensures clarity about the stability of tasks and performance.
- It provides feedback on preventive tasks.
- It provides feedback for your planning cycle.
- It creates objective evidence for the performance of the quality management system.

The test scenario is a detailed document of test cases that cover end to end functionality of a software application in liner statements. The liner statement is considered as a scenario. The test scenario is a high-level classification of testable requirements. These requirements are grouped on the basis of the functionality of a module and obtained from the use cases.

In the test scenario, there is a detailed testing process due to many associated test cases. Before performing the test scenario, the tester has to consider the test cases for each scenario.

In the test scenario, testers need to put themselves in the place of the user because they test the software application under the user's point of view. Preparation of scenarios is the most critical part, and it is necessary to seek advice or help from customers, stakeholders or developers to prepare the scenario.

As per the IEEE Documentation describing plans for, or results of, the testing of a system or component, Types include test case specification, test incident report, test log, test plan, test procedure, test report. Hence the testing of all the above mentioned documents is known as documentation testing.

This is one of the most cost effective approaches to testing. If the documentation is not right: there will be major and costly problems. The documentation can be tested in a number of different ways to many different degrees of complexity. These range from running the documents through a spelling and grammar checking device, to manually reviewing the documentation to remove any ambiguity or inconsistency.

Documentation testing can start at the very beginning of the software process and hence save large amounts of money, since the earlier a defect is found the less it will cost to be fixed.

the most popular testing documentation files are test reports, plans, and checklists. These documents are used to outline the team's workload and keep track of the process. Let's take a look at the key requirements for these files and see how they contribute to the process.

Test strategy. An outline of the full approach to product testing. As the project moves along, developers, designers, product owners can come back to the document and see if the actual performance corresponds to the planned activities.

Test data. The data that testers enter into the software to verify certain features and their outputs. Examples of such data can be fake user profiles, statistics, media content, similar to files that would be uploaded by an end-user in a ready solution.

Test plans. A file that describes the strategy, resources, environment, limitations, and schedule of the testing process. It's the fullest testing document, essential for informed planning. Such a document is distributed between team members and shared with all stakeholders.

Test scenarios. In scenarios, testers break down the product's functionality and interface by modules and provide real-time status updates at all testing stages. A module can be described by a single statement, or require hundreds of statuses, depending on its size and scope.

Test cases. If the test scenario describes the object of testing (what), a scenario describes a procedure (how). These files cover step-by-step guidance, detailed conditions, and current inputs of a testing task. Test cases have their own kinds that depend on the type of testing — functional, UI, physical, logical cases, etc. Test cases compare available resources and current conditions with desired outcomes and determine if the functionality can be released or not.

Traceability Matrix. This software testing documentation maps test cases and their requirements. All entries have their custom IDs — team members and stakeholders can track the progress of any tasks by simply entering its ID to the search.

External documentation collects information from inner documentation but also emphasize on providing a visual data representation — graphs, diagrams, etc.

External reports — these documents collect information on test results and can describe an entire project or a particular piece of functionality.

Test summary report — the file with final test results and findings, presented to stakeholders.

Bug reports — such files keep track of newly encountered bugs and their fixes. We prefer to keep our bug documentation numbered, so it's easier to mention them in further documentation. Reports are concise and focus on offering tangible solutions. Sometimes, bug reports can only include issue description, if the team hasn't yet found the best approach to fixing the problem.

The combination of internal and external documentation is the key to a deep understanding of all testing processes. Although stakeholders typically have access to the majority of documentation, they mostly work with external files, since they are more concise and tackle tangible issues and results. Internal files, on the other hand, are used by team members to optimize the testing process.

Black Box testers don't care about Unit Testing. Their main goal is to validate the application against the requirements without going into the implementation details.

But as a curiosity or Out of the box thinking, have you ever wondered how developers test their own code? What method do they use to test before releasing code for testing? How is dev-testing important in an agile process? The answer to all this is Unit Testing. I want to educate you on the importance of Unit Testing so that development and testing teams can work more collaboratively to design, test and release an excellent application.

Unit Testing is not a new concept. It's been there since the early days of programming. Usually, developers and sometimes White box testers write Unit tests to improve code quality by verifying each and every unit of the code used to implement functional requirements (aka test driven development TDD or test-first development).

Most of us might know the classic definition of Unit Testing

—Unit Testing is the method of verifying the smallest piece of testable code against its purpose. If the purpose or requirement failed then the unit test has failed.

In simple words, Unit Testing means – writing a piece of code (unit test) to verify the code (unit) written for implementing requirements.



Unit Testing is used to design robust software components that help maintain code and eliminate the issues in code units. We all know the importance of finding and fixing defects in the early stage of the software development cycle. Unit Testing serves the same purpose.

Unit Testing is an integral part of the agile software development process. When a nightly build run unit test suite should run and report should be generated. If any of the unit tests have failed then the QA team should not accept that build for verification.

If we set this as a standard process, many defects would be caught in the early development cycle, saving much testing time.

I know many developers hate to write unit tests. They either ignore or write bad unit test cases due to tight scheduled or lack of seriousness (yea they write empty unit tests, so 100% of them pass successfully ;-)). It's important to write good unit tests or don't write them at all. It's even more important to provide sufficient time and a supportive environment for real benefits.

- Testing can be done in the early phases of the software development lifecycle when other modules may not be available for integration.
- Fixing an issue in Unit Testing can fix many other issues occurring in later

development and testing stages.

- Cost of fixing a defect found in Unit Testing is very less than the one found in the system or acceptance testing.
- Code coverage can be measured
- Fewer bugs in the System and Acceptance testing
- Code completeness can be demonstrated using unit tests. This is more useful in the agile process. Testers don't get the functional builds to test until integration is completed. Code completion cannot be justified by showing that you have written and checked in the code. But running Unit tests can demonstrate code completeness.
- Expect robust design and development as developers write test cases by understanding the specifications first.
- Easily identify who broke the build.
- Saves development time: Code completion may take more time but due to decreased defect count overall development time can be saved.

## **GENERAL**

Unit Testing frameworks are mostly used to help write unit tests quickly and easily. Most of the programming languages do not support unit testing with the inbuilt compiler. Third-party open source and commercial tools can be used to make unit testing even more fun.

List of popular Unit Testing tools for different programming languages:

- Java framework – JUnit
- PHP framework – PHPUnit
- C++ frameworks – UnitTest++ and Google C++
- .NET framework – NUnit
- Python framework – py.test



Functional Testing types include:

1. Unit Testing
2. Integration Testing
3. System Testing
4. Sanity Testing
5. Smoke Testing
6. Interface Testing
7. Regression Testing
8. Beta/Acceptance Testing
9. Non-functional Testing types include:

Performance Testing

- Load Testing
- Stress Testing
- Volume Testing
- Security Testing
- Compatibility Testing
- Install Testing
- Recovery Testing
- Reliability Testing
- Usability Testing
- Compliance Testing
- Localization Testing

## **UNIT TESTING**

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.

A unit can be almost anything you want it to be -- a line of code, a method, or a class. Generally though, smaller is better. Smaller tests give you a much more granular view of how your code is performing. There is also the practical aspect that when you test very small units, your tests can be run fast; like a thousand tests in a second fast.

Black Box testers don't care about Unit Testing. Their main goal is to validate the application against the requirements without going into the implementation details.

Unit Testing is not a new concept. It's been there since the early days of programming. Usually, developers and sometimes White box testers write Unit tests to improve code quality by verifying each and every unit of the code used to implement functional requirements (aka test drove development TDD or test-first development).

Most of us might know the classic definition of Unit Testing –

—Unit Testing is the method of verifying the smallest piece of testable code against its purpose. If the purpose or requirement failed then the unit test has failed.

In simple words, Unit Testing means – writing a piece of code (unit test) to verify the code (unit) written for implementing requirements.

## **BLACKBOX TESTING**

During functional testing, testers verify the app features against the user specifications. This is completely different from testing done by developers which is unit testing. It checks whether the code works as expected. Because unit testing focuses on the internal structure of the code, it is called the white box testing. On the other hand, functional testing checks app's functionalities without looking at the internal structure of the code, hence it is called black box testing. Despite how flawless the various individual code components may be, it is essential to check that

the app is functioning as expected, when all components are combined. Here you can find a detailed comparison between functional testing vs unit testing.

## **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.

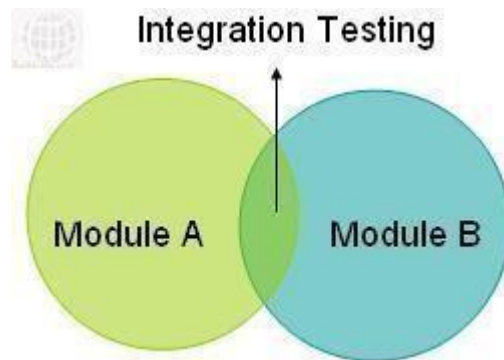
integration testing: Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems. See also component integration testing, system integration testing.

component integration testing: Testing performed to expose defects in the interfaces and interaction between integrated components. System integration testing: Testing the integration of systems and packages; testing interfaces to external organizations (e.g. Electronic Data Interchange, Internet).

Integration tests determine if independently developed units of software work correctly when they are connected to each other. The term has become blurred even by the diffuse standards of the software industry, so I've been wary of using it in my writing. In particular, many people assume integration tests are necessarily broad in scope, while they can be more effectively done with a narrower scope.

As often with these things, it's best to start with a bit of history. When I first learned about integration testing, it was in the 1980's and the waterfall was the dominant influence of software development thinking. In a larger project, we would have a design phase that would specify the interface and behavior of the various modules in the system. Modules would then be assigned to developers to program. It was not unusual for one programmer to be responsible for a single module, but this would be big enough that it could take months to build it. All this work was done in isolation, and when the programmer believed it was finished they would hand it over to QA for testing.

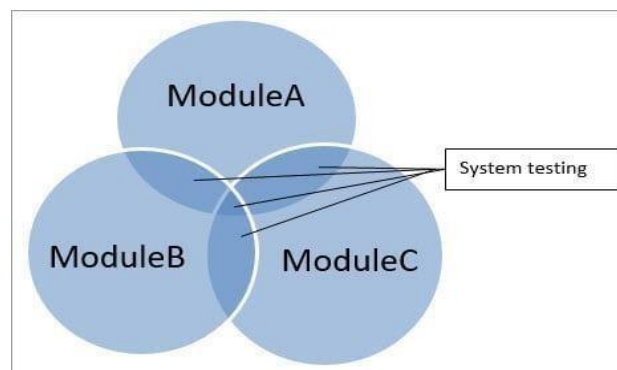
Integration testing tests integration or interfaces between components, interactions to different parts of the system such as an operating system, file system and hardware or interfaces between systems. Integration testing is a key aspect of software testing.



## SYSTEM TESTING

SYSTEM TESTING is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements. System Testing means testing the system as a whole. All the modules/components are integrated in order to verify if the system works as expected or not.

System Testing is done after Integration Testing. This plays an important role in delivering a high-quality product.



System testing is a method of monitoring and assessing the behaviour of the complete and fully-integrated software product or system, on the basis of pre-decided specifications and functional requirements. It is a solution to the question "whether the complete system functions in accordance to its pre-defined requirements?"

It's comes under black box testing i.e. only external working features of the software are evaluated during this testing. It does not requires any internal knowledge of the coding, programming, design, etc., and is completely based on users-perspective.

A black box testing type, system testing is the first testing technique that carries out the task of testing a software product as a whole. This System testing tests the integrated system and validates whether it meets the specified requirements of the client.

System testing is a process of testing the entire system that is fully functional, in order to ensure the system is bound to all the requirements provided by the client in the form of the functional specification or system specification documentation. In most cases, it is done next to the Integration testing, as this testing should be covering the end-to-end system's actual routine. This type of testing requires a dedicated Test Plan and other test documentation derived from the system specification document that should cover both software and hardware requirements. By this test, we uncover the errors. It ensures that all the system works as expected. We check System performance and functionality to get a quality product. System testing is nothing but testing the system as a whole. This testing checks complete end-to-end scenario as per the customer's point of view. Functional and Non-Functional tests also done by System testing. All things are done to maintain trust within the development that the system is defect-free and bug-free. System testing is also intended to test hardware/software requirements specifications. System testing is more of a limited type of testing; it seeks to detect both defects within the -inter-assemblagesll.

## **SANITY TESTING**

Sanity Testing is done when as a QA we do not have sufficient time to run all the test cases, be it Functional Testing, UI, OS or Browser Testing. Sanity testing is a subset of regression testing. After receiving the software build, sanity testing is performed to ensure that the code changes introduced are working as expected. This testing is a checkpoint to determine if testing for the build can proceed or not.

The main purpose of this testing is to determine that the changes or the proposed functionality are working as expected. If the sanity test fails, the build is rejected by the testing team to save time and money. It is performed only after the build has cleared the smoke test and been accepted by the Quality Assurance team for further testing. The focus of the team during this testing process is to validate the functionality of the application and not detailed testing. Let's assume that in an e-commerce project, there are mainly five modules- like login page, home page, user profile page, new user creation, etc. There is a defect in the login page when the password field accepts less than four alpha numeric characters and the requirement mentions that this password field should not be below eight characters. So, the bug is reported by the testing team to the development team to resolve it. When the development team fixes the defect and passes it to the testing team. Then the testing team checks if the changes done are working fine. It also checks if it does not have an impact on other related functionalities. Now there is a functionality to update the password in the user profile page. As part of the sanity check, we would need to validate the login page as well as the profile page to ensure that the checks are working fine at both the places. Sanity tests are generally performed on build where the production deployment is required immediately - like a critical bug fix.

Smoke Testing is done to make sure if the build we received from the development team is testable or not. It is also called as —Day 0ll check. It is done at the —build levelll.

It helps not to waste the testing time to simply testing the whole application when the key features don't work or the key bugs have not been fixed yet. Here our focus will be on primary and core application work flow.

To conduct smoke testing, we don't write test cases. We just pick the necessary test cases from already written test cases.

Do we really write test cases for all testing types? Here in this article, we have given clear idea on choosing testing types to write test cases.

As mentioned earlier, here in Smoke Testing, our main focus will be on core application work flow. So we pick the test cases from our test suite which cover major functionality of the application. In general, we pick minimal number of test cases that won't take more than half an hour to execute.

After receiving a Software build with the minor issues fixes in code or functionality, Sanity testing is carry out to check whether the bugs reported in previous build are fixed & there is regression introduced due to these fixes i.e. not breaking any previously working functionality. The main aim of Sanity testing to check the planned functionality is working as expected. Instead of doing whole regression testing the Sanity testing is perform.

Sanity tests helps to avoid wasting time and cost involved in testing if the build is failed. Tester should reject the build upon build failure.

After completion of regression testing the Sanity testing is started to check the defect fixes & changes done in the software application is not breaking the core functionality of the software. Typically this is done nearing end of SDLC i.e. while releasing the software. You can say that sanity testing is a subset of acceptance testing. We can also say Tester Acceptance Testing for Sanity testing.

## **REGRESSION TESTING**

Regression Testing is a type of testing that is done to verify that a code change in the software does not impact the existing functionality of the product. This is to make sure the product works fine with new functionality, bug fixes or any change in the existing feature. Previously executed test cases are re-executed in order to verify the impact of change.

Regression Testing is a Software Testing type in which test cases are re-executed in order to check whether the previous functionality of the application is working fine and the new changes have not introduced any new bugs.

This test can be performed on a new build when there is a significant change in the original functionality that too even in a single bug fix.

For regression testing to be effective, it needs to be seen as one part of a comprehensive testing methodology that is cost-effective and efficient while still incorporating enough variety—such as well-designed frontend UI automated tests alongside targeted unit testing, based on smart risk prioritization—to prevent any aspects of your software applications from going unchecked. These days, many Agile work environments employing workflow practices such as XP (Extreme Programming), RUP (Rational Unified Process), or Scrum appreciate regression testing as an essential aspect of a dynamic, iterative development and deployment schedule.

But no matter what software development and quality-assurance process your organization uses, if you take the time to put in enough careful planning up front, crafting a clear and diverse testing strategy with automated regression testing at its core, you can help prevent projects from going over budget, keep your team on track, and, most importantly, prevent unexpected bugs from damaging your products and your company's bottom line.

Performance testing is the practice of evaluating how a system performs in terms of responsiveness and stability under a particular workload. Performance tests are typically executed to examine speed, robustness, reliability, and application size. The process incorporates -performance indicators such as:

- Browser, page, and network response times
- Server request processing times
- Acceptable concurrent user volumes
- Processor memory consumption; number and type of errors that might be encountered with app

Performance testing gathers all the tests that verify an application's speed, robustness, reliability, and correct sizing. It examines several indicators such as a browser, page and network response times, server query processing time, number of acceptable concurrent users architected, CPU memory consumption, and number/type of errors which may be encountered when using an application.

Performance testing is the testing that is performed to ascertain how the components of a system are performing under a certain given situation.

Resource usage, scalability, and reliability of the product are also validated under this testing. This testing is the subset of performance engineering, which is focused on addressing performance issues in the design and architecture of a software product.



Software Performance testing is type of testing perform to determine the performance of system to major the measure, validate or verify quality attributes of the system like responsiveness, Speed, Scalability, Stability under variety of load conditions. The system is tested under a mixture of load conditions and check the time required responding by the system under varying workloads. Software performance testing involves the testing of application under test to ensure that application is working as expected under variety of load conditions. The goal of performance testing is not only find the bugs in the system but also eliminate the performance bottlenecks from the system.

Load Testing is type of performance testing to check system with constantly increasing the load on the system until the time load is reaches to its threshold value. Here Increasing load means increasing number of concurrent users, transactions & check the behavior of application under test. It is normally carried out underneath controlled environment in order to distinguish between two different systems. It is also called as —Endurance testingll and —Volume testingll. The main purpose of load testing is to monitor the response time and staying power of application when system is performing well under heavy load. Load testing comes under the Non Functional Testing & it is designed to test the non-functional requirements of a software application.

Load testing is perform to make sure that what amount of load can be withstand the application under test. The successfully executed load testing is only if the specified test cases are executed without any error in allocated time.

Simple examples of load testing:

- Testing printer by sending large job.
- Editing a very large document for testing of word processor
- Continuously reading and writing data into hard disk.
- Running multiple applications simultaneously on server.
- Testing of mail server by accessing thousands of mailboxes.
- In case of zero-volume testing & system fed with zero load.

## ➤ **REQUIREMENT ANALYSIS**

Requirement analysis, also called requirement engineering, is the process of determining user expectations for a new modified product. It encompasses the tasks that determine the need for analysing, documenting, validating and managing software or system requirements. The requirements should be documentable, actionable, measurable, testable and traceable related to identified business needs or opportunities and define to a level of detail, sufficient for system design.

## ➤ **FUNCTIONAL REQUIREMENTS**

It is a technical specification requirement for the software products. It is the first step in the requirement analysis process which lists the requirements of particular software systems including functional, performance and security requirements. The function of the system depends mainly on the quality hardware used to run the software with given functionality.

- **Usability**

It specifies how easy the system must be use. It is easy to ask queries in any format which is short or long, porter stemming algorithm stimulates the desired response for user.

- **Robustness**

It refers to a program that performs well not only under ordinary conditions but also under unusual conditions. It is the ability of the user to cope with errors for irrelevant queries during execution.

- **Security**

The state of providing protected access to resource is security. The system provides good security and unauthorized users cannot access the system there by providing high security.

- **Reliability**

It is the probability of how often the software fails. The measurement is often expressed in MTBF (Mean Time Between Failures). The requirement is

needed in order to ensure that the processes work correctly and completely without being aborted. It can handle any load and survive and survive and even capable of working around any failure.

- **Compatibility**

It is supported by version above all web browsers. Using any web servers like localhost makes the system real-time experience.

- **Flexibility**

The flexibility of the project is provided in such a way that it has the ability to run on different environments being executed by different users.

- **Safety**

Safety is a measure taken to prevent trouble. Every query is processed in a secured manner without letting others to know one's personal information.

## **NON- FUNCTIONAL REQUIREMENTS**

- **Portability**

It is the usability of the same software in different environments. The project can be run in any operating system.

- **Performance**

These requirements determine the resources required, time interval, throughput and everything that deals with the performance of the system.

- **Accuracy**

The result of the requesting query is very accurate and high speed of retrieving information. The degree of security provided by the system is high and effective.

- **Maintainability**

Project is simple as further updates can be easily done without affecting its stability. Maintainability basically defines that how easy it is to maintain the system. It means that how easy it is to maintain the system, analyse, change and test the application. Maintainability of this project is simple as further updates can be easily done without affecting its stability.

## **CHAPTER 8**

### **CONCLUSION & FUTURE ENCHANCEMENT**

#### **CONCLUSION**

In this paper, the classification task is used on student database to predict the student's decision on the basis of previous database. As there are many approaches that are used for data classification, the k-means clustering Classifier and KNN Classifier are used here. Information's like marks, interest were collected from the student's previous database, to predict the training course or a career and 12<sup>th</sup> groups at the end. This study will help to the students to improve the decision of the student. This can help the students improve in their academics, which eventually leads to a good decision in their career.

#### **FUTURE ENHANCEMENT**

## SCREENSHOTS

Final report for higher education X (1) #Uppena - Jala Jala Jalec X localhost:8080/school-to-intermediate X

localhost:8080/school-to-intermediate

School to Intermediate (+2)

Intermediate (+2) to College

Career Development

### School to Intermediate (+2) suggestion

|             |    |           |    |
|-------------|----|-----------|----|
| Mathamatics | 90 | Physics   | 85 |
| Biology     | 80 | Chemistry | 86 |
| Social      | 88 |           |    |

[Suggest](#) [Reset](#)

MPC is recommended

Suggested colleges are

1. Sree chaitanya Junior college
2. Narayana Junior college

Type here to search

12:35 03-03-2021

Final report for higher education X (1) #Uppena - Jala Jala Jalec X localhost:8080/school-to-intermediate X

localhost:8080/school-to-intermediate

School to Intermediate (+2)

Intermediate (+2) to College

Career Development

### School to Intermediate (+2) suggestion

|             |    |           |    |
|-------------|----|-----------|----|
| Mathamatics | 70 | Physics   | 90 |
| Biology     | 89 | Chemistry | 88 |
| Social      | 85 |           |    |

[Suggest](#) [Reset](#)

MPC / BIPC is recommended

Suggested colleges are

1. Sree chaitanya Junior college
2. Narayana Junior college

Type here to search

12:35 03-03-2021

Final report for higher education x (1) #Uppena - Jala Jala Jala: x localhost:8080/school-to-intermediate x

localhost:8080/school-to-intermediate

School to Intermediate (+2)

Intermediate (+2) to College

Career Development

## School to Intermediate (+2) suggestion

|             |    |           |    |
|-------------|----|-----------|----|
| Mathematics | 70 | Physics   | 90 |
| Biology     | 92 | Chemistry | 88 |
| Social      | 85 |           |    |

Suggest Reset

BIPC is recommended

Suggested colleges are

1. Narayana Junior college
2. Sree chaitanya Junior college

Type here to search

12:36 03-03-2021

Final report for higher education x (1) #Uppena - Jala Jala Jala: x localhost:8080/school-to-intermediate x

localhost:8080/school-to-intermediate

School to Intermediate (+2)

Intermediate (+2) to College

Career Development

## School to Intermediate (+2) suggestion

|             |    |           |    |
|-------------|----|-----------|----|
| Mathematics | 70 | Physics   | 82 |
| Biology     | 80 | Chemistry | 88 |
| Social      | 90 |           |    |

Suggest Reset

CEC is recommended

Suggested colleges are

1. Roots Junior college
2. St.Mary's Junior college

Type here to search

12:36 03-03-2021

Final report for higher education x (1) #Uppena - Jala Jala Jala: x localhost:8080/intermediate-to- x

localhost:8080/intermediate-to-college

School to Intermediate (+2)

Intermediate (+2) to College

Career Development

## Intermediate (+2) to College suggestion

|             |    |           |    |
|-------------|----|-----------|----|
| Mathematics | 89 | Physics   | 85 |
| Electronics | 86 | Chemistry | 87 |
| Programming | 90 |           |    |

[Suggest](#) [Reset](#)

CSE / IT are recommended

Suggested colleges are

1. Sathyabhama Institute of technology and sciences
2. Hindustan Institute of technology and sciences

Type here to search

12:37 03-03-2021

Final report for higher education x (1) #Uppena - Jala Jala Jala: x localhost:8080/intermediate-to- x

localhost:8080/intermediate-to-college

School to Intermediate (+2)

Intermediate (+2) to College

Career Development

## Intermediate (+2) to College suggestion

|             |    |           |    |
|-------------|----|-----------|----|
| Mathematics | 89 | Physics   | 85 |
| Electronics | 90 | Chemistry | 87 |
| Programming | 70 |           |    |

[Suggest](#) [Reset](#)

ECE / EEE / EIE are recommended

1. Saastra Institute of technology and sciences
2. SRM Institute of technology and sciences

Type here to search

12:37 03-03-2021

Final report for higher education x (1) #Uppena - lala lala lala x localhost:8080/intermediate-to- x

localhost:8080/intermediate-to-college

School to Intermediate (+2)

Intermediate (+2) to College

Career Development

## Intermediate (+2) to College suggestion

|             |    |           |    |
|-------------|----|-----------|----|
| Mathamatics | 89 | Physics   | 85 |
| Electronics | 70 | Chemistry | 87 |
| Programming | 70 |           |    |

[Suggest](#) [Reset](#)

AI / ECE are recommended

Suggested colleges are

1. Sathyabhama Institute of technology and sciences
2. Saastra Institute of technology and sciences

Type here to search

12:37 03-03-2021

Final report for higher education x (1) Master - Polakuturi Para x localhost:8080/career-developm x

localhost:8080/career-development

School to Intermediate (+2)

Intermediate (+2) to College

Career Development

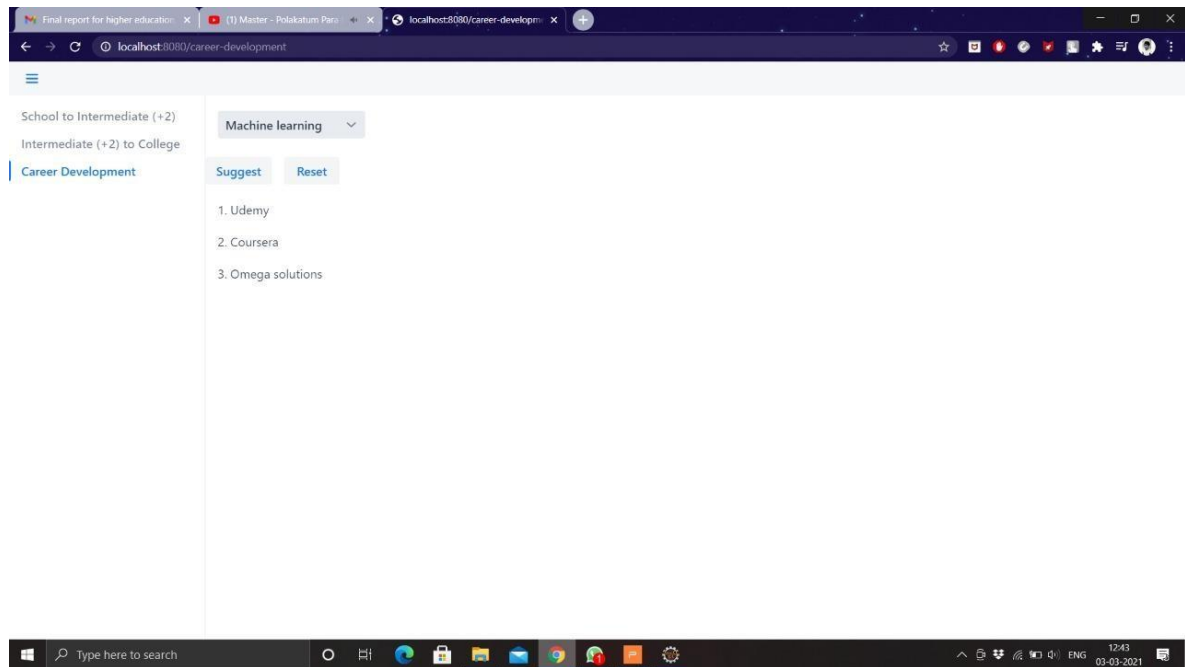
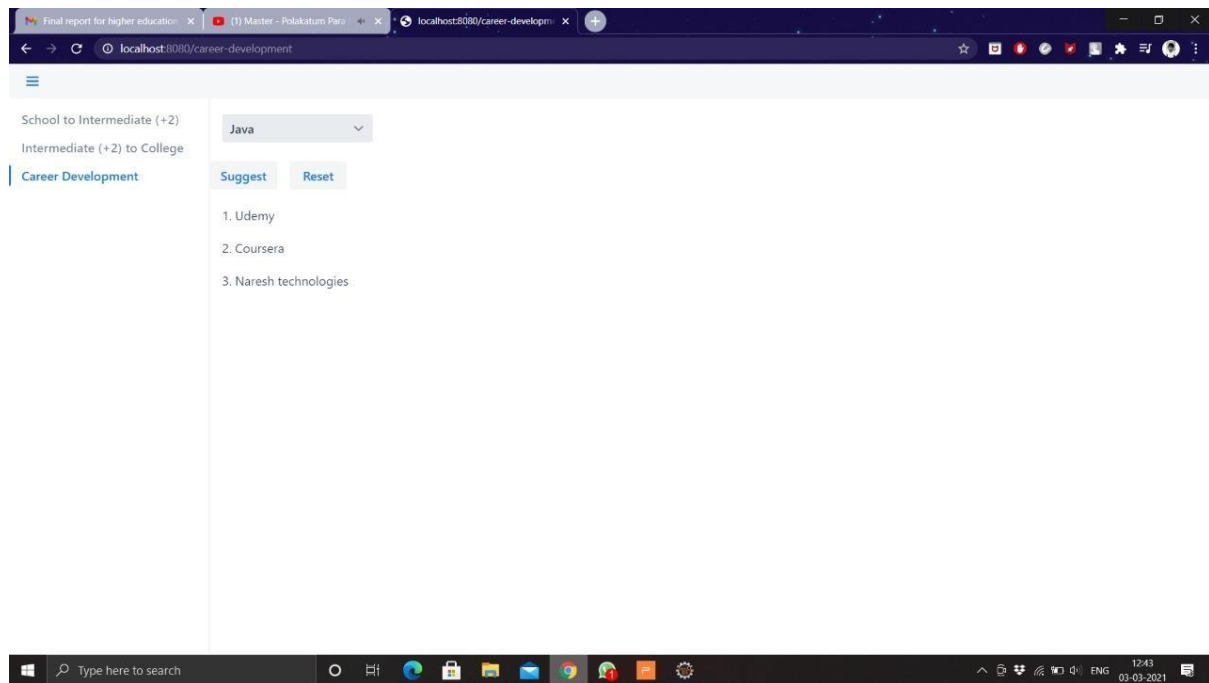
Python

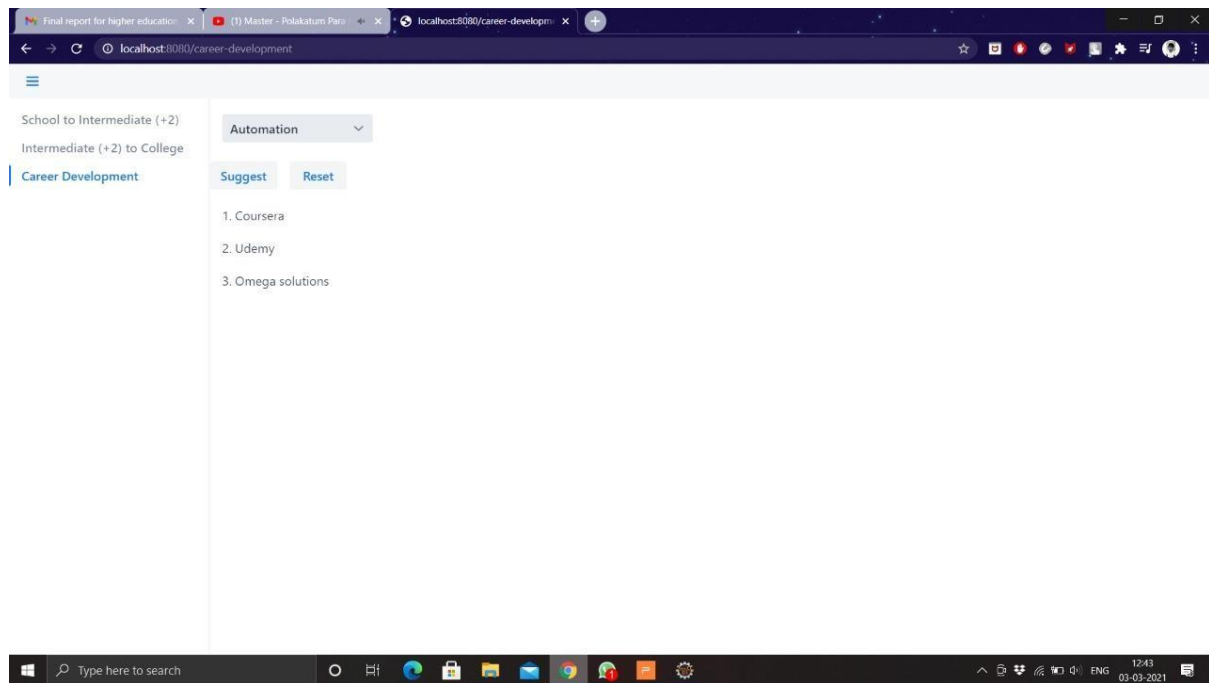
[Suggest](#) [Reset](#)

1. Udeemy
2. Coursera

Type here to search

12:43 03-03-2021





## REFERENCES

- [1] A. Lašáková, L. Bajžíková, and I. Dedze, „Barriers and drivers of innovation in higher education: Case study-based evidence across ten European Universities,“ *Int. J. Educ. Develop.*, vol. 55, pp. 69–79, May 2017.
- [2] Y. V. N. Acevedo and C. E. M. Marín, „System architecture based on learning analytics to educational decision makers toolkit,“ *Adv. Comput. Sci. Eng.*, vol. 13, no. 2, pp. 89–105, 2014.
- [3] A. R. T. Góes, M. T. A. Steiner, and P. J. S. Neto, „Education quality measured by the classification of school performance using quality labels,“ *Appl. Mech. Mater.*, vols. 670–671, pp. 1675–1683, Oct. 2014.
- [4] H. Lounis and T. Fares, „Using efficient machine-learning models to assess two important quality factors: Maintainability and reusability,“ in *Proc. Joint Conf. 21st Int. Workshop Softw. Meas. 6th Int. Conf. Softw. Process Product Meas.*, Nov. 2011, pp. 170–177.
- [5] Y. Nieto, V. García-Díaz, C. Montenegro, and R. G. Crespo, „Supporting academic decision making at higher educational institutions using machine learning-based algorithms,“ *Soft Comput.*, vol. 23, no. 12, pp. 4145–4153, 2018.
- [6] K. V. Pincus, D. E. Stout, J. E. Sorensen, K. D. Stocks, and R. A. Lawson, „Forces for change in higher education and implications for the accounting academy,“ *J. Accounting Educ.*, vol. 40, pp. 1–18, Sep. 2016.
- [7] A.-P. Pavel, A. Fruth, and M.-N. Neacsu, „ICT and e-learning—Catalysts for innovation and quality in higher education,“ *Procedia Econ. Finance*, vol. 23, pp. 704–711, Jan. 2015.
- [8] F. V. Elena, A. M. Manuel, and G. G. S. Carina, „Which use give teachers at La Laguna University to ICTs?“, *Procedia-Social Behav. Sci.*, vol. 93, pp. 1646–1651, Oct. 2013.

- [9] H. M. Vo, C. Zhu, and N. A. Diep, "The effect of blended learning on student performance at course-level in higher education: A meta-analysis," *Stud. Educ. Eval.*, vol. 53, pp. 17–28, Jun. 2017.
- [10] I. M. Taucean and M. Tamasila, "Research challenges for eLearning support in engineering and management training," *Procedia-Social Behav. Sci.*, vol. 124, pp. 210–218, Mar. 2014.
- [11] E. Faham, A. Rezvanfar, S. H. M. Mohammadi, and M. R. Nohooji, "Using system dynamics to develop education for sustainable development in higher education with the emphasis on the sustainability competencies of students," *Technol. Forecast. Social Change*, vol. 123, pp. 307–326, Oct. 2017.
- [12] F. A. Goni, A. G. Chofreh, M. Mukhtar, S. Sahran, S. A. Shukor, and J. J. Klemeš, "Strategic alignment between sustainability and information systems: A case analysis in Malaysian public higher education Institutions," *J. Clean. Prod.*, vol. 168, pp. 263–270, Dec. 2017.
- [13] I. González-González and A. I. Jiménez-Zarco, "Using learning methodologies and resources in the development of critical thinking competency: An exploratory study in a virtual learning environment," *Comput. Hum. Behav.*, vol. 51, pp. 1359–1366, Oct. 2015.