**Academic Performance prediction system**

4.0 Introduction........................................................................................................................... 38

4.1 Overview of the Proposed System ....................................................................................... 38

4.2 General Requirements.......................................................................................................... 39

4.2.1 Functional Requirements for Computerized Product Distribution Monitory System 39

4.2.1.1 Input Design .................................................................................................................... 40

4.2.1.2 Output Design.................................................................................................................. 40

4.2.1.3 Logical Design ................................................................................................................. 40

4.2.2 Non-Functional Requirements for a Computerized Product Distribution Monitory

System .......................................................................................................................................... 41

4.2.3 Software Requirement Specification................................................................................ 43

4.2.4 Hardware Requirement Specification.............................................................................. 43

4.3 System Design........................................................................................................................ 44

4.3.1 Feasibility Study................................................................................................................. 44

4.3.1.1 Technical Feasibility ....................................................................................................... 45

4.3.1.2 Economic Feasibility....................................................................................................... 46

4.3.1.3 Operational Feasibility ................................................................................................... 46

4.3.1.4 Schedule Feasibility ........................................................................................................ 47

4.3.1.5 Resource Feasibility........................................................................................................ 47

4.4 Machine Learning ………………………………………..

4.5 Data Flow Diagram (DFD)................................................................................................... 48

4.5.1 Context Dataflow Diagram.......................................................................................... 50

4.6 Use Case Design..................................................................................................................... 51

4.6.1 Use Case Diagram for a Computerized Product Distribution Monitory System ........ 51

4.6.2 Use Case Diagram.............................................................................................................. 52

4.7 Entity Relationship Diagram for a Computerized Product Distribution Monitory System ..... 53

4.7.1 Entity Relationship Diagram ............................................................................................ 54

4.8 Flowchart Design of Computerized Product Distribution Monitory System.................. 55

4.8.1 Flowchart Symbols............................................................................................................. 55

4.8.2 Flowchart Diagram............................................................................................................ 56

4.8.3 Sequence Diagram ............................................................................................................. 57

4.8.4 Sequence Diagram of the Academic Performance System…………………………………

4.9 UML Activity Diagram …………………………………………………………………………………………………

4.10 System Implementation........................................................................................................ 58

4.10.1 System Testing.................................................................................................................... 58

4.10.2 Software Testing Strategies............................................................................................... 59

4.10.3 System Security .................................................................................................................. 60

4.11 Conclusion ............................................................................................................................. 61

f2.3emp

f4.4 machine learnin phase

f4.5.1 context dataflow

f4.6.2use case

f4.7.1entity relation

f4.8.2a flowchart model

f4.8.2b flowchart prediction system

f4.8.4 sequence diagram

f4.9 uml activity diagram

f5.1ahome interface

f5.1breport interface

t4.1software

t4.2hard requ

t4.3feasibility study

t4.5 data flow d symb

t4.8.1 flowchart d symb

CHAPTER FIVE

FINDINGS, CONCLUSION AND RECOMMENDATIONS

5.0 Introduction........................................................................................................................... 62

5.1 User Interface ........................................................................................................................ 62

5.2 Programming Languages..................................................................................................... 63

5.2.1 Why Using Python ................................................................................................. 64

5.3 Implementation ..................................................................................................................... 64

5.4 Conclusion of the Study........................................................................................................ 65

5.5 Maintenance .......................................................................................................................... 65

5.6 Implications........................................................................................................................... 66

5.7 Future Works and Recommendation.................................................................................. 67

5.8 Conclusion ............................................................................................................................. 67

REFERENCES............................................................................................................................ 68

APPENDICES............................................................................................................................. 71

**CHAPTER FOUR**

**SYSTEM ANALYSIS AND DESIGN**

**4.0 INTRODUCTION**

The process of determining user expectations for a new or modified product is known as requirements analysis. These characteristics, known as requirements, must be quantifiable, relevant, and detailed. Such requirements are frequently referred to as functional specifications in software engineering. The analysis of requirements is an important part of any project.

Requirements analysis entails frequent communication with system users to determine specific feature expectations, resolving conflict or ambiguity in requirements as demanded by various users or groups of users, avoiding feature creep, and documenting all aspects of the project development process from beginning to end.

Requirements analysis is a collaborative effort that necessitates knowledge of hardware, software, and human factors engineering, as well as interpersonal skills.

**4.1 OVERVIEW OF THE PROPOSED SYSTEM**

Due to the large volume of data in educational databases, predicting student performance is a critical job. This task is being handled by educational data mining (EDM). EDM creates methods for discovering data derived from the educational environment. These techniques are used to comprehend students and their learning environments. Educational institutions are frequently interested in how many students will pass/fail for necessary arrangements. Previous research has found that many researchers are intent on selecting an appropriate algorithm for just classification while ignoring solutions to problems encountered during data mining phases such as data high dimensionality, class imbalance, and classification error, among others. These types of issues reduced the model's accuracy. In this domain, several well-known classification algorithms are used, but this paper proposed a student performance prediction model based on supervised learning decision tree classifiers. In addition, an ensemble method is used to improve the classifier's performance. The ensemble methods approach is intended to address classification and prediction problems.

**4.2 GENERAL REQUIREMENTS**

In order to implement an efficient and effective system one needs to understand the business and technical requirements of the implementation process. The main requirement of the proposed system is categorized into:

1. Functional requirement

2. Non-functional requirement

3. Software Requirement Specification

4. Hardware Requirement Specification

**4.2.1 Functional Requirement for Academic Prediction System**

Functional requirements are system features that developers must include in order for users of the system to achieve their objectives. They define the fundamental system behavior under certain conditions.

**4.2.1.1 Input Design**

User-oriented inputs are transformed into a computer-based system format in the input design. The design of menus and prompts is a key component of input design. The user's options are predefined for each possibility. Logical flow, data storage, source, and destination are shown in the data flow diagram. The gathering and grouping of input data into groups of related data.

**4.2.1.2 Output Design**

The goal of the output design is to show output on a screen in a specific format. The screen and audio speakers are the most crucial output devices for this system. The capabilities of the output device, the necessary response times, etc. are taken into account. In order to provide users with information that is clear, precise, and quick, the form design elaborates on how output is displayed and the layout accessible for information capture.

**4.2.1.3 Logical Design**

The logically suggested data is what logical data design is all about. It is possible to construct a form so that the meaning may be understood by each and every piece of data. A clear knowledge and notion of the associated data required to build a form and process information and events should be provided by logical data design.

Few functional requirements of the system shall perform;

1. The system shall create an interface for users to be authenticated in order to proceed registration.
2. The system shall create an interface for users to log into the system.
3. The system shall authenticate provided user information before granting access to user.
4. The system shall allow logged in users to update their account details.
5. The system shall identify each login user and direct them to their corresponding homepage.
6. The system shall allow logged in students to view their current academic performance
7. The system shall allow administration to monitor users’ activities
8. The system shall allow administration to add new user
9. The system shall allow the recovery of account credentials.
10. The system shall generate reports for each student’s performance.

**4.2.2 Non-Functional Requirement**

Non-functional requirements are descriptions of how the system should operate; they list all current requirements that the functional requirement does not address. These nonfunctional criteria include things like;

1. Usability: The simplicity of this system's operation is its most crucial criterion. The system is designed in such a manner that users who are unfamiliar with the technical details of computer systems would find it easy to use and comfortable.
2. Maintainability: The suggested solution is highly adaptable to new trends and platform development.
3. Interoperability: This is the application's capacity to work with many versions of Windows, including Windows 7, Linux, and MAC, which are the most recent trends in computer system devices.

Few non-functional requirements of the system shall perform;

1. The system should present to customer the required feature at any given time.
2. The system should present text using readable font style i.e., no demographic font.
3. The system should authenticate login customer in less than 5 second with an internet speed greater or equal to 200kbps.
4. The system should allow customer to login with their username and password.
5. The system should be accessible to end user at any location in the world.
6. The system should be available at every hour of the day.
7. The system should be responsive to any user device.
8. The system should sort post by time and date.
9. The system should display the date and time a post is created.
10. The system’s functionalities should work on every recent browser.

**4.2.3 Software Requirement**

The developed system is not platform dependent in order to effectively solve the problem that led to the development of this project. As a result, it can run on any platform. The reviews of the developed system's specifications are listed below.

|  |  |
| --- | --- |
| **Operating System** | Windows OS, Linux, and Mac OS (Python must be installed on the selected Operating System used.) |
| **Front-End Technologies** | HTML5, CSS3, JavaScript |
| **Server-side Technologies** | Python3, Flask, WebSocket, Event Emitter |
| **Server** | Python3 |
| **Back-End Tools** | CSV, SQLite, Jupiter Notebook |

*Table 4.1 Software Requirement Specification*

**4.2.4 Hardware Requirement**

|  |  |
| --- | --- |
| **Devices** | Desktop Devices, Laptop |
| **Processor** | x64 Processor: AMD Opteron, AMD Athlon 64, Intel Xeon with Intel EM64T support, Intel Pentium IV with EM64T support or higher |
| **Memory** | Minimum of 1GB RAM |
| **Storage** | Minimum 1GB |

*Table 4.2 Software Requirement Specification*

**4.3 SYSTEM DESIGN**

The process of defining the elements of a system such as the architecture, modules, and components, the various interfaces of those components, and the data that flows through that system is known as system design (Gates 2007). It is intended to meet specific needs and requirements of a business or organization by engineering a coherent and well-functioning system.

A systematic approach to system design is implied by the term systems design. It may take a bottom-up or top-down approach, but in either case, the process is systematic in that it considers all related variables of the system that needs to be created—from the architecture to the necessary hardware and software, all the way down to the data and how it travels and transforms throughout its journey through the system. Then there's systems analysis, systems engineering, and systems architecture to consider.

When engineers were attempting to solve complex control and communications problems just before World War II, the systems design approach first appeared. They needed to be able to formalize their work into a formal discipline with proper methods, particularly in new fields such as information theory, operations research and computer science generally

**4.3.1 FEASIBILITY STUDY**

A feasibility study is simply an evaluation of the viability of a proposed project plan or method. This is accomplished through an examination of technical, economic, legal, operational, and time feasibility factors. As the name implies, you're wondering, "Is this possible?" Do you, for example, have or can you develop the technology to do what you propose? Do you have the necessary people, tools, and resources? And, will the project generate the expected ROI?

Feasibility studies have several advantages, including assisting project owners in determining the pros and cons of undertaking a project before investing significant time and capital into it.

Feasibility studies can also provide critical information to a project owner preventing them from entering into a risky and unattainable venture. They will have a better understanding of how they will operate, what potential obstacles exist, who their competitors are, and what the market is. Feasibility studies can also persuade investors and bankers that investing in a specific project or business is a wise decision.

Feasibility analysis Facts;

1. Technical feasibility
2. Economic feasibility
3. Operational feasibility
4. Schedule feasibility
5. Resource feasibility

**4.3.1.1 Technical Feasibility**

By evaluating the new system's technological viability, one may determine whether it will function as intended and whether the proposed system can be built. The technical evaluation provides information on issues including whether the system's required technology is available, how challenging it will be to construct, and if the knowledge and expertise now held are sufficient to operate it.

The following are some of the technical concerns with feasibility that were brought up during the feasibility stage of the investigation:

1. The software (Visual Studio Code) is running on a windows operating system
2. The minimum hardware required is 2.4 GHz Intel Core 2.
3. The system can be easily expanded.

**4.3.1.2 Economic Feasibility**

Finding out what positive economic advantages the proposed system would provide to the company is the goal of the economic feasibility evaluation. All of the anticipated advantages are identified and quantified. A cost-benefit analysis often forms part of this evaluation. Economic evaluation, which deals with elements that can be measured, quantified, and compared in monetary terms, is an essential component of investment appraisal (Chen 1996). This feasibility analysis compares the development and operating costs to show the prefect's tangible and intangible advantages.

**4.3.1.3 Operational Feasibility**

Operational feasibility is a metric used to assess how successfully a proposed system addresses issues, seizes opportunities, and complies with requirements found during the requirements analysis stage of system development. In terms of development timetable, delivery date, corporate culture, and current business procedures, the operational feasibility study examines how well the planned development projects fit into the existing business environment and objectives.

**4.3.1.4 Schedule Feasibility**

The system is socially feasibly because it can be delivered within the required deadline given our technical expertise.

**4.3.1.5 Resource Feasibility**

This raises issues like how much time is available to develop the new system, when it can be developed, if it obstructs regular company activities, the kind and quantity of resources needed, dependencies, etc. These plans also include mitigation and contingency measures, ensuring that the business is prepared in case the project does go over budget.

|  |  |  |
| --- | --- | --- |
| **Feasibility Questions** | **Answers** | **Comment** |
| With the short time available, can this task be completed successfully? | YES | Depending on the accessibility of the necessary technology and the usability of the selected platforms and technologies, the project can be completed. |
| Are the resources needed for the project on hand? | YES | The project's success can be improved with the help of the available resources. |
| Would the project interfere with ongoing corporate operations? | NO | Because the project may be created outside the firm, it won't in any way interrupt or halt ongoing commercial operations. |
| Are there any backup plans in place in case the system malfunctions to carry on with routine company operations? | NO | When the system malfunctions, users can utilize the manual method until the system is restored since it is not a crucial system (one whose functionality cannot be done manually). |

*Table 4.3 Feasibility Study*

**4.4 Machine Learning**

The brain of machine learning is where all learning occurs. The way a computer learns is comparable to how a person learns. Experience is how people learn. The easier it is to forecast, the more we know. By analogy, our chances of success are lower than they would be in a known circumstance when we encounter one. Machines receive the same training. The computer observes an example in order to create a precise forecast. The machine is capable of predicting the result when we provide a comparable case. However, much like a person, the machine has trouble predicting if it is given a new example.

Learning and inference are at the heart of machine learning. The first way the machine learns is by identifying patterns.

[](https://www.guru99.com/images/tensorflow/082918_1102_WhatisMachi3.png)

*Figure 4.6 Machine learning phase*

**4.5 Data Flow Diagram (DFD)**

Based on Martin and Estrin's "data-flow graph" model of computing, Larry Constantine, the original creator of structured design, created data-flow diagrams.

A data flow diagram (DFD) is a diagram that shows how data moves through a system and how different data is changed. It is a technique for structured analysis and design that may be applied to flowcharting in place of or along with data. System flowcharts that are process-oriented and orientated. Data-flow diagrams (DFDs) were established and made popular for structured analysis and design in the late 1970s (Gane and Sarson 1979). DFDs display the flow of data from external sources into the system as well as how the data was transferred between processes and logically stored.

A data-flow diagram (DFD) shows how data "flows" through an information system graphically. DFDs may also be used for structured design, which visualizes data processing. (2005) Dennis J. et al. Data flow diagrams are created using just four fundamental symbols. Data sources, data flows, data transformations, and data storage are all represented by these symbols. Nodes, which are often enclosed figures like circles, show the locations where data is altered.

Data Flow Diagram Symbols:

-Source or destination of data. Also

called external entity

- Data Flow

-Process

-Data store / storage

*Figure 4.4 data flow diagram symbols*

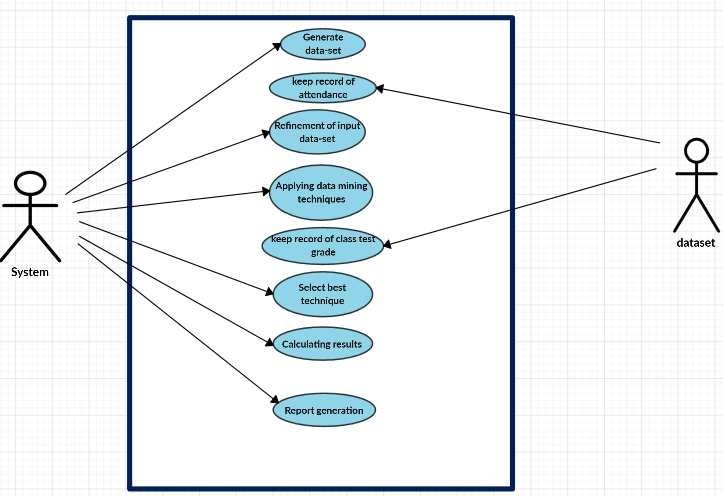
**4.5.1 Context Dataflow Diagram**

**4.6 Use case Design**

**4.6.1 Use case diagram**

A graphical depiction of the high-level system scope is a use case diagram. It contains actors, who are the system’s users, and use cases, which are functional components the system will offer. Use cases depict how a system often interacts with its users (end users and other systems). These interactions show how the system appears to users from an external, or functional, standpoint. A scenario is the name given to each route taken through the use case.

**4.6.2 Use Case Diagram for the Academic performance system**



**4.7 Entity Relationship Diagram for the Academic Performance System**

An abstraction and conceptual representation of data is called an entity-relationship model (ERM). Entity Relationship Modeling is a database modelling technique that creates a certain kind of conceptual system schema. An information system's entities and their connections are graphically represented by an entity-relationship diagram, which is a data modelling approach.

Its components consist of three, they are;

1. The entity is a thing, person, place, or event about which data is being gathered. Consider the information system for a business as an example. Entities would comprise not just customers but also orders and the address of each client.
2. The interaction between the entities constitutes the relationship. The consumer puts an order in the aforementioned example; hence the word "places" refers to the interaction between that specific customer and the order or orders that they place.
3. The link between the things is described mathematically by their cardinality. An entity may not be necessary. Cardinality notation comes in a variety of forms, with the crow's foot notation being one of the most used. One-to-one, also known as 1:1, one-to-many, or 1:M, and many-to-many, also known as M: N, are the three primary cardinal relationships.

**4.7.1 Entity Relationship Diagram**

**4.8 Flowchart Design of the Academic Performance System**

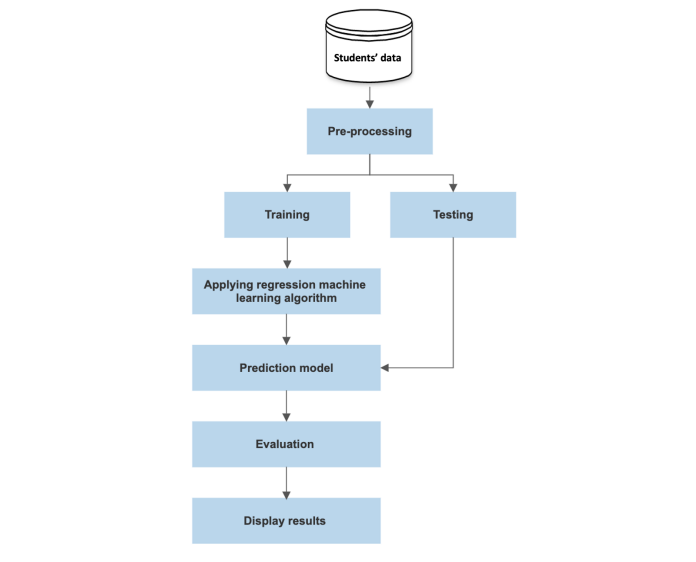
A flowchart is a diagram that shows how a system, computer algorithm, or process works. They are frequently used in many different disciplines to examine, organize, enhance, and convey frequently complicated processes in simple, understandable diagrams. Rectangles, ovals, diamonds, and perhaps many more shapes are used in flowcharts, also known as flow charts, to indicate the kind of step and connecting arrows to indicate flow and sequence. They might be anything from straightforward hand-drawn charts to detailed computer-drawn diagrams showing several processes and pathways. Considering all the different variations, flowcharts are among the most widely used diagrams on the world, being utilized by both technical and non-technical persons in a wide range of industries. The terms Process Flowchart, Process Map, Functional Flowchart, Business Process Mapping, Business Process Modeling and Notation (BPMN), or Process Flow Diagram are sometimes used to refer to flowcharts (PFD). They have a connection with other well-known diagrams like Data Flow Diagrams (DFDs) and UML Activity Diagrams.

**4.8.1 Flowchart Symbols**

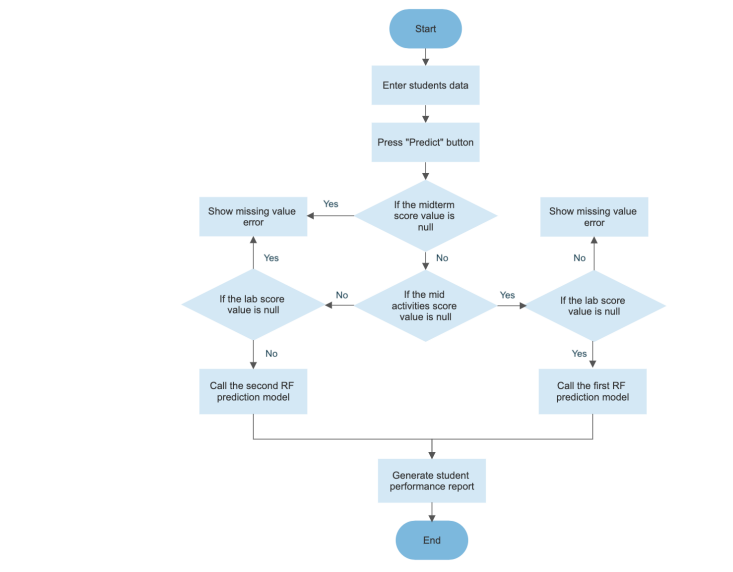
|  |  |  |
| --- | --- | --- |
| **Symbol** | **Name** | **Function** |
|  | Start / end | An oval represents a start or end point |
|  | Arrows | A line is a connector that shows relationships between the representative shapes |
|  | Input / Output | A parallelogram represents input or output |
|  | Process | A rectangle represents a process |
|  | Decision | A diamond indicates a decision |

*Table 4.7 Flowchart Symbols*

**4.8.2 Flowchart Diagram**



*Figure 2. Flowchart model of the Academic Performance System*



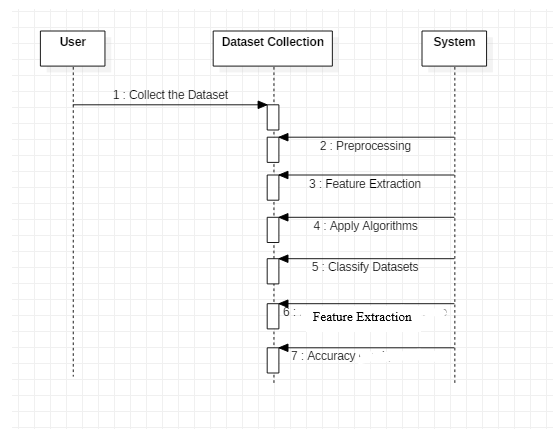
*Figure 2. Flowchart of the prediction System*

**4.8.3 Sequence Diagram**

A sequence diagram is an interaction diagram that demonstrates the order and interactions of several things. It is an implementation of a message sequence diagram.

Object interactions are arranged in temporal sequence in a sequence diagram. It shows the classes and objects involved in the scenario as well as the flow of messages that must be exchanged for the objects to work as intended. In the Logical View of the system being developed, sequence diagrams are often connected to use case realizations. Event diagrams and event scenarios are other names for sequence diagrams.

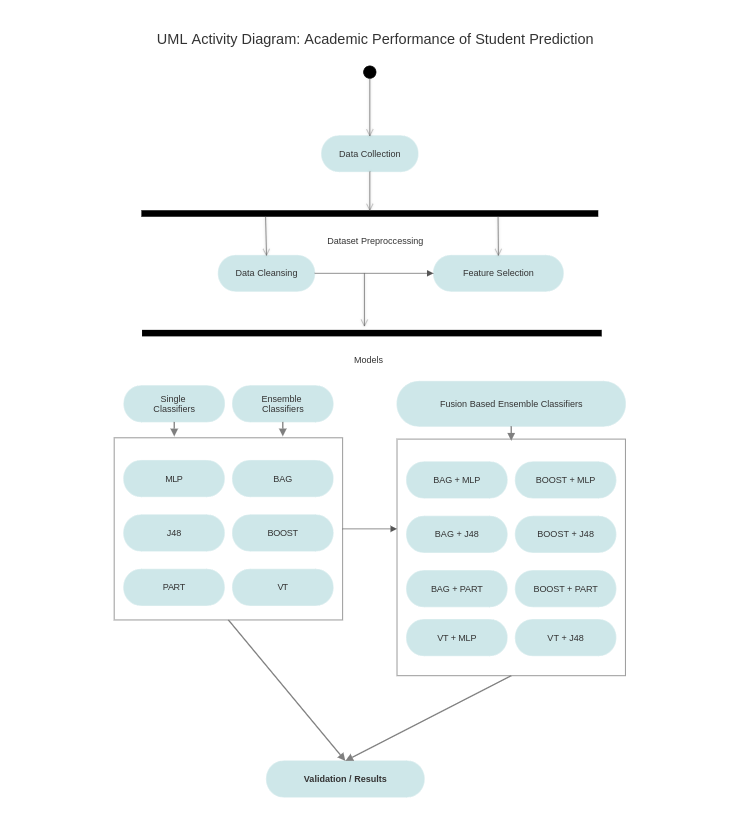
**4.8.4 Sequence Diagram of the Academic Performance System**



**4.9 UML Activity Diagram**

Another crucial behavioral diagram in the UML used to depict the system's dynamic elements is the activity diagram. An activity diagram is simply a more complex flow chart that models the transition between activities.

UML Activity Diagram;



**4.10 System Implementation**

The system implementation approach that was used was the pilot implementation. This is because the Academic Performance System is a new system for most institutes especially for this project focus college (AIT). A company can validate its strategy for full application deployment through a pilot implementation. Running an application pilot can reveal operability problems related to production-like circumstances and give a chance to fix these problems before a complete application roll out. An organization should create a thorough pilot strategy, choose the right participants, organize the pilot environment, and decide how to monitor the pilot in order to properly be ready for a pilot deployment. This might entail either exploring a brand-new idea or method, or applying a tried-and-true method that has been suggested by outsiders but has never been used by the organization.

**4.10.1 System Testing**

The process of testing demonstrates that a program complies with its specification and operates as intended. Testing is used to find mistakes or circumstances where the program's behavior is unacceptable. Testing aims to demonstrate thoroughness, enhance software quality, and offer maintenance assistance. Several guidelines that can work well as testing objectives are stated by Glen Myers [MYE79] in his book "The Art of Software Testing" from 1979:

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

The suggested system was tested using the following methodologies:

* Careful scrutiny of procedural detail is the cornerstone of white-box software testing. By providing test cases that put certain sets of circumstances and/or loops to the test, the program is tested along its logical routes. It is possible to check the "state of the program" at various times to see if the asserted or expected status matches the actual situation. (Pressman, R.S., & Herron, S.R., 1991).
* The focus of black-box testing, also known as behavioral testing, is on the functional specifications of software. The software engineer may determine the input circumstances that will completely exercise all of a program's requirements using this testing methodology. Black box testing attempts to find the errors like;

1. Incorrect functions
2. Interface errors
3. Mistakes while accessing other databases or data structures
4. Mistakes in behavior or performance
5. Initiation and termination mistakes

In Black box testing software is exercised over a full range of inputs and outputs are observed for correctness.

**4.10.2 Software Testing Strategies**

The strategies for testing of the system involve;

1. Unit Testing: Unit testing is the basic level of testing. Unit testing is used to make sure that every software has been thoroughly tested.
2. Integration testing: Integration testing is the next phase. To make sure the software requirements are satisfied, separate program components or programs are combined and tested as a whole system.
3. Acceptance testing: Planning and carrying out various sorts of tests as part of acceptance testing is necessary to show that the developed software system fits the requirements. Finally, after passing all of the testing phases, our product satisfies the criteria.

**4.10.3 System Security**

Security, integrity, privacy, and confidentiality are the four connected problems that make up the system security challenge. They choose the access protocols and file and data structures.

1. When referring to system security, we mean the technical advancements and practices used on the hardware and operating systems to guard against intentional or unintentional damage from a specified danger. Data security, on the other hand, is the safeguarding of data against loss, disclosure, alteration, and erasure.
2. System integrity refers to the programs' proper operation, suitable physical security, and protection from external dangers like wiretapping and eavesdropping. Data integrity, in contrast, ensures that data is not altered or damaged in a way other than what authorized users intended.
3. Confidentiality is a unique designation given to sensitive data in a database to reduce potential privacy breach. Information's requirement for protection is characterized by this quality. The technological method of offering such protection is system security. Contrarily, privacy is mostly a procedural issue related to the way information is utilized.

**4.11 Conclusion**

The creation and execution of a computerized product distribution monitoring system were discussed in this chapter. This system was created using the Agile technique. The UML methodology was used to evaluate the design and functions of the system. The system functionalities were designed using a use case diagram. System databases that store and retrieve data were designed using entity relational modelling (ERM). The breakdown of the various entities and the interaction between the system and the user were done using data flow diagrams. This chapter also covered the system testing technique.

**CHAPTER FIVE**

**FINDINGS, CONCLUSION AND RECOMMENDATIONS**

* 1. **Introduction**

The construction phase of the project began once the thorough design for individual modules was finished. During this phase, integration of the numerous source codes into the overall system was accomplished. for system users to quickly become accustomed to the electronic wallet system. This chapter's main goal is to describe the transition from the system design phase to the operational system and overall, this chapter also includes an overview.

Academic Performance System implements web-based technologies such as;

* PYTHON 3
* VISUAL STUDIO CODE
* WEB SERVER

The system's implementation tools must take the project's requirements and scope into account. The developer's experience must also be taken into account. However, the developer's insufficient experience might have an impact on the creation of a solution.

* 1. **User Interface**

Assuring that users can complete their tasks effectively, efficiently, safely, and with enjoyment is a key objective of interface design. To achieve this, a designer must be aware of the available technology as well as the needs, tasks, and characteristics of the system's potential users (Yeates and Wakefield) (2004, p.318).

The Graphical User Interface (GUI) is used by the information system in its design process. The user points at and clicks on buttons with a pointing device to carry out particular actions. Users are more likely to adopt and routinely utilize the system if they have GUI access.

According to Nielsen (2006), there are ten guidelines for a well-built user interface, which include:

1. System state visibility
2. System compatibility with the outside world
3. User control and freedom
4. Consistency and standards
5. Error avoidance
6. Recognition as opposed to memory recall
7. Flexibility and effectiveness of usage
8. Beautiful and minimalist design
9. Guidance and instructions

**Interface 1**: The student data is entered into the home interface as shown in Figure 5. The required is to enter the student’s ID number to print it in the report, choose the course name, choose the academic level, enter the midterm score, enter the mid activities & lab scores (If available). In addition, it requires demographic data, which is the city where the student lives, the marital status of student (i.e., single or married), has children or not, and has health issues or not. After filling in the data, a button named “Predict” is clicked to perform the prediction model and generate the prediction results

Home interface Add image

**Interface 2**: After getting the prediction results about the student through the home interface, the report interface shown in Figure 6 displays a report in the form of a table containing the student’s information and the prediction results. The prediction results display an expected total score and the expected grade that depends on the total. Once it clicked on the “Print” button, the report page will be prepared for printing.

Report Interface image

* 1. **Programming Languages**

The language that was used to develop the Academic Performance System is a machine language called PYTHON. PYTHON is a widely used, flexible, and all-purpose programming language. It works well as a first language since it is clear and simple to understand, and it is a useful language to have in any programmer's toolkit because it can be used for everything from creating websites to creating software for scientific purposes. It is easy-to-understand grammar makes it the ideal language for someone who is just starting to learn computer programming.

* + 1. **Why Using Python for the Academic Prediction system**

In order to show data (information) visually and often to draw conclusions and make decisions based on the information, data analysis and visualization is utilized. Python is the best language for making graphs like bar charts and line charts, which may be used to express information in an image. The information or data offered is frequently numerical and may be based on a certain time frame. Charts can be as basic or as sophisticated as necessary and there may be several aspects at play.

Benefits;

1. A great library ecosystem
2. A low entry barrier
3. Flexibility
4. Platform independence
5. Readability
6. Good visualization options
   1. **Implementation**

A strategy, a technique, or any other design for carrying out anything is implemented when it is carried out, executed, or put into effect. Implementation is therefore the activity that must come after any initial planning in order for anything to truly occur. Implementation in the context of information technology includes all the steps necessary to get new hardware or software functioning correctly in its environment, including installation, setup, running, testing, and making the necessary adjustments. Sometimes, the terms "deployment" and "deployment" are interchangeable.

* 1. **Conclusion of the Study**

We are in an era of technological advancement and rapid progress in almost all aspects of living. It is therefore of great importance if our schools are designed to assist student in their educational performances as well as lecturers to better the future of education.

This research aims at helping schools (tertiary institutions) predict student performance by using data they already have thereby giving them the power to make informed decisions that will better enhance student performance and increase the retention rate on students. The main objective of this project is to design and implement an academic performance prediction system for schools in such a way students can be able to track their performances in their academic and also for lecturers to be able to track students’ performance so as to help increase teachings.

In achieving the main objective mentioned, this project will specifically concentrate on the following;

* The system shall maintain better user relationship.
  1. **Maintenance**

Making updates to a system after it has been deployed is known as system maintenance. The system needs to be modified in a variety of ways, including small adjustments to fix code problems, larger adjustments to fix design flaws, big additions to fix specification errors, and adding new features that couldn't be included in a timely manner. Three different kinds of system maintenance exist:

1. Repairing fault: is expensive since it may need a significant system redesign in the case of a requirements mistake. Coding mistakes may be easily fixed in this situation.
2. Environmental Adaptation: This sort of maintenance is necessary when the hardware, the operating system platform, or other supporting software, among other environmental factors, change. After then, the application system is changed to accommodate these environmental changes.
3. Functionality Addition: This sort of maintenance is required when organizational or business changes affect the system needs. When compared to other forms of maintenance, the size of the modifications that must be made to the program is frequently substantially larger.
   1. **Implications**

The main implication of this study for this project is that the system enhances effectiveness and efficiency in support activities for the semester. The system would be of immense importance for schools and any other learning institution. Also, the system opens up opportunities for schools worldwide and also for students who finds it difficult in learning. It will help the student know the area he /she needs to improve in their respective academic career.

* 1. **Future Works and Recommendation**

The researcher advises moving on with the system's implementation while evaluating it for the addition of new modules or components that have been left out due to time and financial constraints.

Schools using this system should also;

1. As new issues occur, increase the number of solutions.
2. Administrators who will use or administer the system should have proper training.
3. Regularly backup database in case of a calamity or system breakdown.
4. Make sure that hardware components are constantly functioning properly. If necessary, replacement should be made.

The developer should be engaged for a system or program upgrade as the management's needs grow.

* 1. **Conclusion**

In the chapter one, the problem statement of current system was explained and also the project objectives which are to design an Academic Performance Prediction System. In chapter two the researcher conducted a literature review of existing systems. In Chapter three, the researcher underscored the various methodologies that will be used to implement the system. Chapter four, the researcher used the methodology stated in chapter three to build the system thus, System Analysis, System Design, System implementation and testing the developed system.

**REFERENCES**

1. Bithari, T.B.; Thapa, S.; Hari, K. (**2020**, *2*, 89–98), *Predicting Academic Performance of Engineering Students Using Ensemble Method*. *Tech. J*. [**Google Scholar**] [**CrossRef**]
2. Adejo, O.W.; Connolly, T. (**2018**, *10*, 61–75*), Predicting student academic performance using multi-model heterogeneous ensemble approach*. *J. Appl. Res. High. Educ*. [**Google Scholar**] [**CrossRef**]
3. Jakkula V. (2006), *Tutorial on Support Vector Machine (svm)*; School of EECS, Washington State University: Washington, DC, USA; Volume 37. [**Google Scholar**]
4. Jia, Y.S.; Jia, C.Y.; Qi, H.W. (August 2005), *A new nu-support vector machine for training sets with duplicate samples*. In Proceedings of the 2005 International Conference on Machine Learning and Cybernetics, Guangzhou, China, 18–21; Volume 7, pp. 4370–4373. [**Google Scholar**]
5. Breiman, L. Random forests. (**2001**), *Mach. Learn*, *45*, 5–32. [**Google Scholar**] [**CrossRef**]
6. Akar, Ö.; Güngör, O. (**2012*),*** *Classification of multispectral images using Random Forest algorithm*. *J.Geod. Geoinf*, *1*, 105–112. [**Google Scholar**] [**CrossRef**]
7. Han, J.; Pei, J.; Kamber, M. (2011), *Data Mining: Concepts and Techniques*, 3rd ed.; Morgan Kaufmann: Burlington, MA, USA. [**Google Scholar**]
8. Russell, S.J.; Norvig, P. (2016.), *Artificial Intelligence: A Modern Approach*, 3rd ed.; Pearson: London, UK [**Google Scholar**]
9. Seber, G.A.; Lee, A.J. (2012), *Linear Regression Analysis*; John Wiley & Sons: Hoboken, NJ, USA, Volume 329. [**Google Scholar**]
10. Kavitha, S.; Varuna, S. (19 November 2016); Ramya, R. *A comparative analysis on linear regression and support vector regression*. In Proceedings of the 2016 Online International Conference on Green Engineering and Technologies (IC-GET), Coimbatore, India; pp. 1–5. [**Google Scholar**
11. Wojtas, M.; Chen, K. (**2020),** *Feature Importance Ranking for Deep Learning*. *arXiv*, arXiv:2010.08973. [**Google Scholar**]
12. Ly-Huong T. Pham, Teja Desai-Naik, Laurie Hammond, & Wael Abdel Jabbar (Apr 8, 2021), *Implementation Methodologies*
13. (Hevner et al, m. e. (1992, 1995, 2004). *design science research. design science research*.
14. Adams et al, n. e. (1991,1992,2004). *design science methodology. theoretical frameworks to justify design research studies.*
15. alechina, N. (2014). *Entity-relationship*. Entity-relationship model.
16. Archer. (1984). *data collection*. data collection.
17. beal, V. (1982). application software. *application software*
18. Cooper. (1990, and 2000). design science methodologies. *design or development methodologies.*
19. Eekels et al, h. e. (1991, 1992, 2001). design science methodologies. *design or development methodologies.*
20. Simplilearn (2022), *Feasibility Study and Its Importance in Project Management*
21. Costa, E.B.; Fonseca, B.; Santana, M.A.; de Araújo, F.F.; Rego, J. *Evaluating the effectiveness of educational data mining techniques for early prediction of students’ academic failure in introductory programming courses*. Comput. Hum. Behav. 2017, 73, 247–256.
22. Liao, S.N.; Zingaro, D.; Thai, K.; Alvarado, C.; Griswold, W.G.; Porter, L. *A robust machine learning technique to predict low-performing students.* ACM Trans. Comput. Educ. (TOCE) 2019, 19, 1–19.
23. Hu, Y.H.; Lo, C.L.; Shih, S.P. *Developing early warning systems to predict students’ online learning performance*. Comput. Hum. Behav. 2014, 36, 469–478.
24. Jayaprakash, S.; Krishnan, S.; Jaiganesh, V. *Predicting Students Academic Performance using an Improved Random Forest Classifier*. In Proceedings of the 2020 International Conference on Emerging Smart Computing and Informatics (ESCI), Pune, India, 12–14 March 2020; pp. 238–243.
25. Francis, B.K.; Babu, S.S. *Predicting academic performance of students using a hybrid data mining approach*. J. Med Syst. 2019, 43, 162.
26. Nandeshwar, T. Menzies and A. Nelson (2011), *Learning patterns of university student retention.*
27. [Alexander Ryabtsev](https://djangostars.com/blog/author/alexander-ryabtsev/) (2019), *Why Python is Good for Artificial Intelligence and Machine Learning*. March 11.