**TELONE CENTRE FOR LEARNING**

****

**TECHNICAL DEPARTMENT**

**NAME........................................JOSCELYNE**

**SURNAME................................MATAMBO**

**STUDENT ID............................5105/T1857582T**

**PROGRAMME.........................DIPLOMA IN TELECOMMUNICATION**

**ASSESMENT.........................PROJECT**

**DUE DATE.................................02/08/21**

**DECLARATION**

I Joscelyne Matambo , hereby declare that I am the sole author of this dissertation. I authorize Telone Centre For Learning to lend this thesis to other institutions or individuals for the purpose of scholarly research.

**Table of Contents**

**Introduction ………………………………….……………………….…1.1**

**Background Information………………………………………………1.2**

**Problem definition …………………………….………………………1.3**

**Objectives………………………………………….……………………1.4**

**Functionality………………………………………….…………………1.5**

**Justification…………………………………………….………………1.6**

**Conclusion……………………………………………………..………1.7**

**CHAPTER 1**

**INTRODUCTION**

**TITLE: Online driving school management system**

“Driving School Management System” is a system that allows driving schools to automate all processes and task done for their daily transactions. The system will let driving schools electronically schedule classes and keep records of the students who avail for driving lessons.

This study proposed a Driving School Management System to automate the processes of daily transactions in driving schools. The system will let the school manage the information of the students who enrolls for driving lesson, the schedule of the students up to the payment transactions of the students.  The system will allow electronic recording and keeping of records of the students enrolled in a specific driving school. The proposed system is efficient and reliable to use by driving schools.

## **Background Information**

Melody Driving School is a driving school located in Harare. Which was founded in 2015.

The Driving school has following functions:

**Student Registration:**

It is responsible for registering the students for practical and oral driving lessons. Currently for a student to register for lessons s/he has to physically visit the Company offices in Hatfield and register for lessons.

**Instructors:**

It is responsible for scheduling driving lessons for practical and oral lessons. Since the company started operating, they have been using paper-based system for the scheduling of student and lecturer lessons which has been causing some problems.

**Accounts:**

Responsible for the collection and monitoring of tuition fees payments. For a student to make a pay for his/her lessons s/he has to visit the company offices physically.

**Vehicle Fleet Control:**

Responsible for managing vehicles at the school for the students for taking their exams or for practical lessons.

All these processes are currently done manually and they are so many problems which are being faced during execution of work.

## **Problem Definition**

The company has faced a road block using their current paper-based system. Keeping students’ records organized and accessible to both student and employees have become a hassle, error prone and cumbersome. Currently, paper-based system has caused the students not to have access to their own training records and, the unsecure way of storing files without backups has just been causing a big problem to the operation of the organization also, the manual scheduling of lessons has been causing the lessons to being scheduled incorrectly or having classrooms overbooked.

## **Objectives**

**General Objective –** the main objective of the project is to design and develop a Driving School Management System that will serve as platform to electronically do processes and transactions for the Melody driving school.

1. To provide online registration of students so that they will be a digital repository containing all registered students
2. To develop online scheduling of the lessons for students
3. To provide online payment platform

## **Functionality**

Driving School Management System automates all processes and task done by the driving school.

* Student registration
* Lesson Scheduling
* Payments for the lessons

## **Hypothesis**

A number of solutions can be employed to the current system as follows:

H 0: Developing a Driving Management system

H 1: Improvement of the manual system

H 1: Outsourcing from software companies

Of the above alternatives, Driving Management system is the best more modern alternate.

If partially automated, the system will be flexibly able to retrieve, update information and status quickly, when required without taking much time. Students will be able to view Lessons, schedule their lessons and make online payments.

## **Justification**

**Melody Driving School** will have a system that will assist their daily operation. The system will be efficient in handling all processes electronically and eliminate the manual approach that may encounter a lot of difficulties.

**Driving Instructors.** By having the system, they can easily track students who are scheduled for a driving session. They can easily look for updates and communicate with their students.

**Students**. Students who want to enrol and avail driving lessons will be benefited by the success of the project. It will be easier for them to register, schedule and pay for their driving lesson. Transacting for driving lessons will be convenient for them.

## **Conclusion**

The researchers of the study concluded that the developed Driving School Management System will provide an effective way of managing the different records in a driving school. The system will eliminate all the problems encountered in the manual method which will help the school better improve their daily operations. By implementing the system, the school can render an improved service and satisfying experience to their students and other stakeholders.

**CHAPTER 2: PLANNING PHASE**

# **2.0 INTRODUCTION**

Having undergone project proposal, one looks at the planning phase and certain critical factors have to be taken into consideration. This include the reason why the project is undertaken based on the objectives of the project and problems of the current system at hand which were identified and mentioned in the proposal. Also, it can be established if project can be successfully be undertaken by the resources available and this can be justified by undertaking a feasibility study.

## **2.1 Why Build the System**

With the road block being faced by using paper-based system in keeping students’ records organized and accessible to both student and employees have become a hassle, error prone and cumbersome. Currently, paper-based system has caused the students not to have access to their own training records and, the unsecure way of storing files without backups has just been causing a big problem to the operation of the organization also, the manual scheduling of lessons has been causing the lessons to being scheduled incorrectly or having classrooms overbooked with all these problems there is a need of developing a new system.

## **2.2 BUSINESS VALUE**

Implementing the Online Driving School Management System has many positive outcomes for both the business. From the business point of view, a new system is definitely needed by the end of this year, the business success and its employees as well as customers satisfaction depends on this new system. Not all driving schools have been fortunate enough to have the large amount of success that Melody Driving School is currently having. Because of this, not all driving schools need a system like the one being requested. Not only would Melody School of Driving be one of the few schools in the city with such advance management system but they will also set the standards for future driving schools and be one of the few.

## **2.3 INFORMATION GATHERING METHODOLOGIES**

It is necessary for information gathering so as to have enough understanding of the current system which leads to requirements determination. And to gather useful information, the following fact-finding techniques were employed:

### **2.3.1 Surveys**

[Survey](https://www.qualtrics.com/experience-management/research/survey-basics/) is another method of gathering information for research purposes. Information is gathered through questionnaire, mostly based on individual or group experiences regarding a particular phenomenon.

There are several ways by which this information can be collected. Most notable ways are: web-based questionnaire and paper-based questionnaire (printed form). The results of this method of data collection are generally easy to analyze.

### **2.3.2 Interviews**

[Interview](https://research-methodology.net/research-methods/qualitative-research/interviews/) is a qualitative method of data collection whose results are based on intensive engagement with respondents about a particular study. Usually, interviews are used in order to collect in-depth responses from the professionals being interviewed.

Interview can be structured (formal), semi-structured or unstructured (informal). In essence, an interview method of data collection can be conducted through face-to-face meeting with the interviewee(s) or through telephone.

### **2.3.3 Observations**

Observation method of information gathering is used by monitoring participants in a specific situation or environment at a given time and day. Basically, researchers observe the behavior of the surrounding environments or people that are being studied. This type of study can be [controlled, natural or participant.](https://www.simplypsychology.org/observation.html)

Controlled observation is when the researcher uses a standardized procedure of observing participants or the environment. Natural observation is when participants are being observed in their natural conditions. Participant observation is where the researcher becomes part of the group being studied.

## **2.4 TANGIBLE BENEFITS**

* Decreased response time
* Elimination of job steps
* Reduced stationery expenses
* Fast processing and printing of students billing records and reports
* Reduced time of enquiring
* Reduced work overload

## **2.5 INTANGIBLE BENEFITS**

* Improved student goodwill
* Improved employee morals
* Improved job satisfaction
* Better customer services
* Improved quality of information
* Better decision making by management

### **2.6 FEASIBILITY STUDY**

Feasibility study will be there to verify whether it is feasible to carry out that project. The feasibility can be characterized into technical, economic and operational. This feasibility analysis enables the analyst to provide justification on whether the proposed Driving Management System desired objectives can be achieved within the prevailing economic, financial, organizational and technological constraints and a descriptive comparison between the benefits and the costs.

### **2.6.1 Technical Feasibility**

The proposed system will be characterized by new technology and thus it is essential to validate its technical feasibility. The extent of the successful development and implementation of the system also depend on the availability of technical expertise. An analysis of the current infrastructure at Melody Driving School offices shows that it is sufficient enough for the system to be said to be technically feasible to develop and implement.

The following are some of the reasons that verify the technical feasibility of the system:

- There is a fully fledged network system that has at least a server operating well below their actual capacity.

- The developer who is going to be responsible for the creation and deployment of the system has the necessary technical expertise to carry out the project.

- Most of the users are computer literate and have practical experience working with computers and should not have minimum problems with running the system.

**2.6.2 Economical feasibility**

Economic feasibility will be checking if the expected benefits outweigh the costs to be incurred by the system? Does the team have sufficient resources to finance the proposed system?

The internet allows for the download of open-source developer tools such as the ones which are being used to create the system. This means little or no money is required for the purchase of software. In as far as hardware is concerned, it has been mentioned already that the organisation has the necessary infrastructure in place but it is just that it is not fully utilizing the resources. A survey was also conducted to examine the extent to which the benefits outweigh the costs and the developer was able to come up with the following cost benefit analysis:

### **2.6.4 Operational feasibility**

This defines acceptability of the system by users as a solution no their current problems. This also defines the friendliness of the system to users, thus ease of use. In exploring operational feasibility, the PIECES (performance, information, economy, control, efficiency, services) framework is used.

**Performance** - the new system will provide adequate response time

**Information-**the new system will provide accurate, useful and timely information to users and management.

**Economy** -the current manual system provides cost ineffective information to the business because there are high time costs incurred in retrieving information. The new system will provide timely information and help reduce telecommunication costs.

**Control-** the current manual system is prone to fraud and has no guaranteed security. The new system offers effective control to protect` against fraud and unauthorized access. It reduces the number of errors made during data entry.

**Efficiency-** the new system will take lesser time to process information and to produce more accurate results, thereby making it more efficient than the current one.

**Services-** the new system will provide more reliable services, which are flexible.

## **2.7 RISK ANALYSIS.**

This phase identifies, evaluates, and try to come up with solutions on things that might go wrong in the project before it becomes a threat to the successful completion of the project or implementation of the Proposed System. The below Risk Analysis Log is also supported by the technical, economical and any other risks that are associated with the implementation of the new proposed system. The logs analyse and recommend so as reducing the chances of jeopardizing the new system and any other jobs associated with the implementation.

## **2.8 WORK PLAN**

This also shows the time taken to complete the project. The time is expressed in terms of weeks. Each phase has the start time and the completion date.

The Systems Development Life Cycle shall be used to model the activities that are going to be followed in the system development of the project.

* This model has been chosen over other models, since it is a well-documented and widely used approach.
* This is a linear approach involving the stages that are clearly defined at each stage.
* This approach clearly distinguishes the task break down of all tasks to be done, example being feasibility, analysis, design, coding, testing and maintenance.
* Each phase is given a time allocation from the beginning to the end of the project.
* Targets are easy to set with this approach, and one knows exactly what needs to be done from which period of time.
* There is an elaborately written documentation at each phase, this helps in developing a well-documentedsystem.

**Gantt Chart. Task description**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Project proposal |  |  | | | | |
| Feasibility report |  |  |  | | | |
| System analysis |  | |  |  | | |
| System design |  | | |  |  | |
| Implementation |  | | | |  |  |
| Evaluation and  Maintenance |  | | | | |  |
| Documentation |  | | | | | |

## 2.9 CONCLUSION

The project is quite feasible according to the feasibility study as well as the time frame given. After this stage the next phase of systems development that follows is Analysis of the current system where information will be gathered to find how it is working.

# **CHAPTER 3: ANALYSIS PHASE**

## **3.0 Introduction**

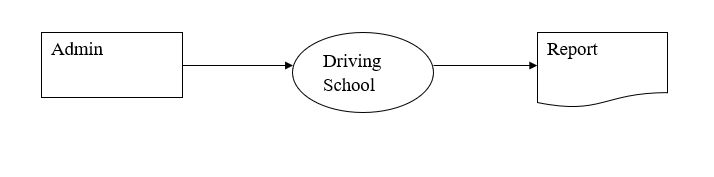
The analysis phase seeks to answer the following question, “What does the system do?” This mainly involves the requirements that the system tries to meet. At the end of the day the deliverable of the analysis phase is the requirements documentation. The main purpose of the analysis phase is therefore to bring all those pieces together to form a logical database model or system containing all entities and their attributes, domain relationships, together with a complete functioning model with its hierarchy, and domain constraints (on attribute), business rules (constraints) and events that trigger functions. The scope here is therefore, “What is to be made, and not how?” It is all about the business and not the system.

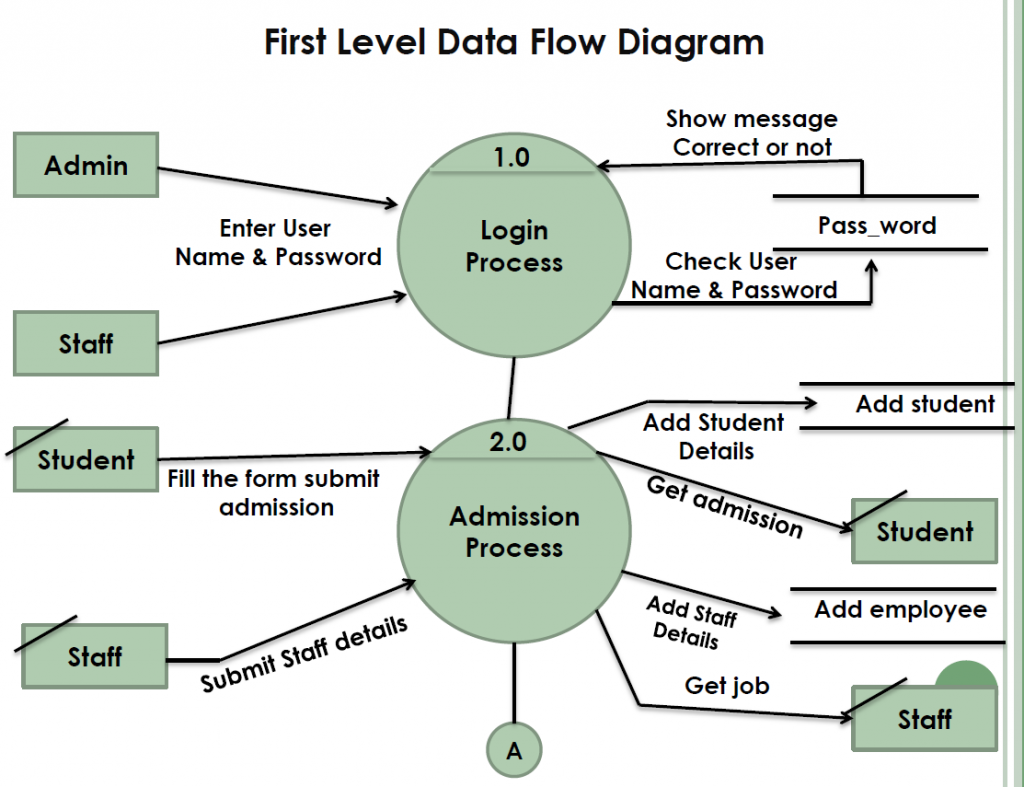
## **3.1 ANALYSIS OF EXISTING SYSTEM**

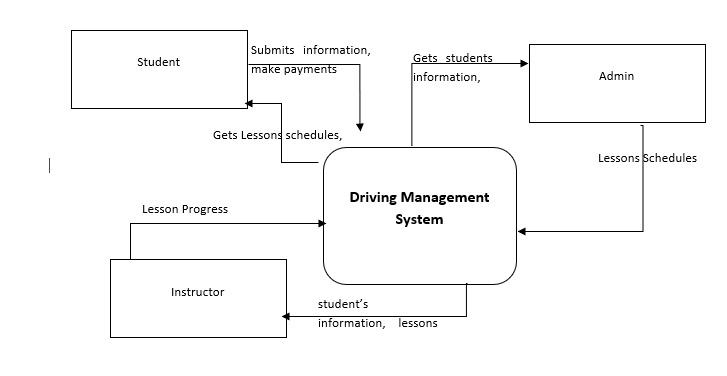
* The Driving School Management System works are done manually.
* The admin has to maintain the driving student details properly. Otherwise, they cannot allot timings for each learner.
* And the Driving school have to preserve whether the learner need license. If they need license the admin should apply for license.
* These processes are also made by manually. So, the manual work is very difficult to run the driving classes.

## **3.2 DATA ANALYSIS-DFD**

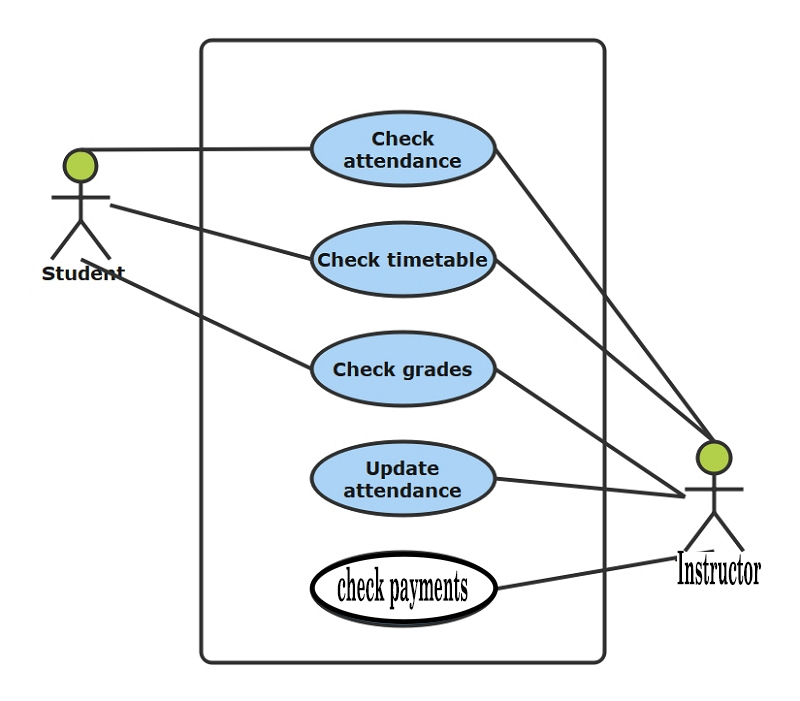
**Admin level 0**







## **case diagrams**



## **3.3 Overview of Proposed System:**

* **Scheduling:** Lesson scheduling is an important part of every driving school. Online Driving Management System allows to create lessons schedules and monitor them. This feature helps prevents double booking of time slots and has a beautiful calendar to display all lessons. Also, an email is sent to students when they are booked in for a class.
* **Student Management**: Keep all your student records in a way that you can access them from anywhere. You can search through them easily and be able to check all the details of a specific student.
* **Online Learning Portal**: Create an online curriculum and provide learning resources for your students to learn online from anywhere with any device.
* **Notes and attachments:** You can create and upload notes on a profile of a students, staff or instructor.
* **Instructors:** Instructors of courses will also have accounts where they can login and check their schedule.
* **Branches:** Most driving schools have branches and that is covered on the system. Create as many branches of your driving school as you want and manage them easily.
* **Stats dashboard:** online driving school management system comes with a stunning dashboard that shows you the important statistics of your school.
* **Invoice Management:** When a student account is created and they sign up for a course. An invoice is created and sent to them. You can keep track of all your invoices and see pending balances. You can download, edit and add payments to invoices as well
* **Payments:** Keep a record of all the payments you receive for future reference as well as update the invoice automatically.
* **Reminders:** Send reminder of late payments and also reminders of upcoming classes to students
* **Fleets:** Keep a record of all your fleet for easy assignment as well as check it’s schedule.
* **Notifications** the system saves and shows notifications of important activities of your driving school.
* **Communications:** This feature allows the admin to send Email to their students, staff, instructors or all users on the system by a click of a button. The history of the messages is kept for future reference.
* **Courses:** here you keep a list of your courses with its details like price, number of classes, instructors etc. This list is used when signing up new students to the driving school.
* **Staff:** The staff module allows driving schools to create accounts for their employees where they can manage daily operations of the school.

## **3.3 WEAKNESSES OF THE CURRENT SYSTEM**

The current driving school management system has the following drawbacks

* The current system is time consuming since it is manual in all its aspects.
* Evaluations and analysis of all students the is difficult and almost impossible because there is no central data repository.
* High stationery cost.
* The system promotes centralization of data to local stations.
* Facilitates corruptive practices due to the intensity of loopholes.

## **3.4 STRENGTHS OF THE CURRENT SYSTEM**

v Driving School Management System contains numerous features that can help driving schools run their business more efficiently.

v One great advantage this software offers is its scalability and flexibility for data management. Using this software admin will be able to manage the most important aspects of business using single software.

v As such, The Driving School Management system is to be developed to replace such paper-based system in order to provide a better controlled and efficient environment which will meet the needs of the day’s services.

## **3.5 EVALUATION OF ALTERNATIVES**

There are options which could also have been considered as an alternative to the development of this system, such alternatives are to be revised now in this section:

* Outsourcing
* Improvement
* Development

## **3.5.1 outsourcing**

Outsourcing is the act of one company contracting with another company to provide services that might otherwise be performed by in-house employees. Often the tasks that are outsourced could be performed by the company itself, but in many cases, there are financial advantages that come from outsourcing. Typically, the function being outsourced is considered non-core to the business.

Drawbacks on outsourcing

* Outsourcing may increase the risk of information leakage, reduce confidentiality, as well as introduce additional privacy and security concerns
* Any sensitive information is more vulnerable, and a company may become very dependent upon it’s outsource providers, which could lead to problems should the outsource provider back out on their contract suddenly.
* Acute when the work is being done in a different country (offshored), since that involves language, cultural and time zone differences.
* Outsourcing IT reduces or completely eradicates direct communication between organization and clients. Limited communication impedes the relationship building process, which may lead to the overall dissatisfaction of the organization and client.
* Organizations that outsource IT services run a risk of receiving poor quality work.
* High charging rates.
* There is also the danger of not being able to control some aspects of the company, as outsourcing may lead to delayed communications and project implementation. Project implementation timelines may suffer as a result

## **3.5.2 improving the current system**

Act of enhancing or making better in terms of quality, value or usefulness. This can be by making ideas, objects or processes more desirable by adding or removing components. The term can be also be applied to people as well, via methods such as performance reviews which are meant to try and improve an employee in some manner.

Drawbacks of improving the current system

* Possibility that the upgrade will worsen the product. Upgrades can also worsen a product subjectively.
* Upgrades of hardware involve a risk that new hardware will not be compatible with other pieces of hardware in a system.
* The main disadvantages are that you might get some new bugs, some of which could even stop you from using older data files.
* More expensive than developing a new system.
* Weaknesses of the old system will resurface.

## **3.5 .3 Development**

This is the development of a system from a company’s internal resources and pool of IT resources. This may involve hiring of experience technical team and pooling of funds to finance the development of the system.

Merits of development

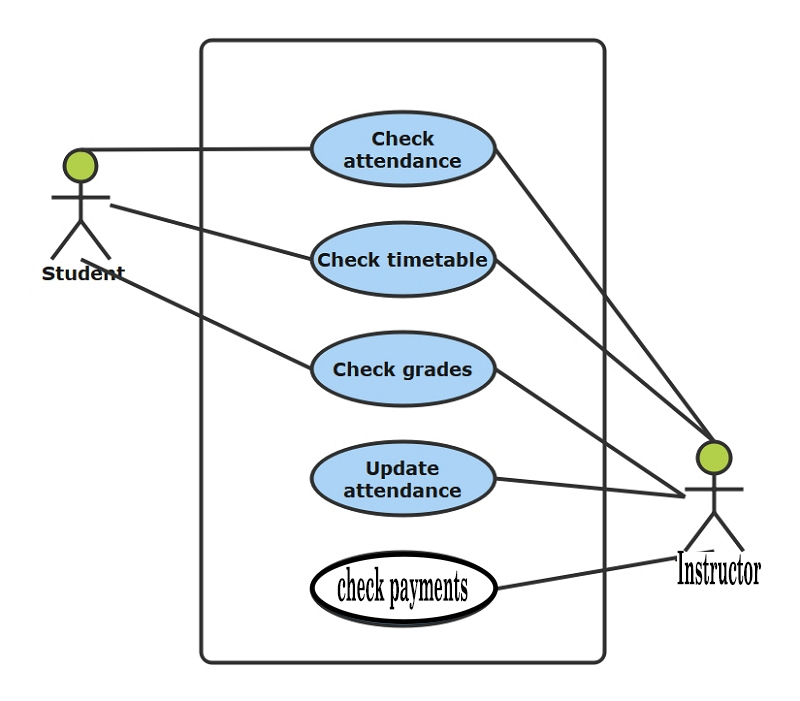
* Legal costs are reduced significantly like licenses and training costs incurred if the system outsourced.
* Implementation will be easier since the personnel responsible for the system development will be readily available.
* It minimizes changes in the business procedures and policies.
* User satisfaction is enhanced.
* Quality Control
* When development takes place in-house, the hands-on coding and testing process translates into the internal team’s system familiarity and results in lower long-term maintenance expenses and more professional application support.
* In-house Design Gives You Control
* Increased Flexibility
* Greater and better functionality since the analyst is more acquainted with the business needs and environment.

## **3.6 REQUIREMENTS ANALYSIS-Use case diagrams**

The purpose of Requirements Analysis is to obtain a thorough and detailed understanding of the business need and to break it down into discrete requirements, which are then clearly defined and reviewed. Requirement analysis can be categorized into two phases namely:

* Functional requirements
* Non-functional requirements

Requirements that define those features of the system that will specifically satisfy a user’s need, or with which the users will directly interact with.



Use cases are used during the analysis phase of a project to identify and partition system functionality. They separate the system into actors, use cases and associations. Associations between actors and use cases are indicated in use case diagrams by solid lines.

An association exists whenever an actor is involved with an interaction described by a use case. Associations are modeled as lines connecting use cases and actors to one another, with an optional arrowhead on one end of the line. The arrowhead is often used to indicating the direction of the initial invocation of the relationship or to indicate the primary actor within the use case. Actors represent roles that are played by users of the system. Those users can be humans, other computers, pieces of hardware, or even other software systems. Use cases describe the behavior of the system when one of these actors sends one particular stimulus. This behavior is described textually. It describes the nature of the stimulus that triggers the use case; the inputs from and outputs to other actors, and the behaviors that convert the inputs to the outputs.

## **3.6.2 Non-functional requirements**

The plan for implementing non-functional requirements is detailed in the architecture. In general, functional requirements define what a system is supposed to do whereas non-functional requirements define how a system is supposed to be. Non-functional requirements are often called qualities of a system. Other terms for non-functional requirements are "constraints", "quality attributes", "quality goals", "quality of service requirements" and "non-behavioral requirements".

Non-functional requirements can be divided into two main categories:

1. Execution qualities, such as security and usability, which are observable at run time.

2. Evolution qualities, such as testability, maintainability, extensibility and scalability, which are embodied in the static structure of the software system.

Security

* Login requirements - access levels.
* Password requirements - length, special characters, expiry, recycling policies
* Inactivity timeouts – durations, actions

Audit:

* Audited elements – what business elements will be audited?
* Audited fields – which data fields will be audited?
* Audit files characteristics - before image, after image, user and time stamp.

Performance:

* Response times - application loading, screen open and refresh times.
* Processing times – functions, calculations, imports, exports.
* Query and Reporting times – initial loads and subsequent loads.

Capacity:

* Throughput – how many transactions per hour does the system need to be able to handle?
* Storage – how much data does the system need to be able to store?
* Year-on-year growth requirements

Availability:

* Hours of operation – when is it available? Considering weekends, holidays, and maintenance times.
* Locations of operation – where should it be available from, what are the connection requirements?

Reliability:

* Mean Time Between Failures – What is the acceptable threshold for down-time? e.g. one a year, 4,000 hours
* Mean Time To Recovery – if broken, how much time is available to get the system back up again?

Integrity:

* Fault trapping (I/O) – how to handle electronic interface failures.
* Bad data trapping - data imports, flag-and-continue or stop the import policies.
* Data integrity – referential integrity in database tables and interfaces
* Image compression and decompression standards

Recovery:

* Recovery process – how do recoveries work, what is the process?
* Recovery time scales – how quickly should a recovery take to perform?
* Backup frequencies – how often is the transaction data, set-up data, and system (code) backed-up?
* Backup generations - what are the requirements for restoring to previous instance(s)?

## **3.7 Conclusion**

This marks the end of the analysis phase of the old driving school management system and the beginning of the designing of the new system. Vital data has been collected using various data collecting methodologies and that data played a pivotal role in the analysis of the old system using various diagrammatic and logical methods. As a result, the loopholes of the old system were exposed at the same time unveiling the inevitable need for the new system. The evaluation of alternatives has also revealed the need of the in-house development of the new system against the popular and attractive options of outsourcing and upgrading the existing system (improvement) hence endorsing the proposed plan. The next stage is the design phase whereby the conceptual appearance of the new system is going to be exhibited.

# **CHAPTER 4 : DESIGN PHASE**

## **4.0 INTRODUCTION**

The starting point is the requirements document delivered by the analysis phase and mapping those requirements into architecture. The requirements identified in the Analysis Phase will be transformed into a system design document that accurately describes the design of the new system and that will be used as an input to system development in the next phase. That architecture defines the components, their interfaces and behaviors. Therefore, this chapter describes a plan to implement the requirements, it represents the ``how'' phase. Details on computer programming languages and environments, machines, packages, application architecture, distributed architecture layering, memory size, platform, algorithms, data structures, global type definitions, interfaces, and many other engineering details are to be established.

## **4.1 SYSTEM DESIGN**

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development.

Kenneth C Laudon: Management Information Systems, “System design refers to detailing how a system will meet the information requirements as determined by the system analysis. The design of an information system is the overall plan or model for that system. Each design represents a unique blend of technical and organizational components.”

The table below illustrates major design specifications and their elements as described in the previous chapter.

|  |  |  |  |
| --- | --- | --- | --- |
| ***SPECIFICATIONS*** | |  |  | | --- | --- | |  | ***ELEMENTS*** | |
| OUTPUTS:  INPUTS:  USER INTERFACE:  DATABASE DESIGN:  PROCESSING:  CONTROLS:  SECURITY:  TRAINING:  CONVERSION: | * Medium * Content * Timing |
| * Origins * Flow * Data entry |
| * Simplicity * Efficiency * Logic * Feedback * Errors |
| * Logical data model * Volume and speed requirements * File organization and design * Record specification |
| * Computations * Program modules * Required reports * Timing of outputs |
| * Input controls(character, limit, reasonableness) * Processing controls(consistency, record counts) * Output controls(totals, samples of output) * Procedural controls(passwords, special forms) |
| * Access controls * Catastrophe plans * Audit trails |
| * Select training techniques * Develop training modules * Identify training facilities |
| * Transfer files * Initiate new procedures * Select testing method * Cut over to new system |

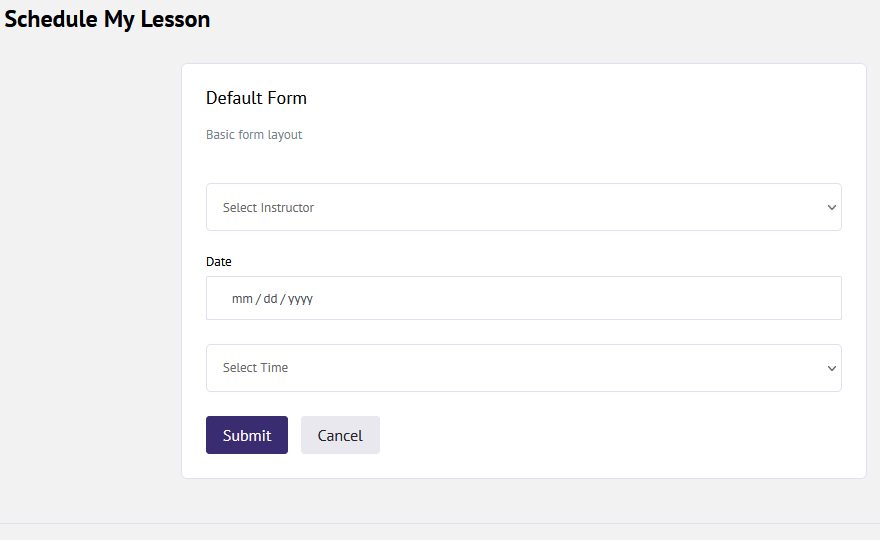
## **Table 4.1.1 System design elements and specifications**

## **4.2 SYSTEM INPUTS**

Input design is the process of connecting the user-originated inputs into a computer to used format. The goal of the input design is to make the data entry logical & free from errors. Errors in the input database controlled by input design.

The goal of designing input data is to make data entry as easy, logical and error free from errors as possible. In entering data, operators need to know the following:

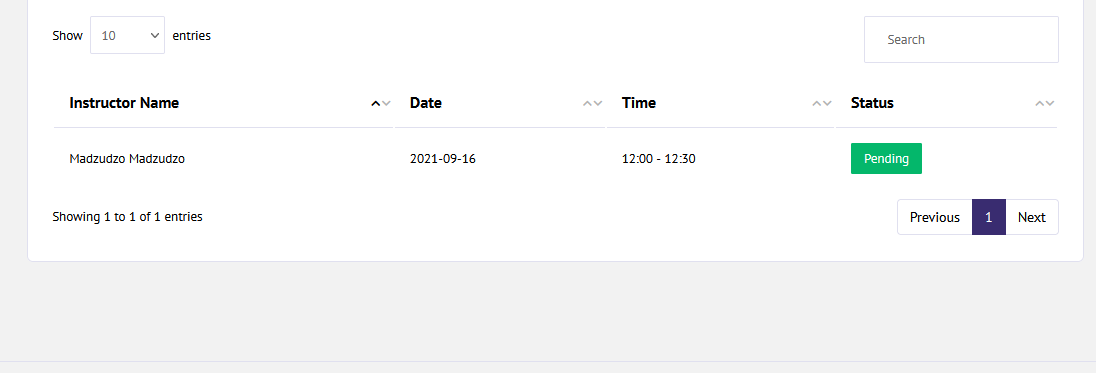
Admin Module, Student Module, Instructor Module,

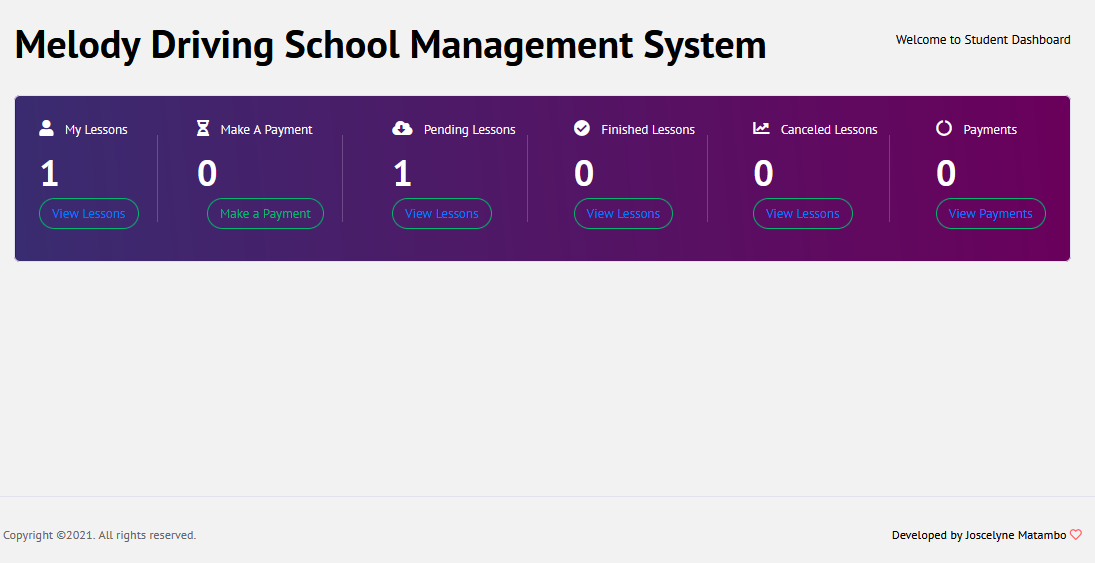


## **4.3 SYSTEM PROCESSES**

## **4.4 SYSTEM OUTPUTS**

The output from the system is either by screen or by hard copies. Output design aims at communicating the results of the processing of the users. The reports are generated to suit the needs of the users. The reports have to be generated with appropriate levels. Presenting the data processed by a computer-based COMPUTER system in an attractive and usable form has become very essential these days’ success and acceptance of a system to some extent depends on good presentation. Therefore, system analyst must know fully how to design output report in an attractive way.





## **4.5 ARCHITECTURE DESIGN**

Joel Henry: Software Project Management, “System architecture design is how the software product is portioned and how the partitions work together to provide functionality.”

Leszek A Maciaszek: Requirements Analysis and System Design, “Architecture design is the description of the system in terms of its modules (components).”

The architectural design involves the layered organization of classes and packages, the assignment of processes to computing facilities, reuse, and component management.

Architectural design resolves the issues with regard to a multi-tier physical architecture as well as with regard to multi-layer logic architecture.

Design comparisons and decisions are hotbeds for ego bumping on the scale of sumo wrestling. Each software engineer knows the best software design, even though none of their designs are the same. Decisions concerning architecture designs will be based upon design qualities such as:

Maintainability: The ease or difficulty of maintaining a network, including correcting defects, improving performance, or adding functionality.

Extendibility: The ease or difficulty of adding function of adding functionality to the system including its network architecture.

Performance: The ability to execute functions fast enough to meet performance goals. (Response time is important to an application.)

Security: The authorization of access to data in a network, which is controlled by the network administrator.

Safety: Isolation of safety-critical components e.g. servers and access points.

Availability: Include redundant components in the architecture so as to make accessibility easier.

Portability: Is a characteristic attributed to a computer program if it can be used in operating systems other than the one in which it was created without requiring major rework.

Testability: the ease or difficulty of testing a software product in order to uncover defects and support corrective activities

These network qualities are influenced by factors such as: Coupling: The degree of connections between classes. It measures the class interdependence. The weaker the coupling the better

Cohesion: The degree of inner self-determination of the class. It measures the strength of the class independence. A highly cohesive class performs one action or achieves a single goal. The stronger the cohesion the better.

Modularity: The characteristic of a network that is divided into a set of units that support ease of network change, promote system understanding, and reduce complexity.

Information hiding (Encapsulation): the characteristic of a network in which details of a specific implementation are not visible outside a portion or segment of that network. This is achieved through the creation of virtual networks popularly known as V-LANS.

Insulation: the characteristic of design and code that allows changes within one portion of the design or a code unit to have no effect on other parts

The architectural design includes decisions about the solution strategy for the client and server aspects of the system. The architectural design is also concerned with the selection of a solution strategy and with modularization of the system. The solution strategy needs to resolve client (user interface) and server (database) issues as well as any middleware needed to “glue” client and server processes.

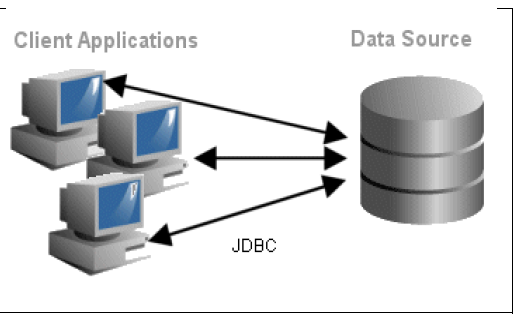
Network Architecture Design

Due to several factors considered the designer is going to use the two-tier network architecture to implement the Melody Driving Management System network.

The end goal is to build a network for convergence––one that has the extensibility to support a variety of new applications in a highly available and secure fashion. While the two-tier architecture represents a simplification of previous designs, it does not represent a compromise in terms of quality, availability, security or management. As the name indicates, the two-tier architecture is a collapsing of layers into intelligent core and unified access

## **4.5.1 Two-Tier Architecture:**

The two-tier architecture is like a client server application. The direct communication takes place between client and server. There is no intermediate between client and server.



## **Table 4.4.1: Two-Tier Architecture Diagram of the proposed system.**

## **4.4.2 Limitations of a Two-Tier Approach**

The two-tier network architecture may not be appropriate for every enterprise. Some enterprises may be limited by physical or organizational constraints. A two-tier network design is ideally suited for enterprises with greenfield environments or with sufficient space and modularity within the building to handle the consolidation and changes necessary to migrate from a three-tier approach. Enterprises with long distances between core and access may find it more cost-effective to include additional network layers to ensure reach of traffic flows. An example of this may be a large manufacturing campus with distributed production locations. In other cases, organizations that occupy historical buildings with older cable plants or restrictions on physical infrastructure changes may also be limited in their ability to cleanly migrate to a two-tier design.

While a two-tier design should decrease the amount of space required for the physical housing of switches, some building layouts may not have the flexibility of providing space where it is required. As a general rule, enterprises with a physical separation between the data center or access layer and the core that is greater than 100 meters for copper runs and 300 meters for multi-mode fiber, could be faced with the relatively more expensive option of using higher powered optics to cover the greater distances. This additional cost should be weighed carefully against the cost of supporting additional network layers. Fundamentally, the largest issues prohibiting two-tier architecture are distance, age of cable plants and the flexibility of the building structure. Assuming none of these are of major concern, the enterprise is best served by collapsing layers and simplifying overall network design.

Performance will be reduced when there are more users. This design is used frequently in decision support systems where the transaction load is light. Two tier software architectures require minimal operator intervention. The two tier architecture works well in relatively homogeneous environments with processing rules (business rules) that do not change very often and when workgroup size is expected to be fewer than 100 users, such as in small businesses.

The most important limitation of the two-tier architecture is that it is not scalable, because each client requires its own database session. The two tier design will scale-up to service 100 users on a network. It appears that beyond this number of users, the performance capacity is exceeded. This is because the client and server exchange "keep alive" messages continuously, even when no work is being done, thereby saturating the network.

Implementing business logic in stored procedures can limit scalability because as more application logic is moved to the database management server, the need for processing power grows. Each client uses the server to execute some part of its application code, and this will ultimately reduce the number of users that can be accommodated.

The two tier architecture limits interoperability by using stored procedures to implement complex processing logic (such as managing distributed database integrity) because stored procedures are normally implemented using a commercial database management system's proprietary language. This means that to change or interoperate with more than one type of database management system, applications may need to be rewritten. Moreover, database management system's proprietary languages are generally not as capable as standard programming languages in that they do not provide a robust programming environment with testing and debugging, version control, and library management capabilities.

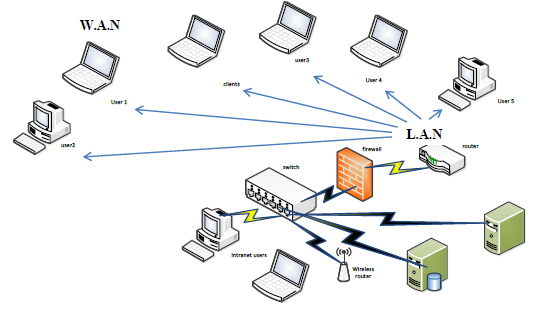
Two tier architectures can be difficult to administer and maintain because when applications reside on the client, every upgrade must be delivered, installed, and tested on each client. The typical lack of uniformity in the client configurations and lack of control over subsequent configuration changes increase administrative workload. The two tiered architecture is not effective running batch programs. The client is typically tied up until the batch job finishes, even if the job executes on the server; thus, the batch job and client users are negatively affected.

## **4.4.3 Benefits of a Two-Tier Approach**

Zimbabwe open university will benefit from a lower network acquisition cost by requiring fewer switches. Installation and maintenance costs are decreased due to the added simplicity of the two-tier design. A simplified network improves the IT organization’s ability to innovate at the application layer and deliver next generation IP applications such as IP Telephony with quality and consistency. Development Issues e.g. Simple structure, Easy to setup and maintain. Business logic and database are physically close, which provides higher performance.

## **4.6 PHYSICAL DESIGN**

Physical design is the hardware layout, where cables are routed, where routers, servers, and workstations are located within a building. Melody Driving School has got an existing infrastructure through which it connects to the internet, so the new system is going to adopt such architecture to implement its network. The following image exhibits the layout of the architecture.



**L.A.N**

**W.A.N**

## ***Table 4.5: Physical diagram of the proposed system.***

***The*** WAN connection is facilitated by a dedicated point to point link of fiber optic since the distance to the remote stations may spun up to more than 500km, hence only the fiber optic link can carter for long distance transmission problems. Fiber optics has a large capacity to carry high speed signals over longer distances without repeaters than other types of cables. The information carrying capacity increases with frequency.

The LAN is comprised of the following devices:

* Router
* Firewall
* Switch
* Wireless router
* Application server
* Network cables (CAT 6)

Router

A router is a device that is going to forward data packets between the LAN and its remote stations, creating an overlay internetwork. A router is connected to two or more outside data lines as shown above and linking them with the LAN. When a data packet comes in on one of the lines, the router reads the address information in the packet to determine its ultimate destination. Then, using information in its routing table or routing policy, it directs the packet to the next network on its journey. Routers perform the "traffic directing" functions on the Internet. A data packet is typically forwarded from one router to another through the networks that constitute the internetwork until it gets to its destination node.

Firewall

A firewall can either be software-based or hardware-based and is used to help keep a network secure. Its primary objective is to control the incoming and outgoing network traffic by analysing the data packets and determining whether it should be allowed through or not, based on a predetermined rule set. The firewall will build a bridge between the LAN that is assumed to be secure and trusted, and another network, usually an external (inter)network, such as the Internet, that is not assumed to be secure and trusted.

Switch

A network switch is a computer networking device that connects network segments or network devices. It commonly refers to a multi-port network bridge that processes and routes data at the data link layer (layer 2) of the OSI model. A switch in an Ethernet-based LAN reads incoming TCP/IP data packets or frames containing destination information as they pass into one or more input ports.

Wireless router

A wireless router is a device that performs the functions of a router but also includes the functions of a wireless access point and a network switch. It is commonly used to provide access to the Internet or a computer network. It does not require a wired link, as the connection is made wirelessly, via radio waves. This device will be used by users of laptops in the organization.

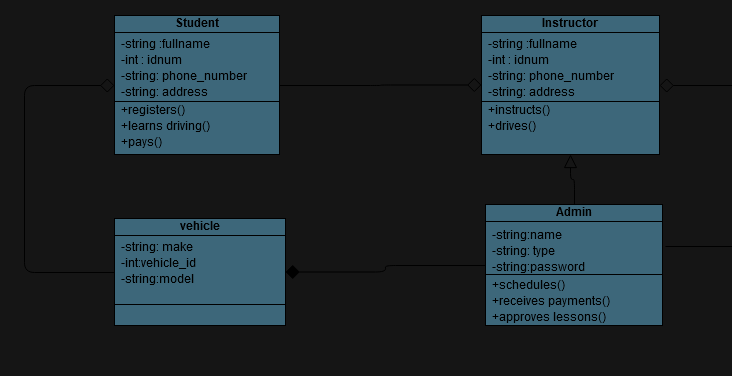
Application server

This is the hosting machine for ZOU Fleet Manager and its database. That is where all the fleet management transactions are going to be done and saved. An application will provide the system with services such as security, data services, transaction support, load balancing, and management of large distributed systems.

Network cables (CAT 6)

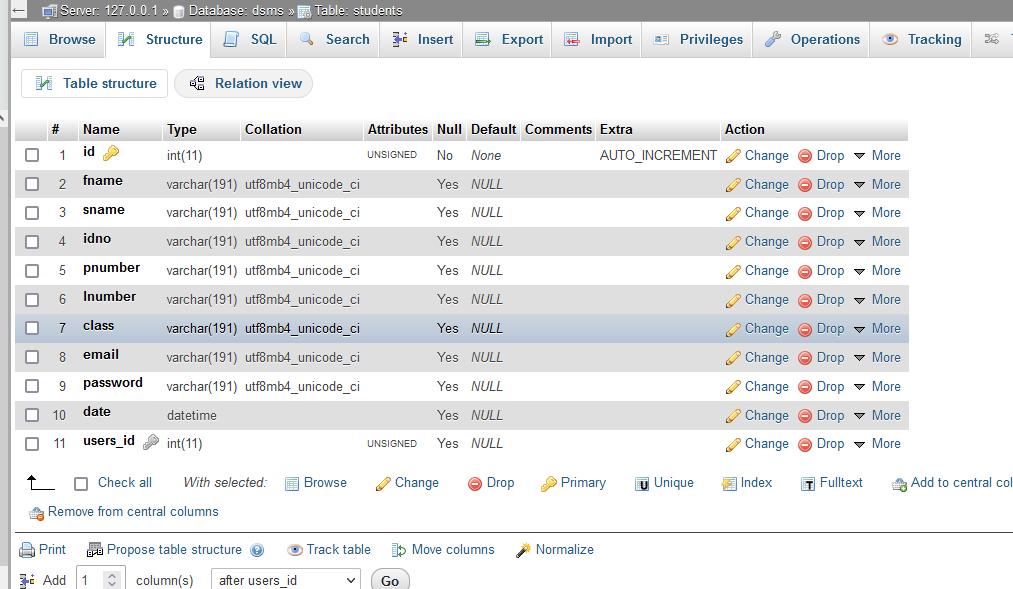
Category 6 cable (Cat 6), is a cable standard for Gigabit Ethernet and other network physical layers. Cat 6 features more stringent specifications for crosstalk and system noise. The cable standard provides performance of up to 250 MHz and is suitable for 10BASE-T, 100BASE-TX (Fast Ethernet), 1000BASE-T/1000BASE-TX (Gigabit Ethernet) and 10GBASE-T (10-Gigabit Ethernet).

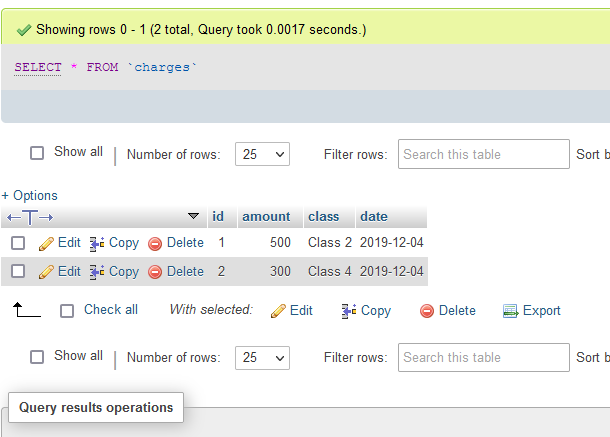
## **4.7 DATABASE DESIGN-entity relationship diagrams, class diagrams ,**

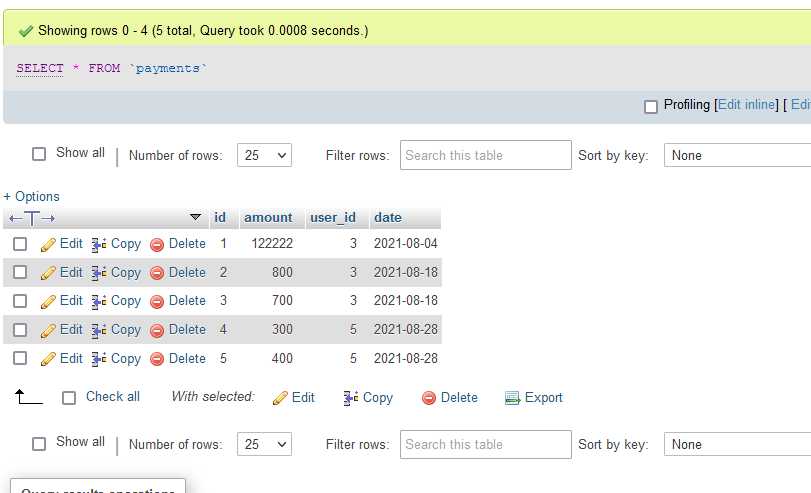


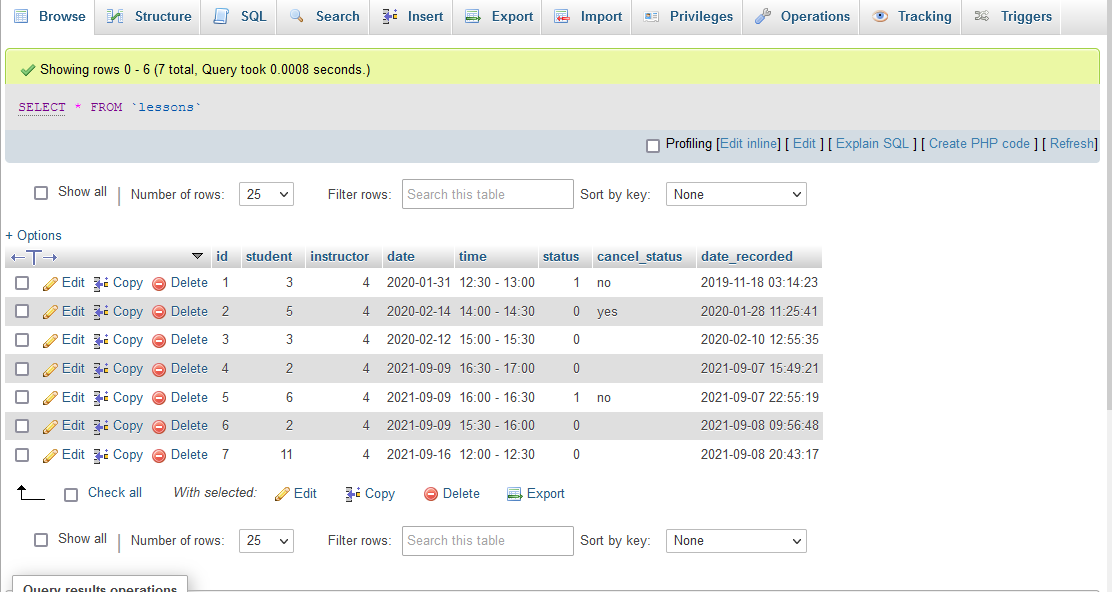
## 

## **table and their fields**



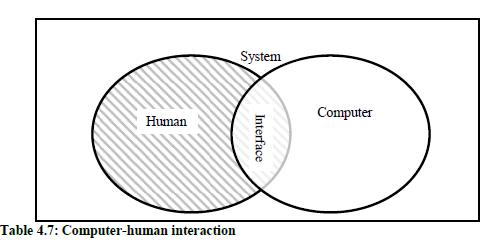






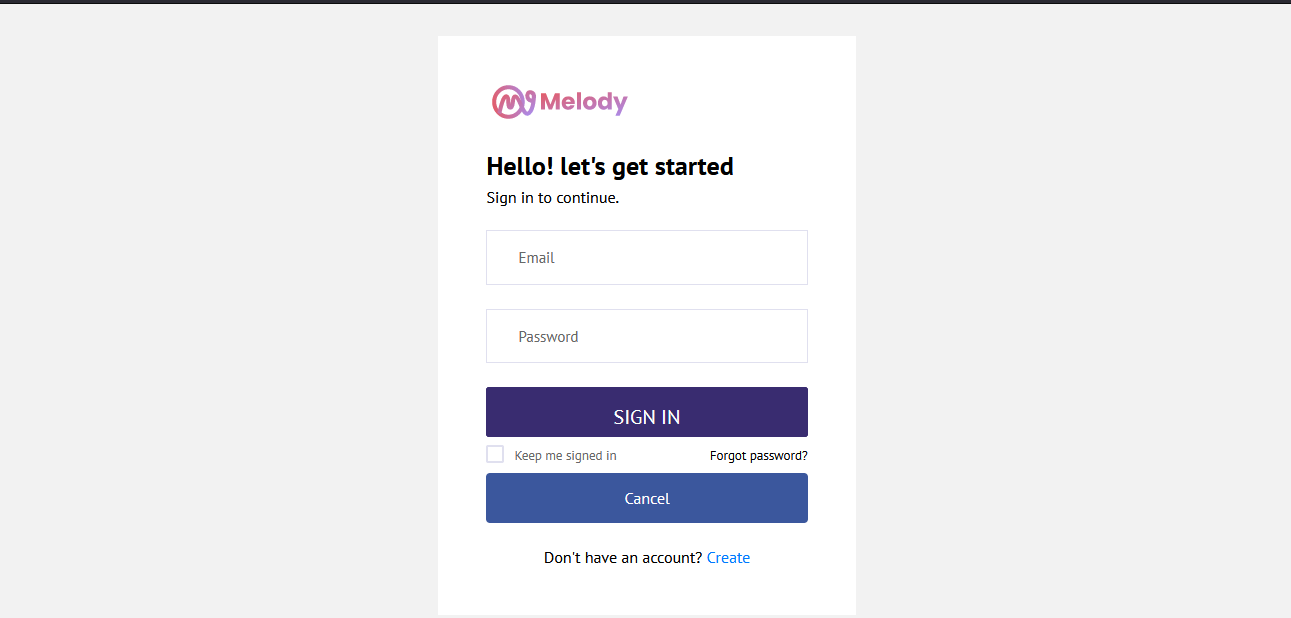
## **4.8 INTERFACE DESIGN**

Deborah J Mayhew: Principles and Guidelines in Software User Interface, “Interface is the means through which two subsystems (the computer and the human) communicate with one another”. Basically, the human is flexible or adaptable and most computer systems are not, inputs must be made in a particular format and outputs are predefined thus for a computer system a human can learn and adapt while the computer cannot. This puts the burden of successful interaction totally on the user hence then need for a computer-human interface. The table below illustrates the relationship between humans, system and computer.



Interface design will take account of the needs, experience and capabilities of the system users through the application of the following design principles although not all principles are applicable to all designs.

|  |  |
| --- | --- |
| ***Principle*** | ***Description*** |
| User familiarity | The interface should use terms and concepts which are drawn from the experience of the people who will make most use of the system. |
| Consistency | The interface should be consistent in that, wherever possible, comparable operations should be activated in the same way. |
| Minimal surprise | Users should never be surprised by the behavior of a system. |
| Recoverability | The interface should include mechanisms to allow users to recover  From errors. |
| User guidance | The interface should provide meaningful feedback when errors  occur and provide context-sensitive user help facilities. |
| User diversity | The interface should provide appropriate interaction facilities for  different types of system user. |



## **4.9 Conclusion**

The basic structure and architecture of the new system has been developed and the next stage is the actual implementation of the system to produce something tangible. This document is going to be the guiding blue print during the coding phase.

# **CHAPTER 5: IMPLEMENTATION AND TESTING**

## **5.0 IMPLEMENTATION: INTRODUCTION**

The implementation stage will involve the installation of software, testing, maintenance and training of users by the developer. The system is then tested to discover any errors so that corrections can be made before it can be installed. Testing process is repeated several times until all identified errors have been corrected.

The system will run parallel to the existing system for two weeks so as not to loose any information should the system fail for any reason. One disadvantage of this method of implementation is that there is duplication of effort imposed upon the user department as it is likely that the same persons have to run both the system during the periods of parallel running.

## **5.1 Coding and Construction**

In this section we will discuss the programming language that will be used for development, programming style, data storage, connection method, processing method as well as the input and output methods.

Coding

The designer used:-

* Php for the connection of my modules to the database.
* HTML and CSS for styling the WEB pages
* JavaScript for validation of all the forms.
* MySql for the database queries

## **5.1.1** **PROGRAM CODE SEGMENTS**

Code for student registration

**<?php  
include** (**"rb.php"**);  
R::*setup*(**'mysql:host=localhost;dbname=dsms'**, **'root'**, **''**); *//for both mysql or mariaDB*$conn=*mysqli\_connect*(**"localhost"**, **"root"**, **""**,**"dsms"**);  
  
**if** (**isset**($\_POST[**'reg'**])){  
  
  
  
 $email=$\_POST[**'email'**];  
 $password=*md5*($\_POST[**'password'**]);  
  
 $init = R::*findOne*(**'users'**, **'email = ?'**, [$email]);  
  
 **if** ($init > 0) {  
 **echo "<script>***alert***('Email already taken by another user!'); *window*.location='register.php'</script>"**;  
 } **else** {  
  
 $users = R::*dispense*(**'users'**);  
 $users->**role** =**"student"**;  
 $users->**email** = $email;  
 $users->**password** = $password;  
 R::*store*($users);  
  
  
 $students = R::*dispense*(**'students'**);  
 $students->**fname** = $\_POST[**'fname'**];;  
 $students->**sname** = $\_POST[**'sname'**];;  
 $students->**idno** = $\_POST[**'idno'**];;  
 $students->**pnumber** = $\_POST[**'pnumber'**];;  
 $students->**lnumber** = $\_POST[**'lnumber'**];;  
 $students->**class** = $\_POST[**'class'**];;  
 $students->**email** = $email;  
 $students->**password** = $password;  
 $students->**date** = *date*(**'Y-m-d H:i:s'**);  
 *// R::store($users);* $users->**ownProductList**[] = $students;  
 R::*store*($users);  
  
 **?>** <**script**>  
 *alert*(**'Succsessfully Saved'**);  
 ***window***.**location** = **"login.php"**;  
 </**script**>  
 **<?php** }  
}  
**?>**<!DOCTYPE **html**>  
<**html lang="en"**>  
  
  
*<!-- Mirrored from www.urbanui.com/melody/template/pages/samples/register.html by HTTrack Website Copier/3.x [XR&CO'2014], Sat, 15 Sep 2018 06:08:53 GMT -->*<**head**>  
 *<!-- Required meta tags -->* <**meta charset="utf-8"**>  
 <**meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no"**>  
 <**title**>Driving School</**title**>  
 *<!-- plugins:css -->* <**link rel="stylesheet" href="vendors/iconfonts/font-awesome/css/all.min.css"**>  
 <**link rel="stylesheet" href="vendors/css/vendor.bundle.base.css"**>  
 <**link rel="stylesheet" href="vendors/css/vendor.bundle.addons.css"**>  
 *<!-- endinject -->  
 <!-- plugin css for this page -->  
 <!-- End plugin css for this page -->  
 <!-- inject:css -->* <**link rel="stylesheet" href="css/style.css"**>  
 *<!-- endinject -->* <**link rel="shortcut icon" href="images/favicon.png"** />  
</**head**>  
  
<**body**>  
 <**div class="container-scroller"**>  
 <**div class="container-fluid page-body-wrapper full-page-wrapper"**>  
 <**div class="content-wrapper d-flex align-items-center auth"**>  
 <**div class="row w-100"**>  
 <**div class="col-lg-4 mx-auto"**>  
 <**div class="auth-form-light text-left p-5"**>  
 <**div class="brand-logo"**>  
 <**img src="images/logo.svg" alt="logo"**>  
 </**div**>  
 <**h4**>Student Registration Form</**h4**>  
 <**h6 class="font-weight-light"**>Signing upff is easy. It only takes a few steps</**h6**>  
 <**form class="pt-3" action="" enctype="multipart/form-data" method="post"**>  
  
 <**div class="form-group"**>  
 <**input type="text" class="form-control form-control-lg" onkeypress="return** *isString*(***event***)**" name="fname" placeholder="Firstname" required**>  
 </**div**>  
 <**div class="form-group"**>  
 <**input type="text" class="form-control form-control-lg" required onkeypress="return** *isString*(***event***)**" name="sname" placeholder="Last Name"**>  
 </**div**>  
 <**div class="form-group"**>  
 <**input type="text" class="form-control form-control-lg" name="idno" placeholder="National ID Number" required**>  
 </**div**>  
 <**div class="form-group"**>  
 <**input type="text" class="form-control form-control-lg" required onkeypress='return** *numbersOnly*(**this**,***event***,**false**,**true**);**' name="pnumber" placeholder="Contact Number" onkeyup="***limit*(**this**);**" onkeydown="***limit*(**this**);**"**>  
 <**div class=" alert-warning" id='result'**></**div**>  
 <**div class=" alert-warning" id='char'**></**div**>  
 </**div**>  
 <**div class="form-group"**>  
 <**input type="text" class="form-control form-control-lg" id="exampleInputUsername1" name="lnumber" placeholder="Provisional Licence Number"**>  
 </**div**>  
  
 <**div class="form-group"**>  
 <**select class="form-control form-control-lg" id="exampleFormControlSelect2" name="class"**>  
 <**option**>Select Class</**option**>  
 <**option value="2"**>CLass 2</**option**>  
 <**option value="4"**>Class 4</**option**>  
 </**select**>  
 </**div**>  
 <**div class="form-group"**>  
 <**input type="email" name="email" class="form-control form-control-lg" id="exampleInputEmail1" placeholder="Email" required**>  
 </**div**>  
 <**div class="form-group"**>  
 <**input type="password" name="password" class="form-control form-control-lg" id="exampleInputPassword1" placeholder="Password" required**>  
 </**div**>  
*<!-- <div class="mb-4">-->  
<!-- <div class="form-check">-->  
<!-- <label class="form-check-label text-muted">-->  
<!-- I accept: <input type="checkbox" value="0" name="agree">-->  
<!-- <div class="alert-danger" id='result'></div>-->  
<!-- </label>-->  
<!-- </div>-->  
<!-- </div>-->* <**div class="mt-3"**>  
 <**input value="Register" class="btn btn-block btn-primary btn-lg font-weight-medium auth-form-btn" type="submit" name="reg"**>  
 </**div**>  
 <**br**>  
 <**div class="mb-2"**>  
 <**input type="button" onClick="*window***.**location**.**href**=**'index.php'" value="Cancel" class="btn btn-block btn-facebook auth-form-btn"**>  
 </**div**>  
 <**div class="text-center mt-4 font-weight-light"**>  
 Already have an account? <**a href="login.php" class="text-primary"**>Login</**a**>  
 </**div**>  
 </**form**>  
 </**div**>  
 </**div**>  
 </**div**>  
 </**div**>  
 *<!-- content-wrapper ends -->* </**div**>  
 *<!-- page-body-wrapper ends -->* </**div**>  
 *<!-- container-scroller -->  
 <!-- plugins:js -->* <**script src="vendors/js/vendor.bundle.base.js"**></**script**>  
 <**script src="vendors/js/vendor.bundle.addons.js"**></**script**>  
 *<!-- endinject -->  
 <!-- inject:js -->* <**script src="js/off-canvas.js"**></**script**>  
 <**script src="js/hoverable-collapse.js"**></**script**>  
 <**script src="js/misc.js"**></**script**>  
 <**script src="js/settings.js"**></**script**>  
 <**script src="js/todolist.js"**></**script**>  
 *<!-- endinject -->*</**body**>  
  
  
*<!-- Mirrored from www.urbanui.com/melody/template/pages/samples/register.html by HTTrack Website Copier/3.x [XR&CO'2014], Sat, 15 Sep 2018 06:08:53 GMT -->*</**html**>  
<**script type="text/javascript"**>  
 **function** *limit*(element){  
 **var** max\_chars = 10 ;  
 **if**(element.**value**.**length** > max\_chars){  
 element.**value**= element.**value**.substr(0,max\_chars);  
 ***document***.getElementById(**"result"**).**innerHTML** = **'Phone number should be 10 digits'**;  
  
 }  
 }  
  
 **function** *numbersOnly*(Sender,evt,isFloat,isNegative) {  
 **if**(Sender.**readOnly**) **return false**;  
  
 **var** key = evt.**which** || !***window***.**event** ? evt.**which** : ***event***.**keyCode**;  
 **var** value = Sender.**value**;  
  
 **if**((key == 46 || key == 44) && isFloat){  
 **var** selected = ***document***.**selection** ? ***document***.**selection**.createRange().**text** : **""**;  
  
  
 **if**(selected.**length** == 0 && value.indexOf(**"."**) == -1 && value.**length** > 0) Sender.**value** += **"."**;  
  
 **return false**;  
 }  
 **if**(key == 45) { *// minus sign '-'* **if**(!isNegative)  
 *// document.getElementById("char").innerHTML = '3 Phone number should contain Integers only';* **return false**;  
 **if**(value.indexOf(**'-'**)== -1) Sender.**value** = **'-'**+value; **else** Sender.**value** = value.substring(1);  
 **if**(Sender.**onchange** != **null**) {  
 **if**(Sender.fireEvent){  
 Sender.**fireEvent**(**'onchange'**);  
 } **else** {  
 **var** e = ***document***.createEvent(**'HTMLEvents'**);  
 e.initEvent(**'change'**, **false**, **false**);  
 Sender.dispatchEvent(e);  
  
 }  
 }  
  
 **var** begin = Sender.**value**.indexOf(**'-'**) < -1 ? 1 : 0;  
 **if**(Sender.setSelectionRange){  
 Sender.setSelectionRange(begin,Sender.**value**.**length**);  
 } **else** {  
 **var** range = Sender.createTextRange();  
 range.moveStart(**'character'**,begin);  
 range.select();  
 }  
  
 **return false**;  
 *//document.getElementById("char").innerHTML = '2 Phone number should contain Integers only';* }  
 **if**(key > 31 && (key < 48 || key > 57))  
 *// alert('1 Phone number should contain Integers only');  
 // document.getElementById("char").innerHTML = '1 Phone number should contain Integers only';* **return false**;  
 }  
  
 **function** *val*(){  
 **if**(frm.**password**.**value** == **""**)  
 {  
 *alert*(**"Enter the Password."**);  
 frm.**password**.focus();  
 **return false**;  
 }  
 **if**((frm.**password**.**value**).**length** < 8)  
 {  
 *alert*(**"Password should be minimum 8 characters."**);  
 frm.**password**.focus();  
 **return false**;  
 }  
  
 **if**((frm.**password**.**value**).**length** > 20)  
 {  
 *alert*(**"Password should be maximum 20 characters."**);  
 frm.**password**.focus();  
 **return false**;  
 }  
  
 **if**(frm.confirmpassword.**value** == **""**)  
 {  
 *alert*(**"Enter the Confirmation Password."**);  
 **return false**;  
 }  
 **if**(frm.confirmpassword.**value** != frm.**password**.**value**)  
 {  
 *alert*(**"Password confirmation does not match."**);  
 **return false**;  
 }  
  
 **return true**;  
 }  
 **function** *isString*(evt){  
 **var** charCode = (evt.**which**) ? evt.**which** : ***event***.**keyCode  
 if** (charCode > 31 && (charCode < 48 || charCode > 57)){  
 **return true**;  
 }  
 **else**{  
 *alert*(**' must have alphanumeric characters only'**);  
 *// document.getElementById("num").innerHTML = "User address must have alphanumeric characters only ";* **return false**;  
 }  
 }  
</**script**>

## **5.2 TESTING**

Testing is a process of executing a program with the interest of finding an error. A good test is one that has high probability of finding the yet undiscovered error. Testing should systematically uncover different classes of errors in a minimum amount of time with a minimum number of efforts.

Software Testing is the process of executing a program or system with the intention of detecting errors (Brien,1996). Testing is usually done for to improve quality, for Verification and Validation (VV) and for reliability estimation. Testing of the system was done so that errors can be discovered and fixed before the system is put to use. This is very crucial so that we make sure that the system is perfect and does not give problems to the end users.

Two classes of inputs are provided to test the process

1. A software configuration that includes a software requirement specification, a design specification and source code.
2. A software configuration that includes a test plan and procedure, any testing tool and test cases and their expected results.

## **5.2.0 SYSTEM TESTING**

System testing is testing that was conducted on the complete. System testing took as its input, all of the integrated software components. The aim behind system testing was to detect defects both within the inter-assemblages and also within the system as a whole.

## **5.2.1 TESTING PROCESS**

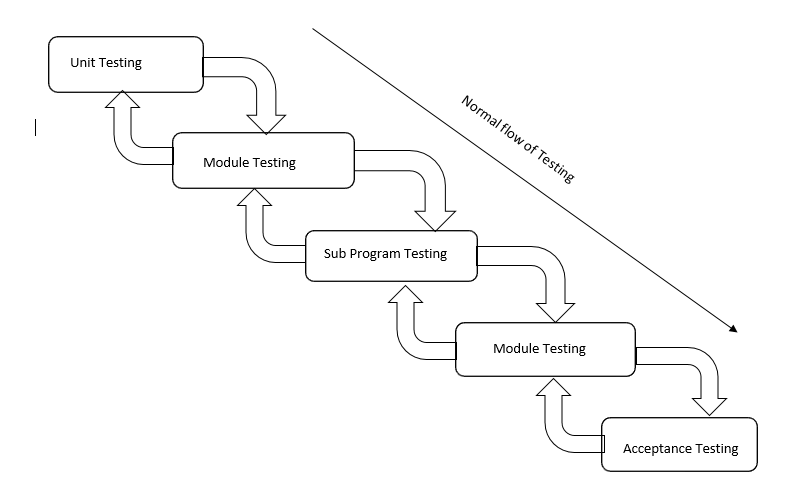


Figure 5.1.1 Testing process

**5.2.1.1 Unit Testing**

During unit testing each form on the system was tested to see if it performed as per requirement. Each component of the system was tested individually to ensure its functionality and integrity.

A test case design method that uses the control structure of the procedural design to derive test cases. By using white box testing methods, the developer could derive test cases that:

* Guarantee that all independent paths within a module have been exercised at least once.
* Exercise all logical decisions on their true and false sides.
* Execute all loops at their boundaries and within their operational bounds.
* Exercise internal data structures to ensure their validity.

During unit testing some errors were raised and all of them were rectified and handled well. The result was quiet satisfactory and it worked well.

**5.2.1.2 Module Testing**

In this case each of the four modules, each module was created separately and tested separately then linked to form one. This testing strategy was instrumental in the understanding of all the validation that needed to be coded as per module. Module testing was rather the test of related unit components of the system. Execution paths, error-handling paths, normal, abnormal and extreme data were thoroughly tested. The idea behind was to check if the system was updating the accounts before the customer left. Also to find out if the program would make subtractions in database once the customer pays his or her bill. The system was capable of producing the systems objectives and therefore concluded to be working properly.

**Subsystem Testing**

System modules that were related formed system subsystems these were tested for data, stress, and error integrity. Database was also tested as a subsystem of its own. This is also called link testing. It tests a collection of modules, which have been integrated into subsystems. It ensures job streams are correct. There is detection of interface mismatches and demanding exercises of the interface between programs.

**5.2.1.3 System Testing**

System testing is the testing of all the integrated modules that makes up the system to check if it was performing the required functions. The system is able to add new records of student. All queries are executed by the system.

**5.2.1.4 Integration Testing**

Integration testing is a system technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested modules and build a program structure that has been dictated by design. Bottom-up integration is the traditional strategy used to integrate the components of a software system into functioning whole. Bottom-up integration consists of unit test followed by testing of the entire system. A sub-system consists of several modules that communicated with other defined interface. The system was done the integration testing. All the modules were tested for their compatibility with other modules. They test was almost successful. All the modules coexisted very well, with almost no bugs. All the modules were encapsulated very well so as to not hamper the execution of other modules.

**5.2.1.5 Security Testing**

It attempts to verify that protection mechanisms built into a system will in fact protect it from improper penetration. The system’s security must of course be tested from in vulnerability form frontal attack.

**5.2.1.6 Black Box Testing**

Black box testing is done to find out the following information as shown in below:

1. Incorrect or missing functions.
2. Interface errors.
3. Errors or database access.
4. Performance error.
5. Termination error.

The mentioned testing is carried out successfully for this application according to the user’s requirement specification.

**5.2.1.7 Test Data Output**

After preparing test data, the system under study is tested using the test data. While testing the system using test data, errors are again uncovered and corrected by using above testing and corrections are also noted for future use.

**5.2.1.8 Performance testing**

Testing conducted to evaluate the compliance of a system or component with specified requirements. For example a performance requirement might state that the system automatically updates in less than 3 seconds.

**5.2.1.9 Usability testing**

Testing to evaluate the extent to which a user can learn to operate, prepare inputs for, and interpret outputs of a system or component. This can be done by all system users interacting with the system.

**5.2.1.10 Acceptance testing**

The goal of system testing is to confirm that the system is complete, meets the organizations’ needs that prompted the system to be developed, and is acceptable to users. The system is tested with data supplied by users rather than simulated data. Acceptance testing helped reveal errors and omissions in the system requirements definition because real data tests the system in different way from the test data. Acceptance testing also revealed requirements problems where the system’s facilities did not really meet the user’s needs. Acceptance testing is done in two stages: alpha and beta testing.

## **5.3 SECURITY**

Security is implemented through the use of physical and software measures. These include passwords, security user levels, burglar bars, tags and security guards. System security aims at protecting the system from vandalism, intentional negative incidences like fraud, theft and accidents. The objective is to avoid the occurrence of damage to the system or minimize the effects of an attack. These accidents can occur to hardware, software, data and the network.

We looked at how the risks may occur and then decided on ways which we can control and protect these risks. A threat to the system is any potential adverse occurrence that can do harm to the application or its data such as a computer virus or unexpected natural disaster or disruption. Disruptions occur when there is power failure or user mistakes causing the network to cut or cease functioning. Some disruptions may also be caused by destruction of data for example a virus may destroy files and other destruction may be catastrophic such as natural.

An example of one of the security measures in the system is the use of password, username and access level. If the user or any unauthorized user tries to login and if any of the text fields is wrong. The system will display an error message and prompt the user to log on again as illustrated by the example below.

## **5.4 INSTALLATION**

Installation is when the developed system is being installed. Users are changing from using the old manual system to using the new system. In this phase the required software is installed on the appropriate hardware converting from the as-is system to the to-be system. Users are moved from using the old system to using the new system.

Steps For Application Software Installation

* The system is installed from the software C.D
* Install system to path. (It is recommended that the system be installed to the program files folder.)
* Verify that the system is installed properly that is all folders are installed.
* Add the system data source.
* Connect data source to the database on the main server housing the system database.

Database Installation

The database management system (DBMS) will use MySQL Server. The DBMS is installed on a central server, which will serve all the other terminals and thus is the Central database

## **5.5 TRAINING**

Training of the user will be done on a personal level as few people hence the intricacy of the system can be taught to the users faster and more effectively at the terminals.

Training is done at two levels

* Module level: this is for the particular modules that concern the particular users.
* System level. This is for management who must appreciate the development of the system and its function. Users who have access rights to all modules also must be versed with the entire system as it functions.

## **5.6 Maintenance**

Following implementation of the system, there comes a time when there is need for its

Review and this is normally done on monthly basis. It is carried out in this phase.

The system should be maintained to make sure that it still conforms to the specifications. Reviews should be done periodically and if the specifications or environment changes then the system should be upgraded. Maintenance of the system is an ongoing process.

### **5.6.1 Types of maintenance**

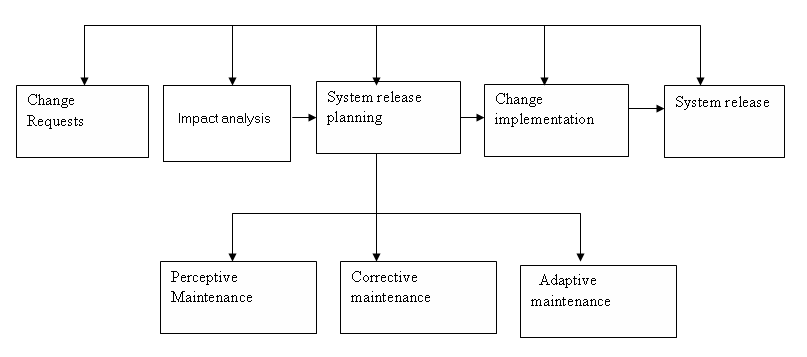
Maintenance is the process of monitoring, evaluating and modifying the operational information system to make necessary improvements. There are a number of approaches to the maintenance process. The need for maintenance will arise as a result of many different situations occurring throughout the life of the organization and these can include the following:

* If the organization grows, it means a change in the way it operates. The system will therefore require modifications to mirror the changes.
* The users of the system may require additional information that will help perform the system tasks better thus the need to make a few additions to meet such requirements.
* It is possible that initially, the requirements may have been incorrectly specified and when the system is now in operation, corrections will have to be made.

We implemented the system and examined it to see if it is meeting the objectives set out in the original specifications. The system analysts and the project team concluded that the system was meeting the specified objectives but since the environment changes so does the customer requirements.

### **5.6.2 Structured Maintenance**

The maintenance process will be carried out as follows:



### Figure 5.6 Structured Maintenance

A change request form is submitted and analysed. If the request is accepted a new release is planned. Release planning involves elements of adaptive, corrective and perfective maintenance.

There are three types of system maintenance are explained:

**5.6.2.1 Corrective maintenance**

This is the maintenance to correct the software faults. Coding errors ere usually relatively cheap to correct, design errors are more expensive as they may involve the rewriting of several program components. Requirements errors are the most expensive to report because of extensive system redesign which may be necessary.

#### **5.6.2.2 Adaptive maintenance**

This is the maintenance to adopt the software to a different operating environment. This type of maintenance is required when some aspect of the system’s environment such as hardware, the platform operating system or other support software changes. The application system must be modified to adapt it to cope with these environment changes.

#### **5.6.2.4 Perfective maintenance**

This is maintenance to add or modify the system’s functionality. This type of maintenance is necessary when the system requirements change in response to organizational or business change. The scale of the changes required to the software is often much greater than for other types of maintenance.

### **5.6.3 Unstructured Maintenance**

**Change**

**request**

**Deliver**

**Repaired**

**System**

**Modify**

**Source**

**code**

**Analyse**

**Source**

**code**

**Figure 5.6 Unstructured maintenance diagram**

At times a system would need a quick repair for the business to run thus the need for this type of maintenance. This is where a request is given by users on an area where a change is requested and the developer analyses the source cone, make changes as per user request and deliver the modified version.

### **5.6.4 Recommendations for maintenance**

The management was taught that maintenance is an important process to be given good attention if the system is to live longer while delivering goods.

The system is important because:

1. A system continues to changes and evolves as it is used.
2. The changes were arising from the change and evolve as it is used.
3. The changes were arising from change request due to the problems. Reports from operating groups who identifies the bugs in the system that must be fixed.
4. Changes were arising from users.

## **5.7 System evaluation**

System evaluation includes measuring the final system against its initial performance goals as well as performing ongoing testing to see that the system continues to meet those goals.

System evaluation is required to assess whether the system is meeting the objectives it was designed to meet, such as:

1. To provide online registration of students so that they will be a digital repository containing all registered students
2. To develop online scheduling of the lessons for students
3. To provide online payment platform

## **5.8 File conversion and System changeover**

### **5.8.1 File conversion**

After the users have verified the results of the testing process the files are the transferred to the new system. Programs need to be designed which would do this conversion process. As the users would be using the system during working hours, the files will be in use, thus the conversion will be done during the weekends or outside working hours.

File conversion follows soon after operational environment has been established and training has been conducted. Under the normal schedule we had set, it took us three weeks to transfer manual data into the new system.

### **5.8.2 System Changeover**

It is the technical process by which the new system replaces the old system. There are a number of approaches to system changeover namely:

#### **5.8.2.1 Direct Changeover**

The old system ceases to function and is replaced immediately by the new system.

##### **Advantages of direct change over**

* Efficient method in so far as it minimizes duplication of work
* Less costs as only one system would be in operation

##### **Disadvantages of direct change over**

* New system may not be entirely correct
* It is difficult to make the system operational when some errors are identified after changeover and need correction
* Requires careful planning, testing and attention to operational detail

#### **5.8.2.2 Parallel Running**

The new and old systems run in parallel for a short period then the old ceases operation after sometime.

##### **Advantages of parallel running**

* Low risk as a results can be verified and a back up option exists

##### **Disadvantages of parallel running**

* Relatively high costs as both systems are in operation at the same time for some time
* Method cannot be used for systems which are not similar

#### **5.8.2.3 Pilot Operation**

The old and new systems operate at the same time but at a pilot site. The rest part of the organization continues using the old system.

##### Advantages of pilot operation

* Moderate costs as only a chosen site would be running two systems at once
* A moderate risk of failure as the new system is only installed at the pilot site.

The developed system is placed in actual site and tested before actual use to check if the system operates according to the set aims and objectives. System processes are checked to see if they perform according to the user requirements.

#### **5.8.2.4 Phased changeover**

With this, the system is implemented in stages or modules across the organization. Phased changeover gives part of the system to the organization and cost is relatively moderate as the system is implemented in stages rather than all at once. Risk is also very moderate because the risk associated is limited to the module being implemented.

## **5.9 System review**

Both the system developer and the system users do this. It is a review of the system performance and this will be done annually. The review will be based on determining the necessary changes needed to be done to the system so as to make it fulfil the new user’s needs, which may have up due to changes in the user-operating environment.

## **5.10 RECOMMENDATIONS**

The researchers of the study highly recommend the implementation of Driving School Management System in order for them to automate processes such as enrolling and recording information of the students, scheduling the driving session of the students, providing payment options to the students and all other important processes. The system is highly recommended for its efficiency and reliability that can be rendered to the schools in automating the processes of their daily transactions. The researchers of the project highlighted that the intended users should have enough knowledge on how the developed system works.

The researchers of the study recommend the following:

1. The developed system should be used by driving school to assist them in conveniently carrying out their tasks.
2. The driving school management system should be implemented by the school in order to improve the service and experience of their students and other stakeholders.

## **5.11 CONCLUSION**

The development of the new System has eluded some key factors in the industry such as a sense of responsibility, time management and planning within limited resources. Given the fact that all the project objectives were met in a constrained time frame and limited resources it can be concluded that the project was a success. This system has a lot of room for expansion and only marks the beginning of automated expertise in the field.

The researchers of the study concluded that the developed Online Driving School Management System will provide an effective way of managing the different records in a driving school. The system will eliminate all the problems encountered in the manual method which will help the school better improve their daily operations. By implementing the system, the school can render an improved service and satisfying experience to their students and other stakeholders.

APPENDIX: User Manual

APPENDIX A

1. Sample Interview Questions
2. Sample Observation sheets
3. Sample questionnaire questions

APPENDIX B

1. Sample program code

*User and Admin dashboard code*

**<?php***session\_start*();  
**include** (**"header.php"**);  
  
$id= $\_SESSION[**'id'**];  
*//include ("rb.php");*$yes= **'yes'**;  
*//R::setup('mysql:host=localhost;dbname=dsms', 'root', ''); //for both mysql or mariaDB*@$init = R::*count*(**'payments'**);  
@$init\_stu = R::*count*(**'payments'**,**'user\_id=?'**,[$id]);  
@$init = R::*count*(**'instructors'**);  
@$finished = R::*count*(**'lessons'**,**'status=?'**,[1]);  
@$finished\_stu = R::*count*(**'lessons'**,**'status=? AND student=?'**,[1,$id]);  
@$finished\_ins = R::*count*(**'lessons'**,**'status=? AND instructor=?'**,[1,$id]);  
@$pending = R::*count*(**'lessons'**,**'status=?'**,[0]);  
@$pending\_stu = R::*count*(**'lessons'**,**'status=? AND student=?'**,[0,$id]);  
@$pending\_ins = R::*count*(**'lessons'**,**'status=? AND instructor=?'**,[0,$id]);  
@$cancel = R::*count*(**'lessons'**,**'cancel\_status=?'**,[$yes]);  
@$cancel\_stu = R::*count*(**'lessons'**,**'cancel\_status=? AND student=?'**,[$yes,$id]);  
@$cancel\_ins = R::*count*(**'lessons'**,**'cancel\_status=? AND instructor=?'**,[$yes,$id]);  
@$all = R::*count*(**'lessons'**);  
@$all\_student\_lessons = R::*count*(**'lessons'**,**'student=?'**,[$id]);  
@$all\_inst\_lessons = R::*count*(**'lessons'**,**'instructor=?'**,[$id]);  
@$initt = R::*count*(**'instructors'**);  
@$initt = R::*count*(**'instructors'**);  
**if**(!**isset**($\_SESSION[**'role'**])){  
  
 **print** (**"<script>*window*.location.assign('login.php')</script>"**);  
  
  
  
 }  
 *// }* **?>** <**div class="main-panel" xmlns="http://www.w3.org/1999/html"**>  
 <**div class="content-wrapper"**>  
 <**div class="page-header"**>  
 <**h1 align="center"**>Melody Driving School Management System</**h1**>  
 <**p**>  
  
 **<?php  
 if** ($\_SESSION[**'role'**] == **"student"**) {  
  
 **echo "Welcome to Student"**;  
 }  
 **?>  
 <?php  
 if** ($\_SESSION[**'role'**] == **"admin"**) {  
 **echo "Welcome to Admin"**;  
 }  
 **?>  
 <?php  
 if** ($\_SESSION[**'role'**] == **"instructor"**) {  
 **echo "Welcome to Instructor"**;  
 }  
 **?>** Dashboard  
 </**p**>  
 </**div**>  
 **<?php  
 if** ($\_SESSION[**'role'**] == **"student"**) {  
 **?>** <**div class="row grid-margin"**>  
 <**div class="col-12"**>  
 <**div class="card card-statistics"**>  
 <**div class="card-body"**>  
 <**div class="d-flex flex-column flex-md-row align-items-center justify-content-between"**>  
 <**div class="statistics-item"**>  
 <**p**>  
 <**i class="icon-sm fa fa-user mr-2"**></**i**>  
 My Lessons  
 </**p**>  
 <**h2**>**<?php echo** $all\_student\_lessons **?>**</**h2**>  
 <**label class="badge badge-outline-success badge-pill"**><**a  
 href="mylessons.php"**>View Lessons </**a**></**label**>  
 </**div**>  
 <**div class="statistics-item"**>  
 <**p**>  
 <**i class="icon-sm fas fa-hourglass-half mr-2"**></**i**>  
 Make A Payment  
 </**p**>  
 <**h2**>**<?php echo** $init\_stu **?>**</**h2**>  
 <**section class="section bg-gray" id="section-modal"**>  
 <**div class="container"**>  
  
 <**label class="badge badge-outline-success badge-pill" data-toggle="modal" data-target="#Chrome"**>Make a Payment</**label**>  
  
  
 *<!-- Modal -->* <**div class="modal fade" id="Chrome" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true"**>  
 <**div class="modal-dialog" role="document"**>  
 <**div class="modal-content"**>  
 <**div class="modal-header"**>  
<**form action="payment.php" method="post" enctype="multipart/form-data"** >  
 <**label**>Enter Amount</**label**>  
 <**input type="text" name="amount" required placeholder="Enter Amount to Pay"**>  
 <**h5 class="modal-title" id="exampleModalLabel"**>Make A payment for your Plan</**h5**>  
 <**button type="button" class="close" data-dismiss="modal" aria-label="Close"**>  
 <**span aria-hidden="true"**>**&times;**</**span**>  
 </**button**>  
 <**INPUT type="submit" value="Pay" class="button" class="btn btn-primary"** />  
  
 </**div**>  
  
 </**div**>  
 </**div**>  
 </**div**>  
 </**form**>  
  
 </**div**>  
 </**section**>  
  
 </**div**>  
 <**div class="statistics-item"**>  
 <**p**>  
 <**i class="icon-sm fas fa-cloud-download-alt mr-2"**></**i**>  
 Pending Lessons  
 </**p**>  
 <**h2**>**<?php echo** $pending\_stu **?>**</**h2**>  
  
 <**label class="badge badge-outline-success badge-pill"**><**a  
 href="lessons.php"**>View Lessons </**a**></**label**>  
 </**div**>  
 <**div class="statistics-item"**>  
 <**p**>  
 <**i class="icon-sm fas fa-check-circle mr-2"**></**i**>  
 Finished Lessons  
 </**p**>  
 <**h2**>**<?php echo** $finished\_stu **?>**</**h2**>  
  
 <**label class="badge badge-outline-success badge-pill"**><**a  
 href="finished.php"**>View Lessons </**a**></**label**>  
 </**div**>  
 <**div class="statistics-item"**>  
 <**p**>  
 <**i class="icon-sm fas fa-chart-line mr-2"**></**i**>  
 Canceled Lessons  
 </**p**>  
 <**h2**>**<?php echo** $cancel\_stu **?>**</**h2**>  
  
 <**label class="badge badge-outline-success badge-pill"**><**a  
 href="cancelled.php"**>View Lessons </**a**></**label**>  
 </**div**>  
 <**div class="statistics-item"**>  
 <**p**>  
 <**i class="icon-sm fas fa-circle-notch mr-2"**></**i**>  
 Payments  
 </**p**>  
 <**h2**>**<?php echo** $init\_stu **?>**</**h2**>  
  
 <**label class="badge badge-outline-success badge-pill"**><**a  
 href="payments.php"**>View Payments </**a**></**label**>  
 </**div**>  
 </**div**>  
 </**div**>  
 </**div**>  
 </**div**>  
 </**div**>  
 **<?php** }  
 **?>  
 <?php  
 if** ($\_SESSION[**'role'**] == **"admin"**) {  
 **?>** <**div class="row grid-margin"**>  
 <**div class="col-12"**>  
 <**div class="card card-statistics"**>  
 <**div class="card-body"**>  
 <**div class="d-flex flex-column flex-md-row align-items-center justify-content-between"**>  
 <**div class="statistics-item"**>  
 <**p**>  
 <**i class="icon-sm fas fa-circle-notch mr-2"**></**i**>  
 Lessons  
 </**p**>  
 <**h2**>**<?php echo** $all**?>**</**h2**>  
 <**label class="badge badge-outline-success badge-pill"**><**a  
 href="lessons.php"**>View Lessons </**a**></**label**>  
 </**div**>  
*<!-- <div class="statistics-item">-->  
<!-- <p>-->  
<!-- <i class="icon-sm fa fa-user mr-2"></i>-->  
<!-- -->  
<!-- </p>-->  
<!-- <h2>54000</h2>-->  
<!-- <a href="lessons.php">View </a>-->  
<!-- </div>-->* <**div class="statistics-item"**>  
 <**p**>  
 <**i class="icon-sm fas fa-hourglass-half mr-2"**>Payments</**i**>  
  
 </**p**>  
 <**h2**>**<?php echo** $init**?>**</**h2**>  
 <**label class="badge badge-outline-success badge-pill"**>  
 <**a href="payments.php"**>View Payments </**a**>  
 </**label**>  
 </**div**>  
 <**div class="statistics-item"**>  
 <**p**>  
 <**i class="icon-sm fas fa-cloud-download-alt mr-2"**></**i**>  
 Pending Lessons  
 </**p**>  
 <**h2**>**<?php echo** $pending**?>**</**h2**>  
 <**label class="badge badge-outline-success badge-pill"**>  
 <**a href="lessons.php"**>View Pending Lessons </**a**>  
 </**label**>  
 </**div**>  
 <**div class="statistics-item"**>  
 <**p**>  
 <**i class="icon-sm fas fa-check-circle mr-2"**></**i**>  
 Finished Lessons  
 </**p**>  
 <**h2**>**<?php echo** $finished **?>**</**h2**>  
 <**label class="badge badge-outline-success badge-pill"**>  
 <**a href="finished.php"**>View Pending Lessons </**a**>  
 </**label**>  
 </**div**>  
 <**div class="statistics-item"**>  
 <**p**>  
 <**i class="icon-sm fas fa-chart-line mr-2"**></**i**>  
 Cancelled Lessons  
 </**p**>  
 <**h2**>**<?php echo** $cancel**?>**</**h2**>  
 <**label class="badge badge-outline-success badge-pill"**>  
 <**a href="cancelled.php"**>View Cancelled Lessons </**a**>  
 </**label**>  
 </**div**>  
 <**div class="statistics-item"**>  
 <**p**>  
 <**i class="icon-sm fas fa-circle-notch mr-2"**></**i**>  
 Settings  
 </**p**>  
 <**h2**>7500</**h2**>  
 <**label class="badge badge-outline-success badge-pill"**>  
 <**a href="charge\_lessons.php"**> Settings </**a**>  
 </**label**>  
  
 </**div**>  
 </**div**>  
 </**div**>  
 </**div**>  
 </**div**>  
 </**div**>  
 **<?php** }  
 **?>  
 <?php***// if ($\_SESSION['role'] == "instructor") {  
//* **?>***<!-- <div class="row grid-margin">-->  
<!-- <div class="col-12">-->  
<!-- <div class="card card-statistics">-->  
<!-- <div class="card-body">-->  
<!-- <div class="d-flex flex-column flex-md-row align-items-center justify-content-between">-->  
<!-- <div class="statistics-item">-->  
<!-- <p>-->  
<!-- <i class="icon-sm fa fa-user mr-2"></i>-->  
<!-- My Lessons-->  
<!-- </p>-->  
<!-- <h2>-->***<?php** *//echo $all\_inst\_lessons* **?>***<!--</h2>-->  
<!-- <label class="badge badge-outline-success badge-pill"> <a-->  
<!-- href="mylessons.php">View Lessons</a></label>-->  
<!-- </div>-->  
<!---->  
<!-- <div class="statistics-item">-->  
<!-- <p>-->  
<!-- <i class="icon-sm fas fa-cloud-download-alt mr-2"></i>-->  
<!-- Pending Lessons-->  
<!-- </p>-->  
<!-- <h2>-->***<?php** *//echo $pending\_ins* **?>***<!--</h2>-->  
<!-- <label class="badge badge-outline-success badge-pill"><a-->  
<!-- href="lessons.php">View Lessons</a></label>-->  
<!-- </div>-->  
<!-- <div class="statistics-item">-->  
<!-- <p>-->  
<!-- <i class="icon-sm fas fa-check-circle mr-2"></i>-->  
<!-- Finished Lessons-->  
<!-- </p>-->  
<!-- <h2>-->***<?php** *//echo $finished\_ins* **?>***<!--</h2>-->  
<!-- <label class="badge badge-outline-success badge-pill"><a-->  
<!-- href="finished.php">View Lessons</a></label>-->  
<!-- </div>-->  
<!-- <div class="statistics-item">-->  
<!-- <p>-->  
<!-- <i class="icon-sm fas fa-chart-line mr-2"></i>-->  
<!-- Canceled Lessons-->  
<!-- </p>-->  
<!-- <h2>-->***<?php** *//echo $cancel\_ins* **?>***<!--</h2>-->  
<!-- <label class="badge badge-outline-success badge-pill"><a-->  
<!-- href="cancelled.php">View Lessons</a></label>-->  
<!---->  
<!-- </div>-->  
<!---->  
<!-- </div>-->  
<!-- </div>-->  
<!-- </div>-->  
<!-- </div>-->  
<!-- </div>-->  
<!-- -->***<?php***// }  
//* **?>** </**div**>  
 *<!-- content-wrapper ends -->  
 <!-- partial:partials/\_footer.html -->* **<?php  
 include**(**"footer.php"**);  
  
**?>**

Bibliography

[1]MengXF, Lu HJ, et al. Data extraction from the Web based on pre-defined schema. Journal of Computers Sciences and Technology, 2001,17(4):377~388.

[2]Silberschatz A, Stonebraker M, Ullman JD. Database systems: Achievements and opportunities. CACM, 1991,34(10):110~120.

[3]Silberschatz A, Stonebraker M, Ullman JD. Database research, achievements and opportunities

into the 21st century. SIGMOD Record, 1996,25(1):52~63.

[4]Silberschatz A, Zdonik SB. Strategic directions in database systems——Breaking out of the box. ACM Computing Surveys, 1996,28(4):764~778.

[5]Diao Y, Altinel M, Franklin MJ, Zhang H, Fischer P. Path sharing and predicate evaluation for high-performance XML filtering.TODS, 2003,28(4):296~336.

[6]Meng XF, Wang S, Wong KF. Overview of a Chinesenatural language interface to databases: Nchiql. IJCPOL, 2001,14(3):213~232.

[7]Meng XF, Liu S, Wang S. Word segmentation based on database semantic in Nchiql. Journal of Computer Science and Technology,2000,15(4):346~354.

[8]Zhang X, Meng XF, WangS. KingBase lite: A smart mobile embedded database system. In: Proc. of Fourth Int’l Conf. on HighPerformance Computing in Asia-Pacific Region(HPC Asia 2000). Volume II, IEEE Press, 2000. 806~811.

[9]Goldman R. Integrated query and search of databases,XML, and the Web [Ph.D Thesis]. Stanford University, 2000.

[10]Meng XF. Research on the technology of Web information integration. Computer Applications and Software, 2003,20(11):32~36(in Chinese with English abstract).

[11]Halverson A, Burger J, Galanis J. Mixed mode XML query processing. In: Freytag JC,Lockemann PC, Abiteboul S, Carey MJ,Selinger PG, Heuer A, eds. Proc. of the 29th Int’l Conf. on Very Large Data Bases (VLDB). Berlin: Morgan Kaufmann, 2003.225~236.

[12]Bobineau C, Bouganim L, Pucheral P, Valduriez P. PicoDBMS: Scaling down database techniques for the smartcard. In: Abbadi AE, Brodie ML, Chakravarthy S, Dayal U, Kamel N, Schlageter G, Whang K-Y, eds. Proc. of the 26th Int’l Conf. on Very LargeData Bases (VLDB). Morgan Kaufmann, 2000. 11~20.