A Cost Effective School Management System for Disadvantaged Schools in the Free State Province Using the Software as a Service (SaaS) delivery model

By

Ms Elizabeth B Kuriakose

Dissertation submitted in fulfilment of the requirements for the degree

Magister Technologiae: Information Technology

In the

School of Information Technology

Of the

Faculty of Engineering and Information Technology

Of the

Central University of Technology, Free State

Supervisor: Mr Gerald Maina Muriithi

March 2014

DECLARATION OF INDEPENDENT WORK

I, ELIZABETH B KURIAKOSE, hereby declare that this research project, submitted for the degree Magister Technologiae: Information Technology, is my own independent work and has not been submitted before to any institution by me or anyone else as part of any qualification.

……………………………………………… ………………………..

Student's signature Date

ACKNOWLEDGEMENTS

A dissertation is never the singular work of an individual; it is conceived, developed and presented only through the help of countless individuals.

I would like to take this opportunity to acknowledge and extend my heartfelt gratitude to the following persons who have made the completion of this dissertation possible:

Mr Gerald Maina Muriithi, my guide and my mentor, who has supervised my dissertation.

The Central University of Technology, Free State, and the Innovation Fund which has provided the financial and material support for my research.

The many programmers and developers who, through their many years of of programming, have helped me solve the hardest of errors and guided me throughout my project.

My brother and sister-in-law, Mr Rengith Baby Kuriakose and Mrs Reenu Rengith, for having to put up with me through the most difficult phase of my life and encouraging me to be a better person.

My parents, Mr Baby Kuriakose and Mrs Lissy Baby Kuriakose, for inspiring me to dream big and live my life to the fullest.

Last but not least, to almighty God for having blessed me abundantly all through my life.

ABSTRACT

The aim of this study was to create a dynamic software system that captures all information related to a student and delivers it to the educators, principal, higher authorities and parents. In order to achieve this aim, an investigation was launched as to the development of a cost-effective school management system for disadvantaged schools in the Free State Province using the Software as a Service (SaaS) delivery model. Although a variety of other school management systems exist in the market, they are often expensive and difficult to maintain.

Details such as previous academic performances, disciplinary actions taken against a student in the current school, ailments the student suffers from and parental details are some of the information that will help an educator to better understand a student. The software that is currently in use fails to deliver this information.

Designing the software as a multitenant system, helps accommodate different schools under the same database, while the shared database, shared schema reduces back-end costs. Database design was carried out in such a way that tenant data is logically isolated and that data integrity is maintained throughout.

What makes the software explained in this study cost effective is the method of delivery that was employed, which is SaaS. Here, software is not purchased, there is no upfront capital and the yearly license fee is eliminated, as schools need only pay a monthly rental fee for the services they use. Since all services are provided through the Internet, there is no need for system space; the only requirement is a high-speed Internet network.

TABLE OF CONTENTS

List of Figures

List of Abbreviations

Admin	–	Administrator
ADO.NET	–	Active Data Object
ASP	–	Asynchronous Server Page
CLR	–	Common Language Runtime
DLL	–	Dynamic Link Library
E-R	–	Entity Relationship
HTTP	–	Hyper Text Transfer Protocol
JIT	–	Just In Time
Mac OS	–	Macintosh Operating system
MSIL	–	Microsoft Intermediate Language
MS Visio	–	Microsoft Visio
n.d.	–	No date available
PHP	–	Personal Home Page
SaaS	–	Software as a Service
SLA	–	Service Level Agreement
SQL	–	Structured Query Language
URL	–	Uniform Resource Locater
IT	–	Information Technology

Chapter 1: Introduction

Students are managed by a group of teachers under the ministry of education to achieve a common goal. That goal is helping them achieve academic proficiency, and widen their horizons through a solid academic foundation, **but how can a modern technology such as Software as a Service (SaaS) aid in this?**

In the past, SaaS has been used in social networking sites and in large industries, but this study attempts to answer the question as to how such a technology can be brought down to grass-roots level.

To improve students' academic achievement, it is vital to understand not only the students' past academic record, but also where they come from, including their physical and emotional living conditions and the mental constraints they are facing. Furthermore, the higher authorities, principal, parents and teachers need to be aware of the constraints that other teachers are facing with regard to students. Overall, it is important that good communication exists between educators, students, parents and higher authorities.

The question can rightly be posed as to how all of this is possible in a time when teachers are bound to a curriculum that restricts them to a tough timetable and leaves little time for most teachers to discuss with peers the difficulties they face. If a virtual interaction medium could be set up, teachers need not meet each other physically. Suppose a web service could be set up in which the teachers, the principal, the parents and higher authorities could interact virtually, communication would be enhanced. For example, teachers could log into this service and find a detailed database of a student, containing the physical and mental constraints he/she faces, his/her past academic performance and so forth. Each teacher could explain in the forum provided by the service the difficulties they face, while the authorities and other teachers could offer possible solutions to these problems. It is crucial that the higher authorities are able to understand their employees and the many difficulties they face in the imparting of knowledge to the students. Parents can also be a part of this process: they can log into the website and keep abreast of their children's progress as well as of the sort of behaviour they exhibit in class.

Good teachers should aim to treat their students equally, neither showing nor giving one student preference over the other while, at the same time, paying equal attention to every student and imparting to them knowledge to the best of their ability. Furthermore, a good teacher is not one who treats all students equally, but understands a student's potential as well as their varied capabilities, and challenges him or her accordingly. In this way, the students are encouraged to reach the pinnacle of their abilities.

How can teachers understand their students? To answer this question, more detailed questions should be asked: What is the student's background, what is his/her family setup? Does he/she have parents or does he/she live with a guardian? Does he/she suffer from any illness or disorder which may affect his/her performance? If so, what are the possible treatments for this illness or disorder? What was the student's performance in the past in previous schools? These questions bring to the fore the importance of the parents' involvement in the daily progress of a student's academic life.

In light of the above, the aim of this study was to design a cost-effective school management system so as to understand the students and their background better, and to include the teachers, the principal and the parents in the academic development of a child by delivering a better perspective of the learners.

1.1 Problem Statement

Schools in South Africa use a system that is known as the South African School Administration and Management System[1] (SA-SAMS). SA-SAMS is a fairly comprehensive software that attempts to capture the basic information pertaining to the students, parents and staff members of that school. SA-SAMS also provides a financial wizard, timetable generator and other tools that assist in the management and administration of the school.

SA- SAMS was developed in 2005 and tested in several schools around South Africa. It was made compulsory in 2008 in all schools [1].

SA-SAMS records information such as the students studying in a class and their academic profiles, behavioural records, exam results and parent- guardian information.

Although SA-SAMS is comprehensive in its purpose of assisting a principal in managing a school by collecting relevant information pertaining to the members of the school and all the activities occurring in a school, it fails in delivering the information collected about a student to the educators and parents. Only the principal and higher authorities have access to the information recorded in SA-SAMS.

SA-SAMS provides patches or fixes for the bugs that crop up over time. These patches have to be downloaded and installed onto the system by each individual school, and this at times may prove to be a hassle for the busy schedule of a principal.

Currently, SA-SAMS is updated annually. Principals use the latest version for a year, report on possible errors and limitations, and wait for a year before the corrected and updated system is delivered in CD format to the school. If an additional feature is desired, it takes approximately one year for it to be fixed and made available.

Taking all of these limitations of SA-SAMS into consideration, this thesis addressed the design and development of a more cost effective cloud based School Management System targeting disadvantaged schools in the Free State province of South Africa.

The system is designed so that, above all, it is affordable and within the budget assigned to a school. The system becomes cost effective because it is hosted in the cloud. Hosting a system in the cloud makes it available to everyone in the virtual world while, at the same time, eliminating the cost of ownership of the system and the annual license fee associated with owning software. Users only pay for the features they use and there is less wastage of resources. ~~either~~

Another advantage of hosting software on the cloud is the timely update of the system. New features and patches for fixing bugs are ~~coded~~ directly added ~~on~~ to the system, eliminating lengthy waiting periods for upgrades common with on-premise approach employed by the current implementation of SA-SAMS.

As stated before, SA-SAMS also lacks a feature to foster communication between the main stakeholders such as teachers and parents. In view of this~~,~~ a discussion forum is developed in which teachers can post their challenges and seek views from their peers. In this way, they can ~~also~~ learn of the difficulties faced by other teachers and suggest possible solutions. By viewing and participating in the discussion, the principal comes to learn of the teachers' challenges, can offer solutions and can take preventative measures to avoid the same problems in the future.
Instead of calling meetings to announce and discuss upcoming events, an event notification system is developed. Discussion about these events can be held by the principal and educators in the discussion forum.

1.2 Research Objectives

In view of the above discussion, the following are the main objectives of this thesis:

- Investigate the existing school management system, and identify the main limitations.
- Design, develop and test a cloud based school management system that incorporates features missing from the current implementation of SA-SAMS.
- Evaluate the cloud based system and determine how well it meets the needs of the school community by carrying out usability tests for selected modules.

1.3 Organisation of dissertation

This dissertation consists of nine chapters.

The first chapter provided background information on the questions motivated this study, as well as a brief overview of the system that is to be developed.

Chapter 2 provides an overview of cloud computing and a set of related technologies such as multi-tenancy that will be used to design a cloud based solution.

Chapter 3 presents the research methodology that was employed in carrying out the work.

Chapter 4 analyses the requirements of all stakeholders including teachers, parents, learners, school principals and the ministry of education.

Chapter 5 presents the design of a shared multi-tenant database that caters for the needs of multiple schools.

Chapter 6 discusses the implementation of the system.

Chapter 7 presents the user interface design, i.e. how a user interacts with the system.

Chapter 8 presents a detailed step by step description of how the system was deployed to the cloud.

Chapter 9 presents the results of a usability test that was carried out to evaluate the viability of the system from the users's perspective.

Chapter 10 provides the conclusions drawn from the research as well as future recommendations for future research.

Chapter 2: Literature Review

2.1 Cloud computing

2.1.1 Definition

A widely accepted definition for cloud computing is the one proposed by the US **National Institute of Standards and Technology (NIST) [23]. NIST defines** cloud computing as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [23].

Amburst et.al [26] argue that with cloud computing, tenants no longer require the large capital outlays that are required to deploy their service or the human expense that is required to operate it. They need also not be bothered about the over provisioning of computing resources for a service that fails to take off or under provisioning for one that becomes widely popular thus missing important customers and revenue. In their view, cloud computing refers to services that are delivered through the internet and hardware and datacenters that provide those services [26].

According to the NIST definition, cloud computing is composed of five essential characteristics**,** three service models, and four deployment models [23]:

- It is sold on demand, usually on a minute or hourly basis.

- All these capabilities are available over a network and accessed through standard mechanisms. Clients access these facilities through different platforms such as mobile networks, laptops etc.
- The provider pools computing resources and makes them available to multiple clients. Using a multi-tenant model, different customers can be served with varying physical and virtual sources.
- Services can be scaled up or scaled down as per demand of the customer. Often customers are given a feeling of unlimited resources.
- Resources are often continuosly metered and monitored. Resource usage can be monitored, controlled and reported. This allows both provider and customer to have a transparent view of the provided resources.

The deployment of cloud computing services differs according to the number of users and the market that it targets. **Public clouds** are accessible to every user on the internet and anyone can access its services [23, 26] an example of public cloud is the Amazon Web Services [15] **Private clouds**, on the other hand, are applications that are developed exclusively for the customers in an organization [23, 26]. The **community cloud** serves customers from multiple organizations that have shared concerns. These clouds are usually owned and managed by one or more of the organization within the community [23]. When a person uses the resources of a public cloud to develop a private cloud, it is defined as a **hybrid cloud** [23]. Whatever the deployment method, the goal of cloud computing is to provide easy, scalable access to computing resources and IT services.

2.1.2 Cloud Service Types

Cloud computing offers IT companies and organisations ways and methods to optimise their IT capacity without adding infrastructure, personnel or software. Based on the requirements of the user, there are three service types available in a cloud computing model. They are as follows [23]:

2.1.2.1 Software as a service (SaaS)

SaaS is the delivery of software and its services through the internet [23,26]. Here a particular application is accessed by multiple tenants using a browser. It is mainly implemented in sales, Human Resources and Enterprise Resource Planning. A detailed explanation of SaaS will follow in the coming sections.

2.1.2.2 Platform as a service (PaaS)

In PaaS software and product development tools are hosted on the provider's infrastructure. Developers create applications on the provider's platform through the Internet. PaaS providers may use application program interfaces (APIs), website portals or gateway software installed on the customer's computer [23]. Examples of PaaS providers are Force.com and GoogleApps.

2.1.2.3 Infrastructure as a Service (IaaS)

IaaS allows users to create and access virtual data storage centres on the infrastructure provided. Infrastructure may be anything from storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software. Here users can increase their storage space as demand increases or decreases. This follows a pay-for-what-you-use model that is similar to paying for electricity or water since you only pay for what you use [23].

2.2 Software as a Service (SaaS)

2.2.1 Definition

SaaS has been defined as the newest way to market software. SaaS gives a person the right to use particular software running on a provider's cloud interface [23]. As long as one utilises only the necessary features, one pays only for what one uses. When one uses software through SaaS, one also purchases a hosting and infrastructure website along with the rights to use the software. SaaS providers maintain the hardware, perform upgrades, backup one's data (sometimes), and otherwise perform all of the services and activities required to keep the software running. SaaS has redefined the software deployment model. Earlier upfront capital and yearly license fees would have been required to purchase software, afterwards it would have to go through a lengthy implementation phase. SaaS cuts out this processes, by implementing a pay-as-you-go internet delivered service relationship [29].
SaaS maybe considered as a form of software evolution.

Figure 1 illustrates the early implementation of SaaS by companies and how SaaS architecture is also evolving to give way to newer architecture.

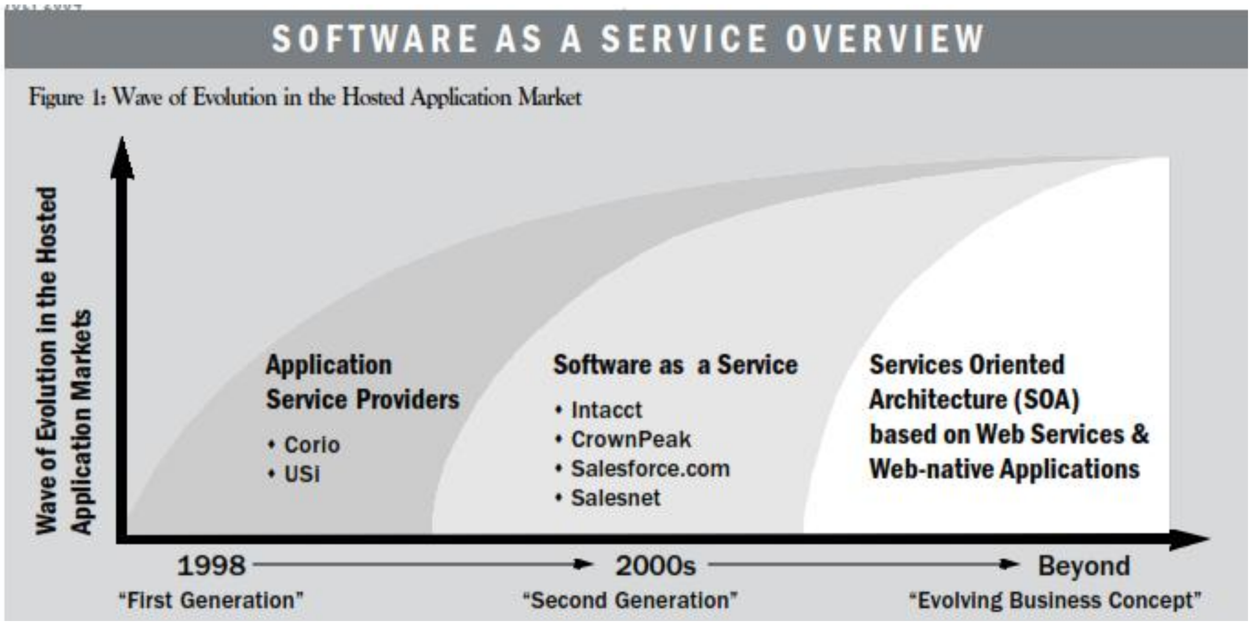

SOFTWARE AS A SERVICE OVERVIEW
Figure 1: Wave of Evolution in the Hosted Application Market
Wave of Evolution in the Hosted Application Markets
Application Service Providers
• Corio
• USi
Software as a Service
• Intacct
• CrownPeak
• Salesforce.com
• Salesnet
Services Oriented Architecture (SOA) based on Web Services & Web-native Applications
1998
2000s
Beyond
"First Generation"
"Second Generation"
"Evolving Business Concept"

Figure 1 Software as a service overview [29]

To gain better understanding of SaaS, a brief overview of the software that existed in the 1990s is needed [4]. In the 1990s a source code control system had to be purchased, after which a server had to be set up in order to install the software. Data that run in these servers ran a high risk of being lost: a botched upgrade or hardware failure could lead to loss of data. Costs associated with the design were borne by the user. Depending on functionality, a decision had to be made whether or not this software was to be accessed by others in the network, designing the system in such a way that it is secure from competitors. In this way, big design overheads were incurred. Finally, the cost of maintaining these servers was quite high. Everything from the high electricity consumed by the servers to the technology used to cool and maintain the servers had to be taken into consideration.

If one were to outsource most of these activities to an IT services firm, it would manage the hardware and the software, including the security model. All that would be left would be to use the software. This is exactly how a SaaS application works.

2.2.2 Purchasing software

When referring to the purchase of software, there is a misunderstanding that software is bought and the buyer has complete ownership. In reality, a licensed copy of the software is leased to the person who purchases the software and the person is granted usage for a set time period, after which periodic renewals must be carried out by the purchaser. With each upgrade the purchaser is usually required to purchase the upgraded software as well [4], [28].

2.2.3 Economics of software purchase

Purchasing software, in fact, means that you first paying for a particular version of that software, and afterwards paying for upgrades of that software. For example, a new version of MS Word is released every few years. Upgrades actually cost less than purchasing the software.
When companies sell licenses for software, they make updates to the software annually or even daily Companies usually release two types of updates, namely minor updates and major updates

2.2.3.1 Minor updates

Minor updates are usually free and often include bug fixes or features that were intended to be in the major release, but were delayed, or these updates might simply be the introduction of capabilities with ‘lessel value to customers. A great number of software will automatically notify the user, download the update, and install it [4].

2.2.3.2 Major updates

Major updates usually require the purchase of an upgrade. Major updates are usually more significant; they introduce capabilities that have ‘higher’ value to their customers or are intended to make the product appealing to additional markets. They are released only a few years after the main product has been released [4].

2.2.4 Understanding economics of software purchases

Assuming that, for software, the minor updates are provided regularly and install themselves into the system, whereas major updates have to be purchased. Furthermore, with each update, something new and valuable has been introduced to the consumer. If a graph were to be plotted for value against time, the minor updates would increase the value of the software marginally and the major updates would display a large increase in the curve. This is demonstrated by figure 1 below [4].

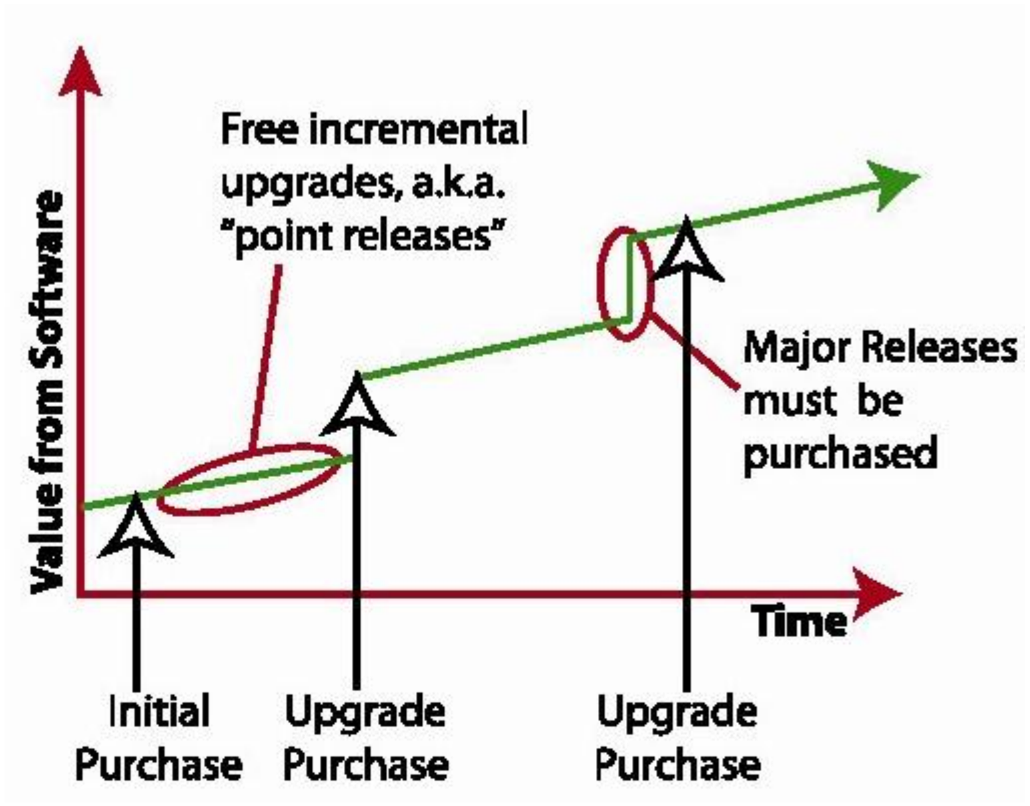

Value from Software
Free incremental upgrades, a.k.a. "point releases"
Major Releases must be purchased
Time
Initial Purchase
Upgrade Purchase
Upgrade Purchase

Figure 2 Increase in software value with each update[4]

As each new customer buys the newly released software, the company obtains more revenue. Meanwhile the existing customer buys an upgrade for the existing software, generating more income for the company selling the software. In this way, a company profits from both finding new customers and keeping existing customers. Until a large customer base has been established for a company, the primary focus of a company is finding new customers for the product. Based on economics principles, satisfaction of the existing customer threatens to become a secondary priority. Even though this is a vague picture, it nonetheless has an influence on the software market.

The green area of the figure 2 indicates the value of the purchases of these timely upgrades to the customer. Value is often a function of how much the software is used. It can also be noted from figure 2 that the value of a major upgrade can be experienced only once the software has been purchased. A customer can benefit from the incremental value of the minor releases of this upgrade only after the upgrade has been made.

2.2.5 Software Costs

In a typical organization, the IT budget is allocated for three areas namely: software, hardware and people services [28, 30]. Of the three, software is directly involved in computing and information processing. Ideally that would the goal of IT organization, whereas hardware and people services are required to help manage the software efficiently. Although hardware and people services are vital, any organization if given the opportunity would add software functionality without the extra hardware, but not include extra hardware if not required by the software [28, 30]. Chong and Carraro [28] state that in an IT based environment, majority of the budget is spent on hardware and people services, leaving only a minority to be spent on software.

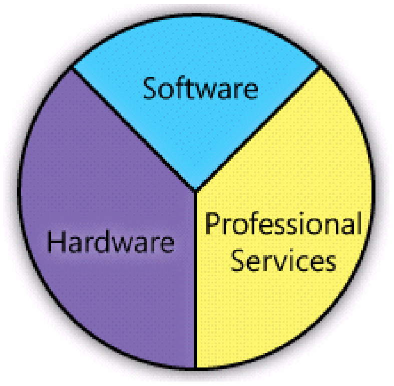


Figure 3: Typical budget for an on-premise based software [28]

In the model illustrated above (Figure 3) the hardware budget is used for purchasing desktops and mobile computers for end users, servers to host data and applications and components to network them together.

Apart from hosting of software there are license costs that are associated with the purchase of software and its timely upgrades that can be represented in the step chart represented in figure 4. The assessment is set out in the form of a step chart to indicate that, once software has been purchased, it does not go up again until its respective upgrade or update has been purchased. There are several costs relating to the ownership of software.

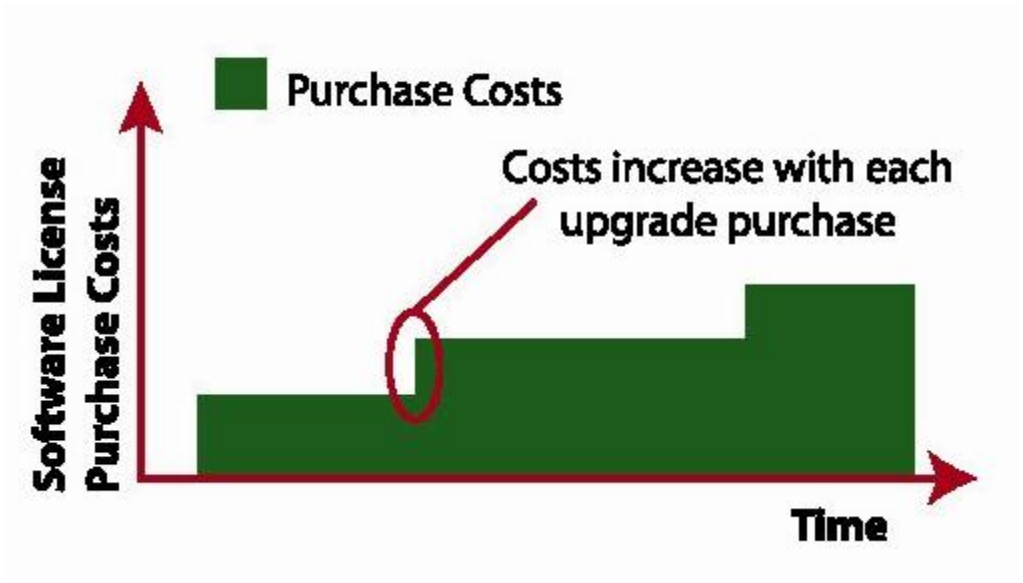


Figure 4 Licensing cost as time progresses [4]

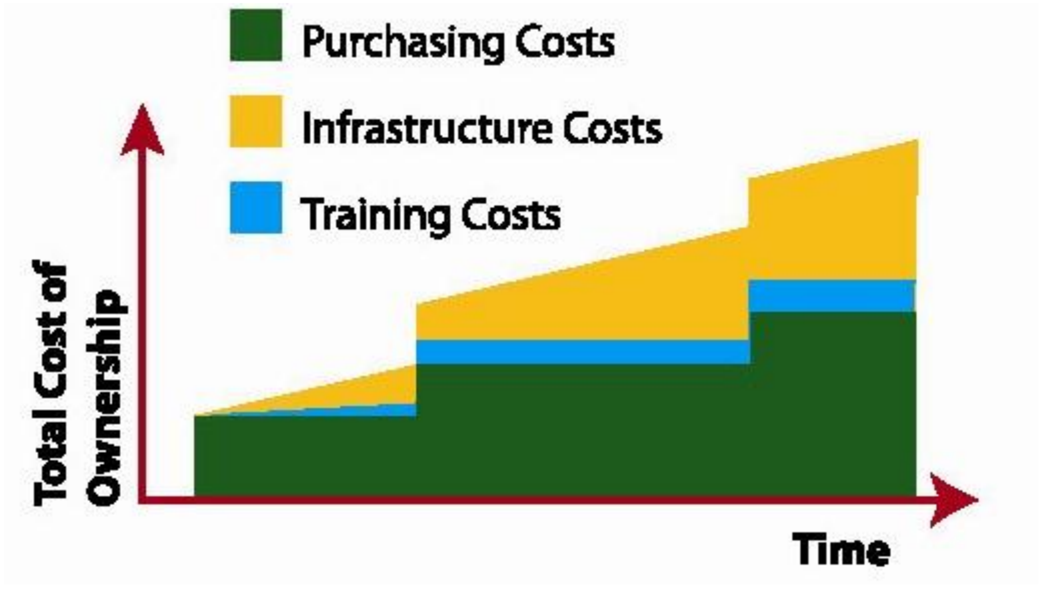


Figure 5 Cost of ownership as time progresses [4]

The purchase of ownership of software is an ever-increasing expense. With different software this cost will increase or decrease respectively. It is thus clear that the cost of software does not end with its purchase.

2.3 Economics of SaaS

The purchasing of SaaS may be compared to magazine or satellite television subscription: you pay as long as you are subscribed to the service. The major and minor upgrades are included in the service automatically, but the more services you require, the greater the fee. In short, it may be described as paying for what you use. When one wants to stop using the product, one simply stops paying the fee.

The model for creating value with SaaS products is the same as that with licensed software as illustrated in figure 5, the perspective of the customer as shown in figure 6 below.

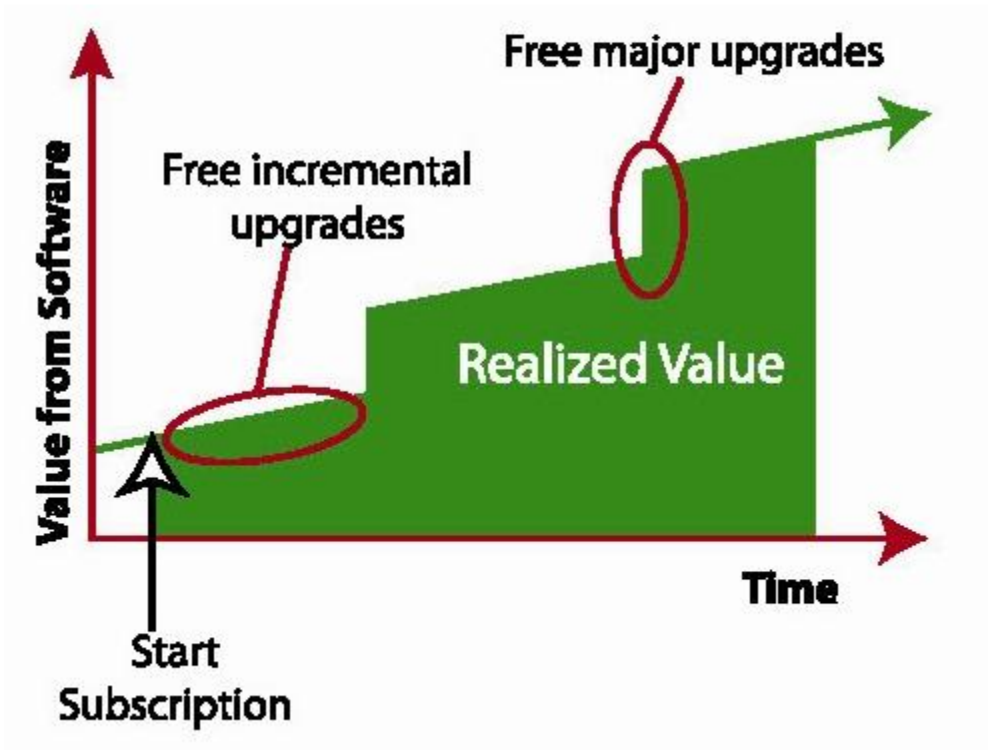


Figure 6 Value of SaaS as time progresses [4]

2.3.1 The cost associated with SaaS

In an organization that relies on SaaS the budget allocation model is quite different as illustrated in figure 7.

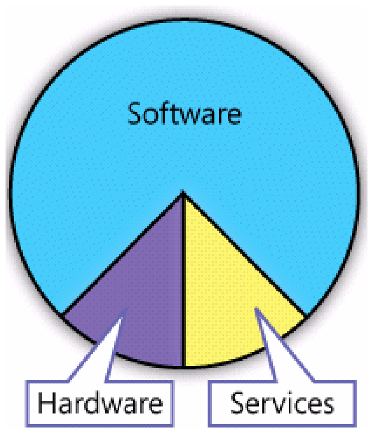


Figure 7: Typical budget for a SaaS environment [30]

In this model the SaaS vendor maintatins all the associated data and critical applications on his servers. The vendor supports this hardware and software with a dedicated support staff, hence relieving the customer any need to support the software. Application delivered through the internet also significantly help place a lesser demand on the desktop computer hence allowing the desktop a longer life [28, 30].

Figure 8 is another illustration of the cost of SaaS as time progresses.

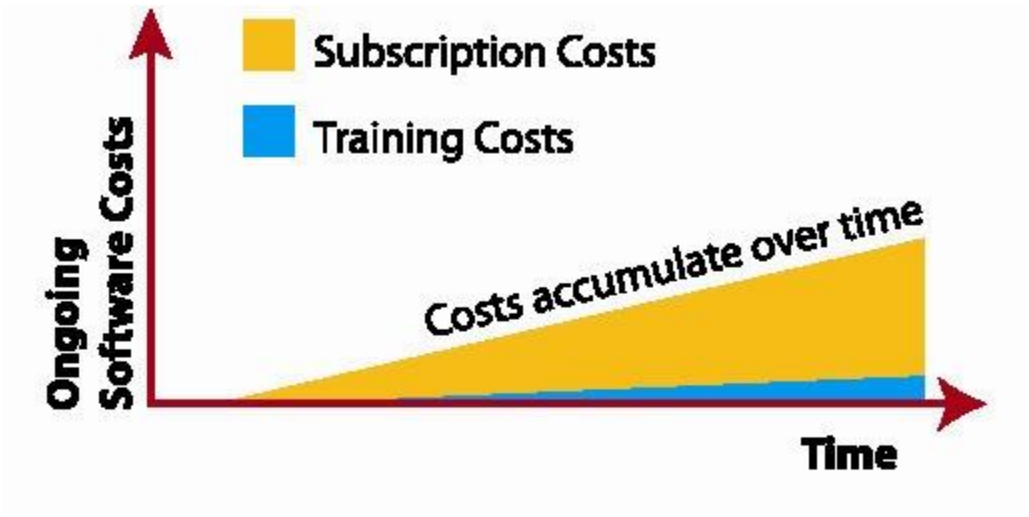


Figure 8 the cost of a using SaaS as time progresses [4]

2.4 Comparison of independent software vendors and software as a service

Both models involve costs that increase over time. For several technical reasons, the SaaS architecture is more efficient and has lower costs for the software company, which tends to result in lower costs for the customers. This is not always the end result, but is usually the case.

Another factor is the financial pressure on the SaaS provider. Where a software licensing model creates pressure to prioritise finding new customers, a SaaS model creates pressure to keep existing customers [4, 29, 30].

SaaS providers get the same revenue from a new customer as from an existing customer, as opposed to the ‘new vs. upgrade’ dynamic seen with software licensing models. In most cases, it is cheaper to keep an existing customer than to find a new one. Consequently, financial pressure mounts to retain existing customers. This pressure can drive a different behaviour, more like that of a retail sales model, where keeping your existing customers is critical [4].

The SaaS model ultimately provides the same type of products as a software licensing model, but with a better economic model, one that is lower in cost to the customer and structurally inclined to keep improving for the customer with every new release [4, 29, 30].

Personally, one likes the idea of purchasing from a company that is financially motivated to keep its customers happy, not one that is pressured to find another customer as soon as existing ones write a cheque [4].

The best companies try to reinvent themselves and improve their products continuously. Over time, the best companies will move to SaaS models, which align their financials with their objectives [4].

2.5 Examples Areas of application for SaaS

SaaS finds applications in a number of areas, of which some are described below [17].

2.5.1 Business intelligence

Business intelligence (BI) is a relatively new category within SaaS, and numerous vendors have appeared on the market. These include such established providers as System Application Product (through Business Objects), but many of the vendors are smaller companies. Among the companies that are offering BI SaaS are Adaptive Planning, Birst, Blink Logic, SAP/Business Objects, Cloud9 Analytics, Good Data, Host Analytics, Kognitio, LucidEra, Oco, PivotLink and Telemetree.

2.5.2 Collaboration

Web 2.0 is popular and many organisations are obtaining their collaboration applications via SaaS. Technologies used for collaboration include everything from email and instant messaging to wikis and blogs. Among the leading vendors in team collaboration software, are Designlinks International, IntraLinks, Grove Technologies, Huddle/Ninian Solutions, Jive Software and TeamSpace. For web conferencing, vendors include Adobe, Cisco, Citrix, IBM, InterCall and Microsoft.

2.5.3 Enterprise content management (ECM)

ECM technology is being adopted by more organisations in their attempt to search for ways to reduce paperwork and improve workflow. Companies deploy ECM to create, manage and distribute a variety of content types. Among the vendors in the SaaS ECM market are EMC, IntraLinks, Microsoft and SpringCM.

2.5.4 Enterprise resource planning (ERP)

One of the trends in the ERP market is the emergence of SaaS delivery of these enterprise applications used to manage organisations' back-office processes such as human resources, accounting, and order management. SaaS ERP offers a pay-as-you-go alternative to packaged applications or custom-built ERP systems. The vendors offering SaaS ERP include ADP, Coda, Glovia, Intacct, NetSuite and Applicor.

2.5.4.1 Supply chain management (SCM)

Applications that help organisations run their supply chains more efficiently have not yet reached the stage of notable SaaS adoption [19]. Within the SCM category certain areas such as supplier relationship management, global trade management and supply chain event management have started to show movement towards adopting SaaS. Among the vendors addressing this market are Aravo, GT Nexus and Sterling Commerce.

2.6 Advantages of SaaS

One of the main advantages of SaaS is easy administration, automatic updates and patch management that helps fix bugs, while compatibility ensures that all users have the same version of software. Delivering software through SaaS ensures global accessibility and collaboration [21].

2.7 Disadvantages of SaaS

Some of the main disadvantages of using SaaS are, firstly, that organisations find it very difficult to relinquish control or trust third parties to manage their applications and data. Secondly, some vertical markets require industry-specific business applications for which SaaS solutions are not available, as SaaS is incompatible with some business models. Finally, organisations without clear objectives and defined business processes will be no better off with a SaaS solution than with an on-premise solution [21].

2.8 Multitenancy

2.8.1 Definition

SaaS finds difficulty in adaptation mainly because of trust or a lack of trust [6]. Data, as the most important asset of any business, have to be managed safely and responsibly. Data is the heart of SaaS; SaaS applications provide customers with centralised, network-based access to data with less overhead than is possible when using a locally installed application. However, in order to take advantage of the benefits of SaaS, an organisation must surrender a level of control over its own data, trusting the SaaS vendor to keep it safe. The SaaS architect has the added responsibility of creating an architecture that is robust and secure and can deliver data efficiently to the client.

Multitenancy is a software architecture principle where a single instance of the software runs on a server, serving multiple client organisations (tenants). Where as in multi-instance architecture, separate software instances (or hardware systems) are set up for different client organisations. In multitenant architecture, a software application is designed to virtually partition its data and configuration, and each client organisation works with a customised virtual application instance [7].

In simpler words, when using a multitenant architecture, the same copy of software is used by different clients using their own data. Data architecture is an area in which the optimal degree of isolation for a SaaS application can vary significantly depending on technical and business considerations. Experienced data architects are used to considering a broad spectrum of choices when designing architecture to meet a specific set of challenges, and SaaS is certainly no exception.

In the following sections a detailed examination of three broad approaches, each of which lies at a different location in the continuum between isolation and sharing will be carried out [6]. Not all organisations require maximum security for the isolation of their data, nor do they require a backup of all existing information on server. Based on different financial and security requirements, the following models are available.

2.8.2 Different types of Multi-tenant Architecture approaches

There are three different database approaches to implementing multitenant architecture:

- Separate database approach
- Shared database, separate schema approach
- Shared database, shared schema approach

2.8.2.1 Separate database approach

In the separate database approach application code and computing resources are shared among users, but each tenant has logically isolated data. This means that data with respect to a particular client is stored in a separate database from other clients. This measure ensures that information is in no way copied or mismatched, ensuring privacy of data [6], [28].

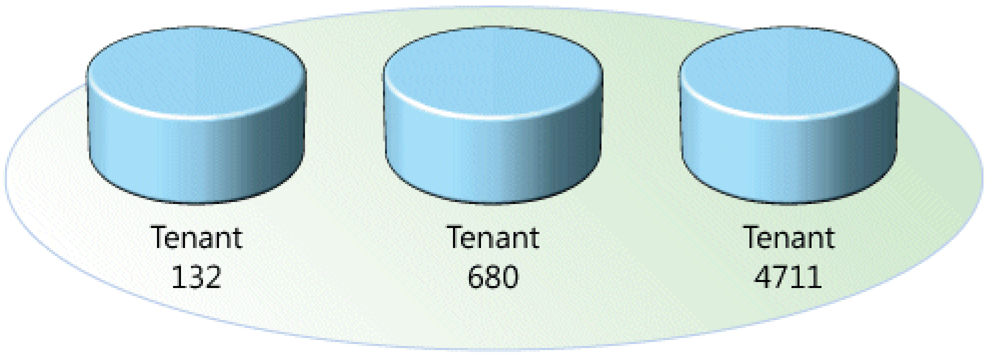


Figure 9 Seperate database approach [6]

2.8.2.1.2 Advantages of the separate database approach

The separate database approach ensures that data is completely isolated, and any mixing of data pertaining to two different users is eliminated. Since data is completely isolated it is easier to make backups and easier to restore after a fault [6].

2.8.2.1.3 Disadvantages of the separate database approach

The separate database approach incurs higher hardware and maintenance costs, which makes it unsuitable for organizations with a fewer number of clients. The separate database approach emphasises security; however, many organisations do not require that much security in their day-to-day functioning [6].

2.8.2.2 Shared database, separate schema approach

Here, each tenant shares the same database, but have different tables according to their needs. When grouped, they form a schema for that particular tenant.

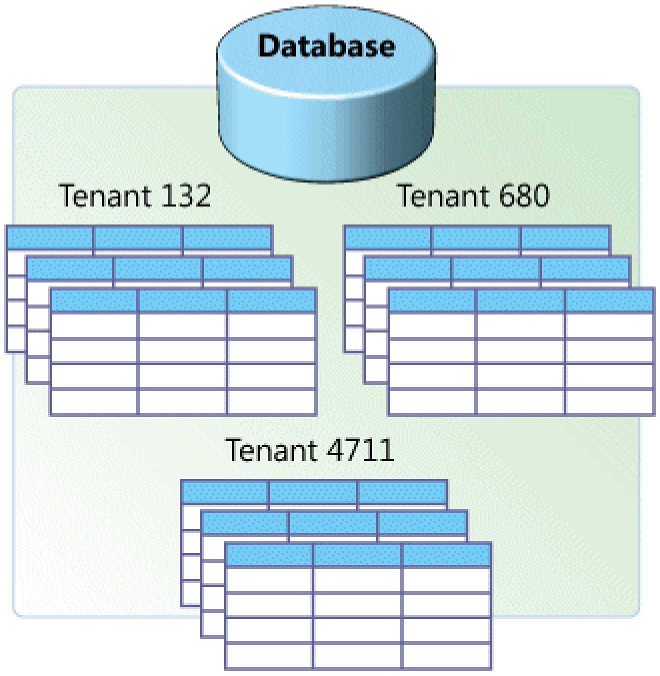


Figure 10 Shared database, separate schema approach [6]

Like the separate database approach, the separate schema method can be implemented easily and tenants can extend their data model with ease [6], [28].

2.8.2.2.1 Advantages of the shared database, separate schema approach

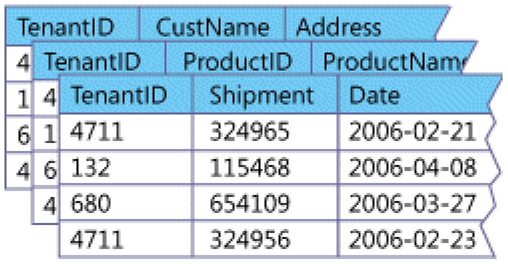
The shared database, separate schema approach is suitable for organisations that have 100 tables per user or less; hence, it requires lesser hardware and maintenance cost. This approach can typically accommodate more tenants per server than the separate database approach can [6].

2.8.2.2.2 Disadvantages of the shared database, separate schema approach

Some of the disadvantages of the shared database, separate schema approach is, firstly, that it is of lesser security. Secondly, clients will have to share information on the same database, which some may find uncomfortable [6].

2.8.2.3 Shared database, shared schema approach

The shared database, shared schema approach has the lowest hardware and maintenance cost [6]. It also allows the largest number of tenants per database. Because of the larger tenant base, additional design overhead will have to be incurred to ensure that the data of clients do not get mixed up. In this approach, different clients use the same database and same tables to hold their data. A tenant ID column associates every record with the appropriate tenan [6], [28].



TenantID	CustName	Address
4		
1		
6		
4		

TenantID	ProductID	ProductNam
4		
1		
6		
4		

TenantID	Shipment	Date
4711	324965	2006-02-21
132	115468	2006-04-08
680	654109	2006-03-27
4711	324956	2006-02-23

Figure 11 Shared database, separate schema approach [6]

2.8.2.3.1 Advantages of the shared database, shared schema approach

The shared database, shared schema method accommodates more tenants than any other method. Of all the methods, it has the lowest of hardware and maintenance cost.

2.8.2.3.2 Disadvantages of the shared database, shared schema approach

This approach is the least secure among the three multitenancy models. During software bugs or errors, a single row wipeout can affect thousands of tenants. Additional design overhead must be incurred in order to increase security.

2.9 Choosing a Multi-tenant Architecture

Choosing an approach must be based on the following considerations:

2.9.1 Economic considerations

Those clients who can manage a higher expenditure can look to utilise the separate database approach. This allows maximum isolation of their data but, at the same time, includes higher maintenance and hardware cost. The shared database, separate schema approach incurs lower costs than that of the separate database approach. The former approach is for those who have limited funds, but want the maximum security they can get for their money. The least expensive and most accommodating of the three approaches is the shared database, shared schema approach which, through sharing, decreases cost greatly. [6]

2.9.2 Security considerations

The most secure of the three approaches is the separate database approach, which isolates the data of each tenant. Every other approach, through their sharing of data, provides only a lesser security [6], [28].

2.9.3 Tenant considerations

If a large number of tenants are required to be housed, the best approach would be the shared database, shared schema. Through the design of the shared database, shared schema architecture, thousands of tenants can be accommodated. However, if a tenant requires large amounts of storage space, the best approach would be the separate database approach. Depending on how many more concurrent end users the average tenant is required to support, the more appropriate a more isolated approach will be to meet end-user requirements [6]. If per-tenant value-added services are expected, such as per-tenant backup and restore capabilites a more isolated approach is a better option [6, 28].

2.9.4 Regulatory consideration

Companies, organisations and governments are often subject to regulatory law that can affect their security and record storage needs. After Investigation of the regulatory environments that a prospective customers might occupy in the markets in which it is desired to operate, it must be determined whether or not they present any considerations that will affect the decision making process. [6]

2.9.5 Skill set considerations

Chong et.al [28] states that “Designing single-instance, multitenant architecture is still a very new skill, so subject matter expertise can be hard to come by. If architects and support staff do not have a great deal of experience building SaaS applications, they will need to acquire the necessary knowledge, or people who already have the knowledge will have to be hired [6]. In some cases, a more isolated approach may allow staff to leverage more of their existing knowledge of traditional software development than a more shared approach would”.

2.10 E-learning systems that have been implement in cloud

Many countries around the world have already opted for a school management system that has been implemented using cloud computing. Some of these countries are Saudi Arabia, Canada, England and the United States of America [31].

Fujitsu is a Japanese company that specializes in providing IT resources to customers. In 1998 Fujitsu started developing an intranet based E-learning application known as internet navigware. Internet navigware is a training management application that allows various functionalities such as development of teaching materials to learning and results management [32]. By 2008 Fujitsu had started the conversion of internet navigware to a SaaS based application. Sakamoto [32] justifies the decision to migrate to SaaS by three reasons:

1. **Customer preference**

   Many customers prefer an application delivered over the internet to one that is intranet based. Customers need not own software and companies need only provide a software and its needed function when required, in this case internet navigware needs to be delivered only during training time.

2. **Stable source of income for company**

   As far as a company is concerned, once a SaaS customer ~~ased~~ has been registered there is a stable source of income. Monthly sales can be generated from metered usage.

3. **Consolidation of resources**

   From a developers perspective when developing an application they have to ensure that the system is compatible on all types of Operating systems. At the same time when developing a SaaS application they only create a copy of software that is delivered to all customers. Thereby deploying a system through SaaS reduces man hours spent on development and testing.

2.11 Summary

The proposed system will adopt the SaaS delivery method, the following factors need to be taken into account:

- The organisation for which the software is to be developed is a school.
- A school has limited funds available to spend on software.
- An average public school in South Africa has around 1000 users.

Based on these observations the **shared database, separate schema approach** has been selected as the **multi-tenant data base architecture** for the proposed system.

Chapter 3 Reseach Methodology

As discussed earlier, the work proposed in this thesis involves the design and evaluation of an enhanced version of SAMs that will run on the cloud. A Design Science Research Methodology (DSRM) is proposed. Hevner [25] argues that Design Science Research involves the design and implementation of important technology based solutions for unsolved and important problems. Hevner further argues that such solutions are implemented with the purpose of improving the overall efficiency of an organization.

This chapter provides an overview of the DSRM and discusses how this approach was applied in carrying out the work.

3.1 Design Science

Hevner et.al [25] defines design science "as the other side of the Information system research cycle which creates and evaluates IT artifacts intended to solve identified organizational problems." [25]. According to Hevner et al [25], DSRM is composed of six stages as explained below:

1. Problem identification and motivation
2. Definition of objectives
3. Development
4. Demonstration
5. Evaluation
6. Communication

3.1.1 Problem identification and Motivation

In this stage, the problem or limitation is clearly identified and a justification for the solution provided. By justifying the solution, it motivates the researcher and audience to accept the solution while helping the audience to understand the reasoning associated with the researcher’s understanding of the problem [26].

The current software employed in schools across South Africa is the south african student administration and management software (SA-SAMS). SA-SAMS only assists in the administration of schools, which means information collected by SA-SAMS is only accessible to the higher authorities and the principal. SA-SAMS also sufferes from the flaws that many software systems suffer from, such as only yearly updation of features, the software patches for existing bugs have to be downloaded and installed to be used. Much of the information collected by SA-SAMS would be useful to an educator, who has no prior knowledge about a student. The present academic progress of a student would be helpful to a parent, instead of finding out the Childs progress during parent teacher meetings and while collecting report cards at the end of a term.

As explained before, the current deployment approach adopted for SA-SAMs is not cost-effective because each school needs to individually install and maintain a copy on-premise. On the one hand, this duplication leads to higher implementation costs and lengthy upgrade cycles, sometimes stretching to a year. In addition, SAMs fails to address the needs of all stakeholders in the school community such as parents and teachers. A cloud based solution, where one centralized implementation of SAMs serves multiple schools may address these limitations in a much more cost effective way.

3.1.2 Definition of objectives.

Objectives maybe specified as either qualitative or quantitative. Qualitative objectives are solutions to limitations that have not been addressed yet, whereas quantitative objectives are desirable solutions to problems that exist and why they will be better than existing solutions [26]. In the work proposed here, a detailed assessment of the current implementation of SAMs was carried out based on visits to selected schools in the Free state province of South Africa. Since all schools use the same system, the visits were limited to two representative schools: Lere-la-Thuto High School and Samuel Johnson Secondary School, Xhariep, Free State. They helped as a basis for defining the problem and finding a solution which can be extended to other educational institutions facing the similar problems.

3.1.3 Design and Development

This step involves the actual creation of the artifact; this may involve creating a model or making an actual construct. It may also refer to the methods or instantiations that can be used to create the artifact. Design involves determining the functionality and architecture of an artifact and finally its creation [26].

At this stage, a multi-tenant, SaaS-ready database that addresses the data needs of multiple schools is designed and developed. The data model has to address the security, performance and scalability needs of the multiple tenants.

In addition, a sharable, SaaS-ready application was designed which securely interacts with the multi-tenant database designed above and serves the individual needs of each school. Although the schools share the same database and application code base, strong data isolation techniques were used to allow schools to access only the data that belong to them. Application interfaces also had to be customised to automatically reflect the individual school profiles.

Software tools were selected, namely servers, database management system and application development tools, which are suitable for developing SaaS applications. The system was coded and implemented as per the design specifications (derived from user needs assessment).

3.1.4 Demonstration

By demonstrating the created artifact, one should be able to prove that it conforms to functionality and does one or more instances of the mentioned limitations. One of the main resources for demonstration is the knowledge of using the artifact to solve the problem [26].

Testing of the software was carried out to ensure that the software was functioning according to specification and requirements. Each unit was tested to ensure that it was free from errors. The next step to do is an integration testing to make sure that the different units in the software function together without errors. The final check by the designer is to do a system testing, where all the components are brought together to evaluate whether they work together or not.

3.1.5 Evaluation

Evaluation of an artifact refers to the observation of an artifact to measure how well it supports the solution of the problem. Evaluation may take different forms depending on the problem and the artifact developed. Evaluation may be carried out by conducting user satisfaction surveys, client feedback or simulation. It may also be carried out more quantitatively by measuring system performance such as response time or availability [26].

The system was deployed on Internet Cloud Servers, in this case Ms Azure. Schools were given access through a web portal. The system was evaluated to gauge user satisfaction levels by means of a usability test that assessed how well the users perceived the performance of key modules of the system. Users recorded their perceptions on a questionnaire.

3.1.6 Communication

Communication is the process by which the problem and the artifact created to counter the problem is communicated to the world. A problem and the artifact created to counter it may be relayed to the world through articles in journals, paper presentations and thesises [26].
A journal paper reporting on the work carried out in this dissertation is being reviewed for publication by the Interim journal. In addition, a presentation on the system was made at the annual research seminar held at the Central University of Technology.

Figure 12 is a flow chart representation of the steps that went into the design and deployment of the system.

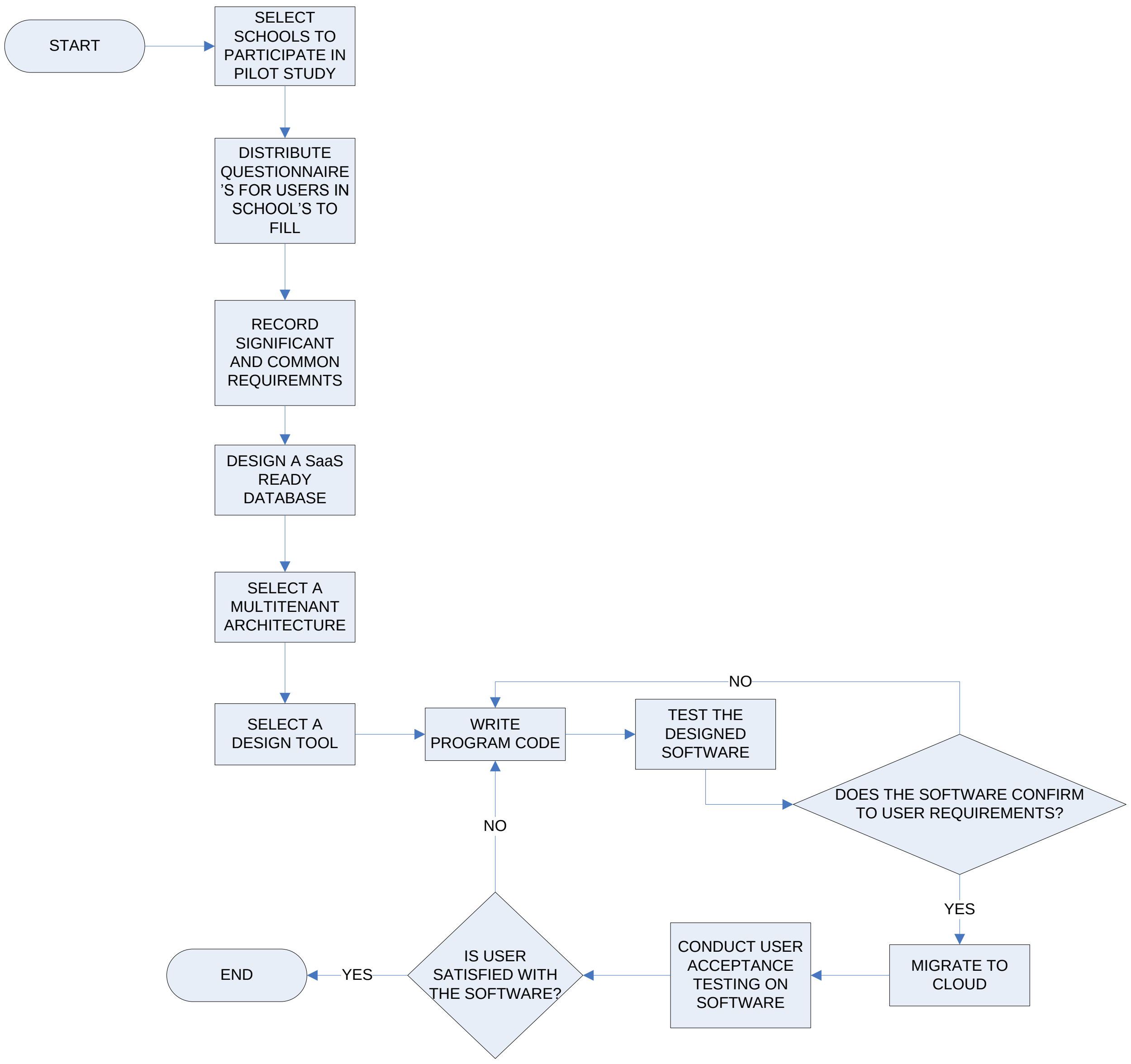

START
SELECT SCHOOLS TO PARTICIPATE IN PILOT STUDY
DISTRIBUTE QUESTIONNAIRE'S FOR USERS IN SCHOOL'S TO FILL
RECORD SIGNIFICANT AND COMMON REQUIREMNTS
DESIGN A SaaS READY DATABASE
SELECT A MULTITENANT ARCHITECTURE
SELECT A DESIGN TOOL
WRITE PROGRAM CODE
TEST THE DESIGNED SOFTWARE
NO
DOES THE SOFTWARE CONFIRM TO USER REQUIREMENTS?
YES
MIGRATE TO CLOUD
CONDUCT USER ACCEPTANCE TESTING ON SOFTWARE
IS USER SATISFIED WITH THE SOFTWARE?
NO
YES
END

Figure 12 Research Process

Chapter 4: User Requirements

This step relates to the the first two stages of the DSRM described above.

4.1 User requirements

The school management system is a multi-user system. The users of the system and their requirements are described in the following sections.

4.1.1 Principal

The principal's job is one that carries a lot of responsibility, as he makes a lot of decisions regarding the staff members, students and school as a whole. The school management system aims to help a principal manage the school more effectively. A principal should have access to all relevant information regarding each member under his/her management, be it the professional information regarding educators, medical details, academic records, and personal information of students, and professional and personal information of the students' parents and guardians.

The purpose of providing principals with the above information is for them to gain deeper understanding and be better able to assess the students and educators. By assessing students' past and present academic records, principals are able to evaluate their progress more efficiently.

The principal and authorised officials are the only people who have authority to **start discussions in the forums** provided in the school management system, i.e. only they can create topics in the forum for posting questions and holding discussions. Members who post content in the discussion forum have to wait for the principal’s approval of their content before it can be published. Using the discussion forum, virtual staff meetings can be held. The principal also has the authority to add a disciplinary record to a student and suggest the best disciplinary action to be taken against the student.

A principal using the notification system can notify the users in the school of upcoming activities and events. Educators can also notify the principal of the disciplinary issues that students have, which the principal can then review and offer suitable punitive actions to.

For a better understanding, each of the principal’s privileges regarding the information of a student, parent and educator is illustrated separately.

4.1.1.2 Principal flow chart with respect to school activities

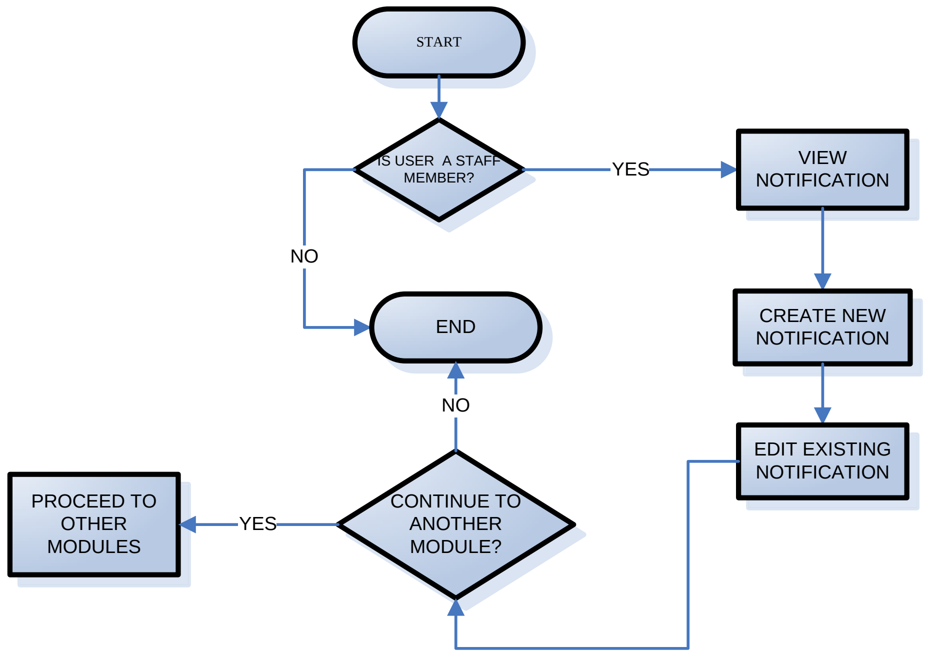


Figure 13 Notification module and how principal enters information

In figure 13, the user enters the system. If verified as the principal, the user may proceed to view, create or edit notifications. If not, he/she may proceed to another module or exit the system.

4.1.1.3 Viewing staff details

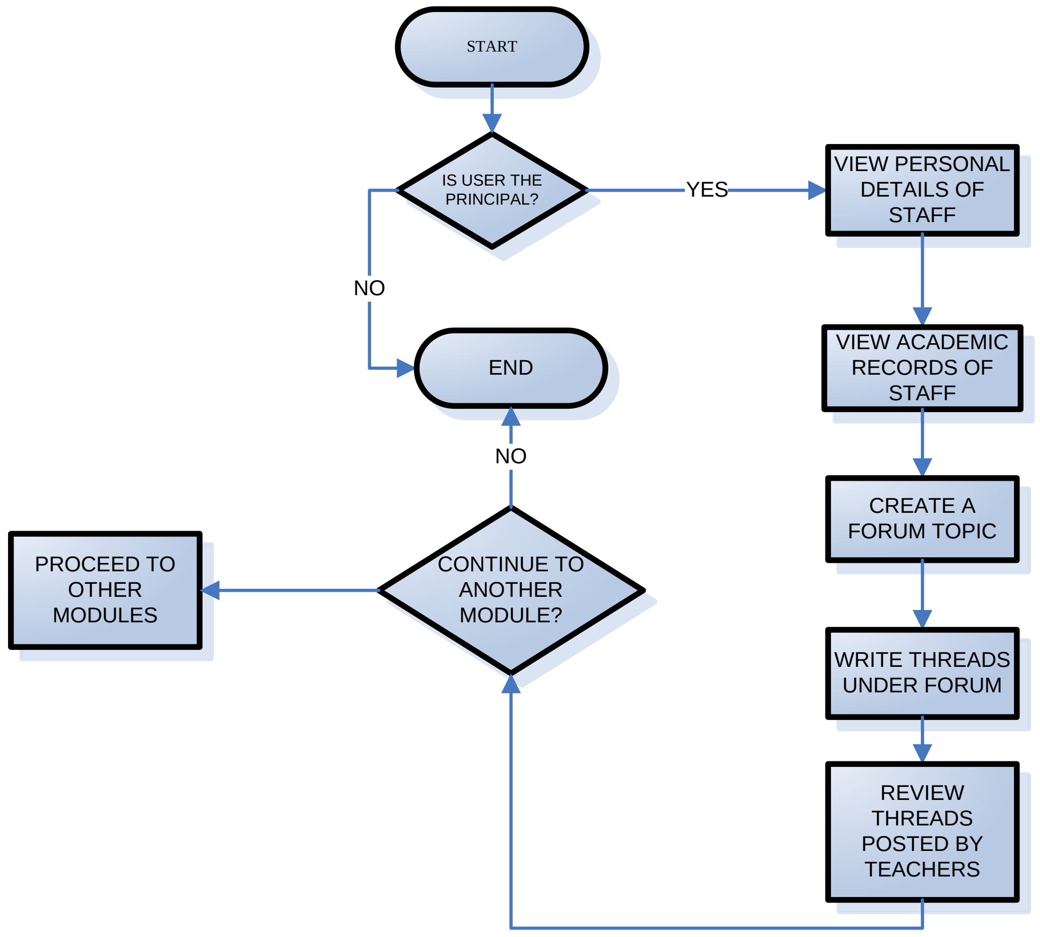

START
IS USER THE PRINCIPAL?
YES
VIEW PERSONAL DETAILS OF STAFF
NO
END
VIEW ACADEMIC RECORDS OF STAFF
NO
CREATE A FORUM TOPIC
PROCEED TO OTHER MODULES
CONTINUE TO ANOTHER MODULE?
WRITE THREADS UNDER FORUM
REVIEW THREADS POSTED BY TEACHERS

Figure 14 Viewing of staff information by principal

In the figure 14, the user enters the system. If verified as the principal, the user may proceed to view staff information, for example, the staff's personal and academic records. The principal can also create a new forum topic, add threads under the topic and review the threads written by the teachers. After visiting the staff module, a principal may exit the system or proceed to other modules.

4.1.1.4 Principal flow chart with respect to student

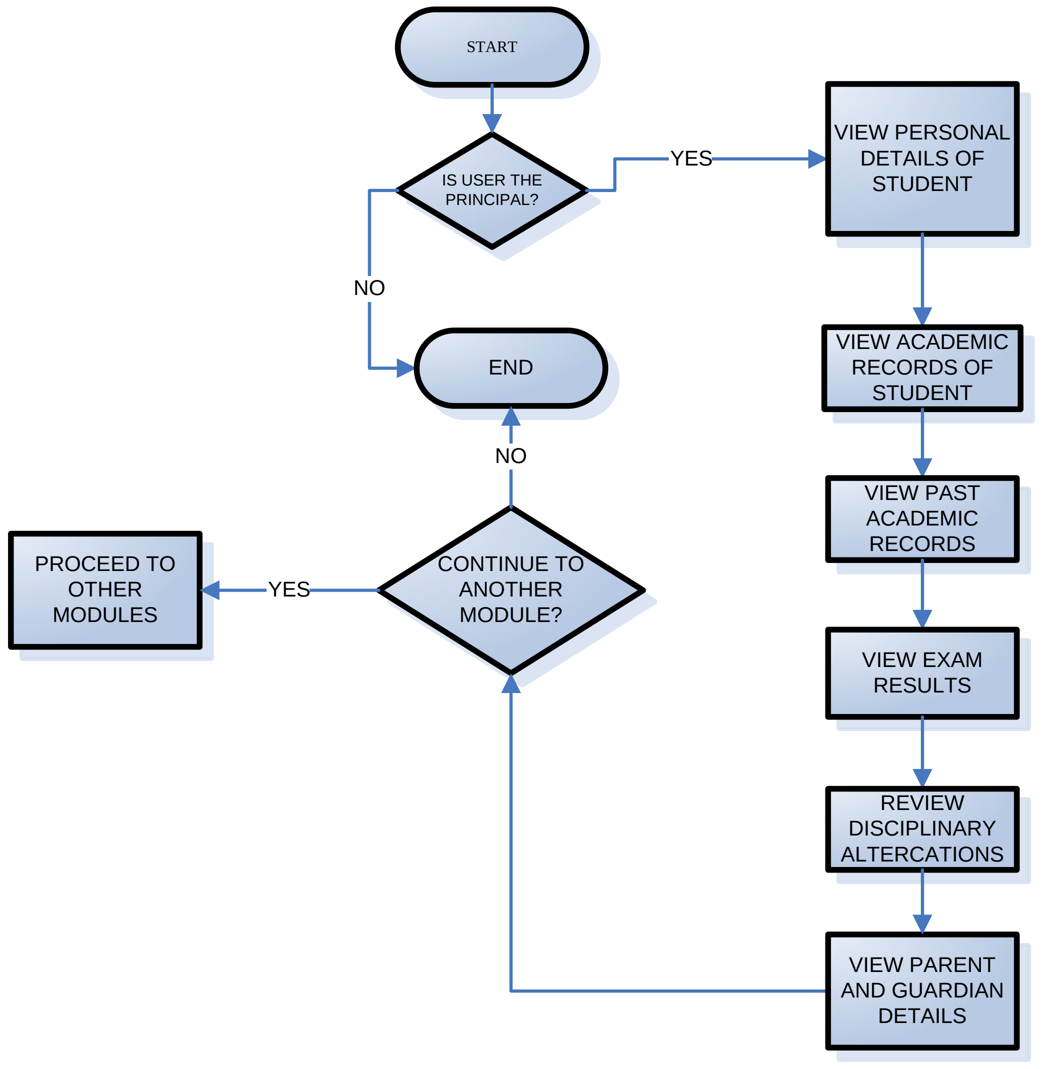

START
IS USER THE PRINCIPAL?
YES
NO
VIEW PERSONAL DETAILS OF STUDENT
VIEW ACADEMIC RECORDS OF STUDENT
END
NO
VIEW PAST ACADEMIC RECORDS
PROCEED TO OTHER MODULES
YES
CONTINUE TO ANOTHER MODULE?
VIEW EXAM RESULTS
REVIEW DISCIPLINARY ALTERCATIONS
VIEW PARENT AND GUARDIAN DETAILS

Figure 15 Viewing of student details by principal

In figure 15, the user enters the system. If verified as the principal, the user may proceed to view student records such as current and previous academic records, personal details, examination results, and details of parents or guardians. The principal can also review any disciplinary matter regarding the student. After visiting the staff module, a principal may exit the system or proceed to other modules.

4.1.2 Ministry of education

The members of the ministry of education have the same privileges as those of the principal and they are given the same priority as the principal.

4.1.3 Administrator

The administrator can be any person who is assigned by the school management to add, edit and delete the records in the school management system. When using the school management system for the first time, there is great amounts of information that must be completed. Information such as setting the academic year, entering student medical records, assigning designations to new staff members and so forth are completed so that other users can start using the system effectively.

An example of information of this sort is the registering of each user by entering the relevant information pertaining to each user. When the necessary information has been entered a username and password are created. This password can later be changed when the users log into their accounts.

4.1.3.1 Entering staff information

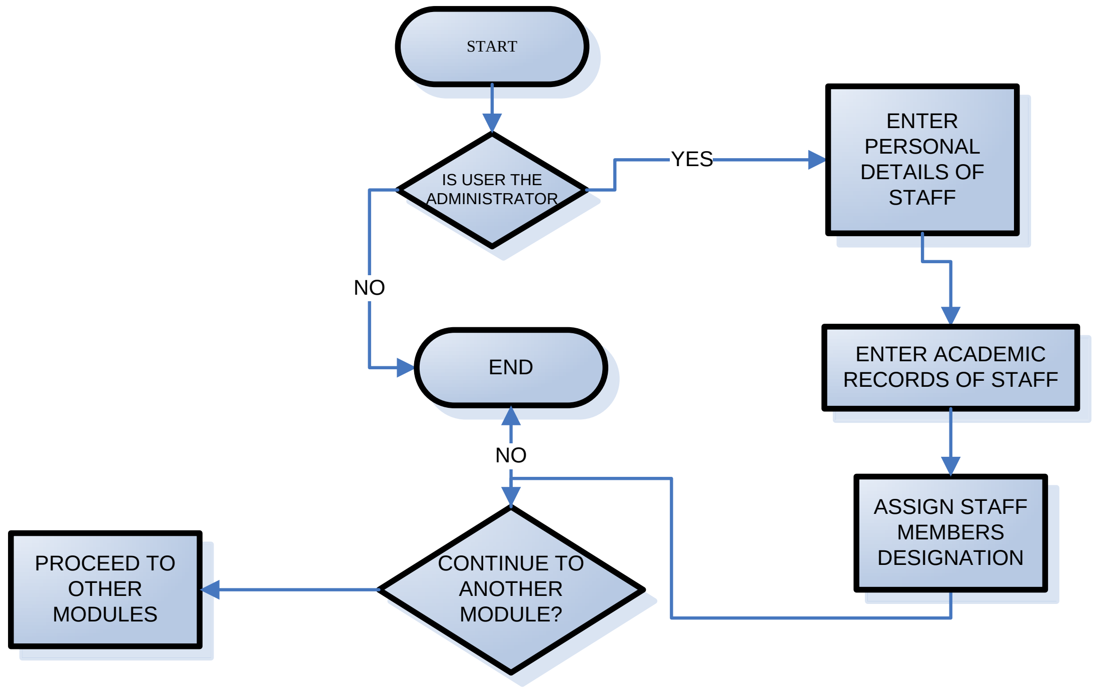

START
IS USER THE ADMINISTRATOR
YES
ENTER PERSONAL DETAILS OF STAFF
NO
END
ENTER ACADEMIC RECORDS OF STAFF
NO
ASSIGN STAFF MEMBERS DESIGNATION
PROCEED TO OTHER MODULES
CONTINUE TO ANOTHER MODULE?

Figure 16 Entering of staff information and assigning of designation by school admin

Any user should be verified in order to confirm that he/she is the school administratior. Once verified, he/she can proceed to the staff section to enter information related to staff members. This information may be staff personal details, academic records or designation assignment. After the information has been captured, the administrator may exit the system or proceed to another module. Figure 16 illustrates the processes explained above.

4.1.3.2 Entering student information

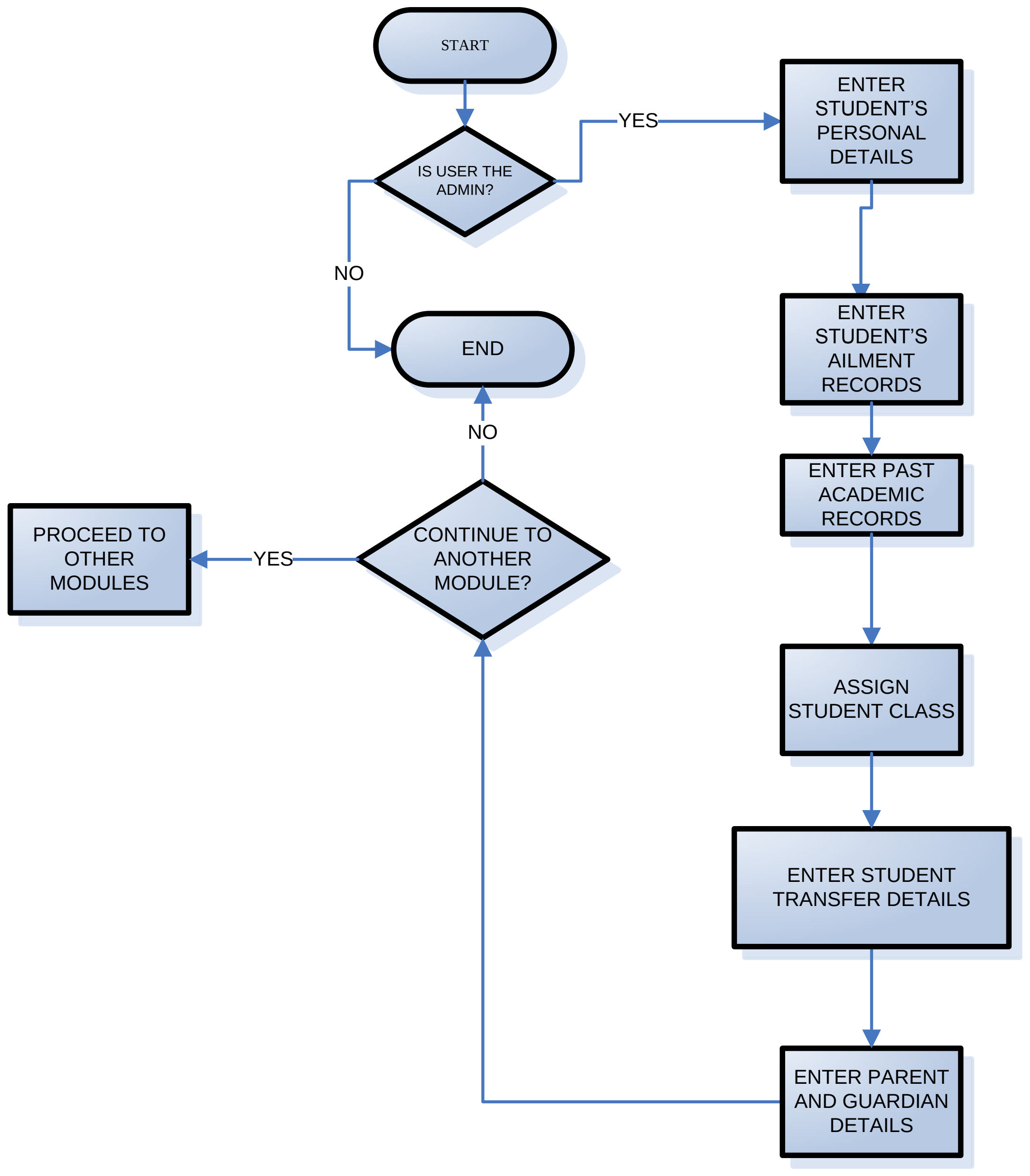

START
IS USER THE ADMIN?
YES
ENTER STUDENT'S PERSONAL DETAILS
NO
END
ENTER STUDENT'S AILMENT RECORDS
NO
ENTER PAST ACADEMIC RECORDS
PROCEED TO OTHER MODULES
YES
CONTINUE TO ANOTHER MODULE?
ASSIGN STUDENT CLASS
ENTER STUDENT TRANSFER DETAILS
ENTER PARENT AND GUARDIAN DETAILS

Figure 17 Entering of student information by school admin

After verification a school administrator enters the system to record student information such as details of parents or guardian, past academic information, ailment information, personal details and class assignment. If all this information has been entered and captured successfully, the administrator proceeds to the next module or exits the system. The process described in this section is explained in figure 17.

4.1.3.3 Entering information about the school

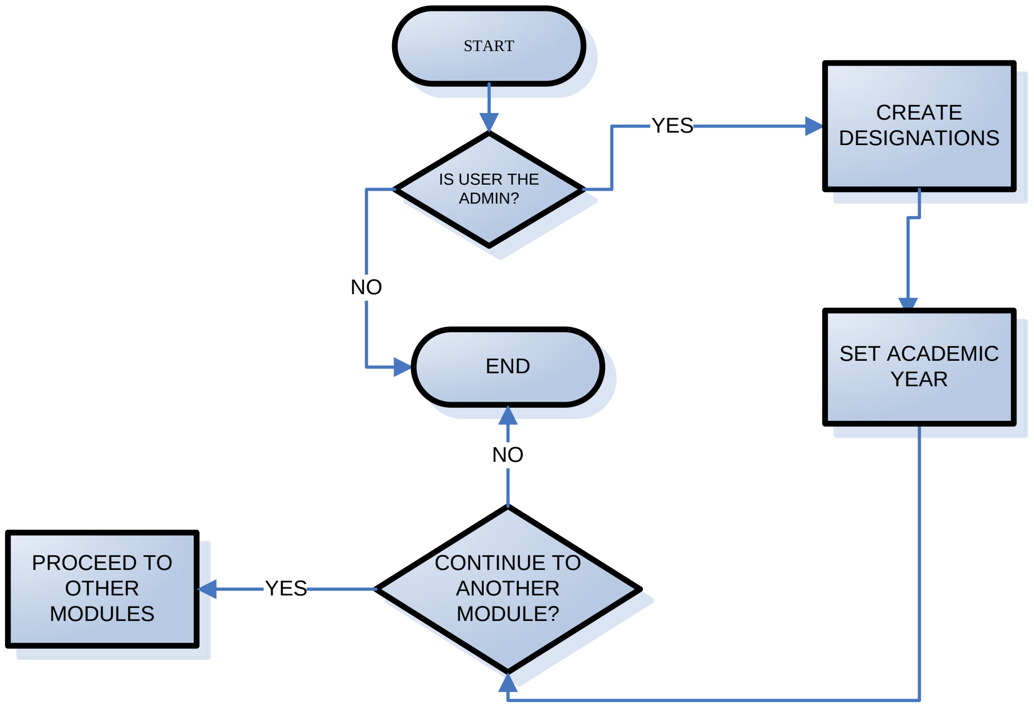


Figure 18 Entering of school information by school admin

The figure 18 is a flowchart showing the steps involved when information is enetered by a school administrator. The school administrator also has the function of setting a new academic year every year as well as creating designations to staff members. Once verified, the school administrator proceeds to perform these functions. When the tasks are completed, he/she either exits the system or proceeds to another module.

4.1.4 Educators/teachers

The school management system provides an educator with all the information regarding a student. Through accessing and understanding the data, an educator should have more insight into students and their background. As mentioned earlier, the types of information include medical records, past academic records, details of the student’s parents and those who act as guardians, if applicable.

Through the discussion forum, teachers can discuss and bring to the notice of the higher authorities some of the difficulties they face in their day-to-day activities. Teachers can also post disciplinary issues regarding students and forward them to the principal, who will approve or disapprove them and suggest the requisite disciplinary measures to be taken. Teachers are notified of upcoming activities through the online notification system.

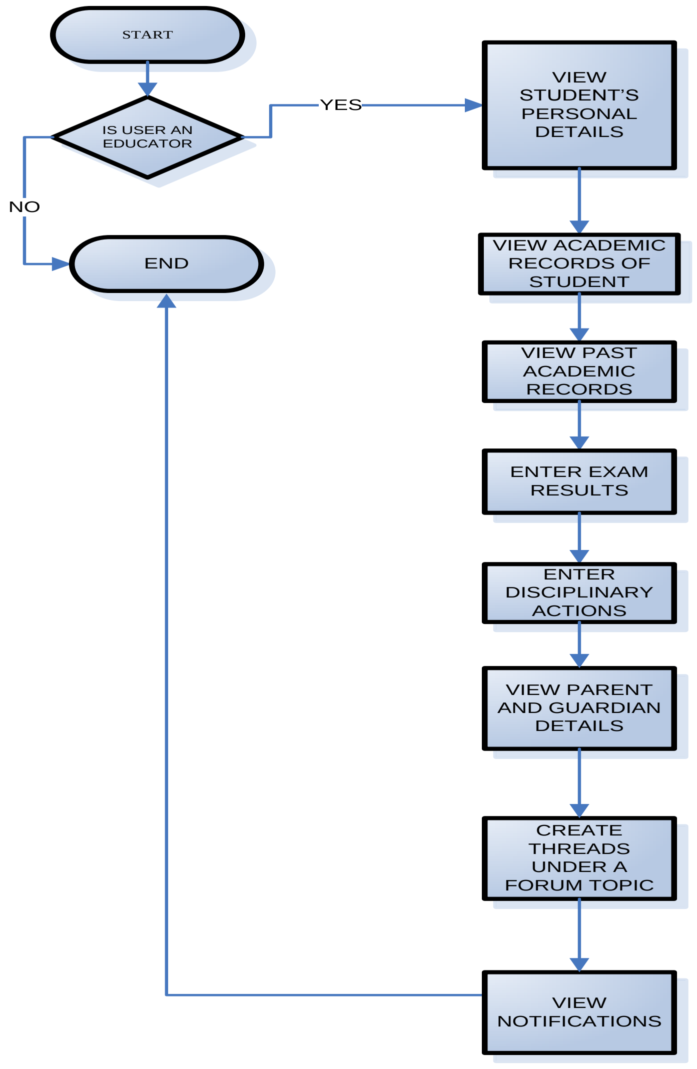

START
IS USER AN EDUCATOR
YES
NO
END
VIEW STUDENT'S PERSONAL DETAILS
VIEW ACADEMIC RECORDS OF STUDENT
VIEW PAST ACADEMIC RECORDS
ENTER EXAM RESULTS
ENTER DISCIPLINARY ACTIONS
VIEW PARENT AND GUARDIAN DETAILS
CREATE THREADS UNDER A FORUM TOPIC
VIEW NOTIFICATIONS

Figure 19 Entering of staff information by school admin

Once verified as an educator, a user can view student records such as past and current academic records, personal information, parent and guardian information, and notifications. An educator can capture information related to a student, for instance examination results and disciplinary issues. A staff member can also create threads under a forum. A pictorial explanantion of the activities explained is given in figure19.

4.1.5 Parents

By use of the system, parents can monitor their child's academic progress and be notified about their child's disciplinary matters as soon as they occur. This is done so as to keep the parents abreast of all the activities pertaining to their child. Parents can also be notified through the notification system of the upcoming events and activities in the school.

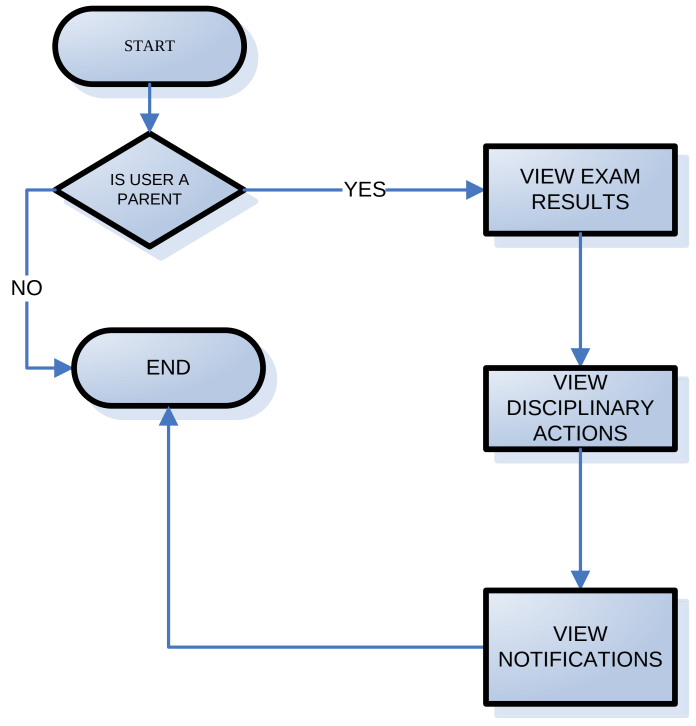


Figure 20 Parent activities using the system

A user, if verified as a parent, may enter the system to view the results of his/her child. A parent will be notified of the child's undisciplined behaviour, if exhibited. Upcoming events will also be notified to the parent. Once a parent has viewed the required information, he/she may exit the system. A pictorial explananation of the activities explained is given in figure 20.

4.1.6 Guardians

If parents are not able to fulfil their duties, a guardian is assigned to carry out those duties. He/she has the same privileges as that of the parents regarding use of the system.

4.1.7 Students

Students can access their examination results through the system. They can also learn of upcoming events through the online notification system.

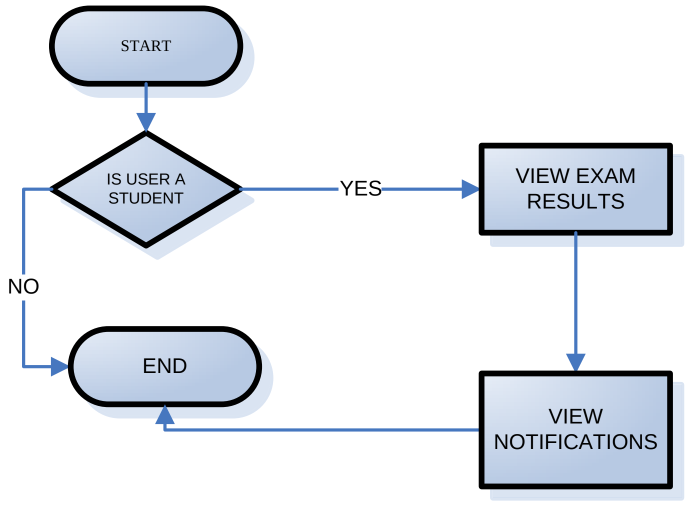


Figure 21 Illustrations of student activities

A student, after verification, may view his/her exam results for each term, as well as notification of upcoming events in the school. A pictorial explanantion of the activities explained is given in figure 21.

4.1.8 Site administrator

The site administrator is responsible for accepting schools that have requested to use the system.

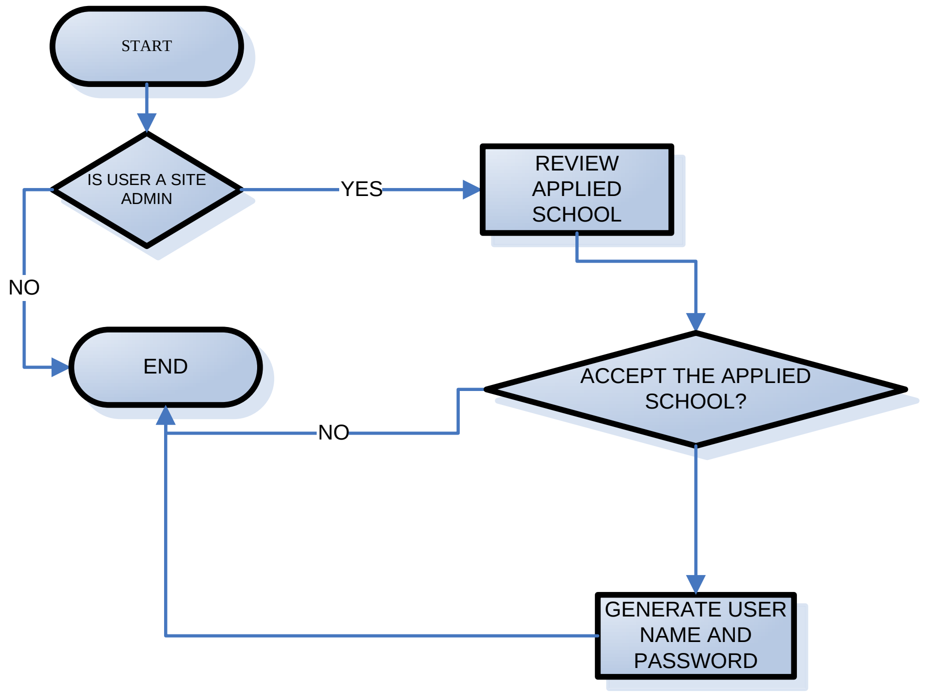


Figure 22 Site admin activities

After verification, a site administrator enters the system. Schools that wish to use the system must be registered. The site administrator reviews registered schools and, if all details have been enetered correctly accepts them. After accepting a school, a username and password is generated, and the school is included in the system. A pictorial explanantion of the activities explained is given in figure 22.

4.2 System requirements

As mentioned previously, the cost-effective school management system is developed with the intention of implementing it in cloud. In order to access an SaaS application, it is mandatory that a user has access to high-speed Internet and a web browser.

4.2.1 Functional requirements

Functional requirements and non-functional requirements refer to aspects of the environment that must be present for an application to run smoothly. As an analyst or a designer it is important to ensure that the following matters are addressed during design:

- Security
- Scalability
- Availability/reliability
- Performance
- Configurability
- Flexibility
- Usability
- Interoperability

Not all of these requirements have to be met during analysis and design, and for every SaaS application the functional requirement may change. These requirements are subject to change depending on the following factors:

- Budget
- Target audience of application

- Level of security required

Below follows an explanation of functional requirements.

4.2.2 Security requirements

Specific security demands must be met when developing a cloud or SaaS application [7]. Firstly, a user must ask a system for permission to access the system and, based on the user right, the system must grant permission. Before permission is granted, authentication and verification must be carried out by the system. When using the shared database, shared schema method, depending on a user’s authentication information, his/her relevant data must be accessed and produced.

4.2.3 Scalability

Scalability is the ability to add or remove schools without affecting performance. It is one of the most important advantages of a cloud approach – because the cloud scales up and down as the demand of SaaS application changes – typically by adding or removing virtual machines
Software must be solid and well designed at architectural level so that it can support the theoretical maximum of the number of clients [8].

4.2.4 Reliability

Reliability refers to the underlying success of any product; a product must be able to deliver the expected performance when a client needs it. A clearly defined service level agreement (SLA) requires that any SaaS offering must adhere to requirements or face legal action. Improvement in technology and increasing competition have made customers less tolerant of errors and delay in service. Furthermore, software must be designed so that it is 99.9% bug free. Any sort of data loss is unacceptable to customers. The SLAs of the host should be aligned with those of the design engineer. That is during development of an application the exixting service level agreements between the hosts should be taken into consideration by the design engineer. [8]

4.2.5 Performance

The performance of software is measured in terms of its ability to act before the client starts thinking about its speed.

Data that are frequently accessed must be cached, which aids in the process of retrieving the data more easily. It is important to ensure that all queries in the database use the appropriate indexes, which helps to make the database search and query process more efficient. When the operations in a network increase, it slows down query processing; thus, by reducing network operations, performance can be improved. If excessive traffic occurs in the network, the best thing to do would be to correctly load balance and, if possible, localise servers. Compressing data where possible also improves performance. Finally, it is vital that software not be over engineered.

The above points such as caching and database indexing may not always be the solution for performance and should be done as a calculated exercise [8].

4.2.6 Configurability

Configurability refers to the variations and changes that a client is allowed to make to an application. Configuration allows users to feel at home and to adopt a more organised approach to an application. Although it increases the popularity of a vendor's offering, it may prove to be difficult to program.

4.2.7 Flexibility/extensibility

Software should be extended, and new additions must be made periodically. This allows existing users to do more with the services that are offered, and will attract more new customers.

In order to extend existing software, provisions must be made available in architectural design of the software. However, the degree to which a software needs extension, cannot be predetermined. Hence, the maximum space needs to be allotted during the design phase. It is also important that existing customers be given the choice of accepting these new changes or continuing with the existing version [8].

4.2.8 Usability

Usability means taking into consideration the user experience, rather than providing users with simply any user interface. When the number of users in an SaaS environment increase, for example, tenfold, usability becomes even more difficult to provide for. It becomes difficult to provide every user's desired experience. A single software instance needs to provide a unique experience and functionalities to different customers and, in turn, the various requirement of each customer. During the graphic user interface (GUI) development, this fact should be kept in mind [8].

4.2.9 Interoperability

There is always a need for integration of the system under development with other systems providing specialised services. As precaution, software must be developed so that it interoperates with other software in order to produce service-oriented architecture. However, interoperability should not be provided at the cost of security.

4.3 Summary

The school management system has to be designed in such a manner that all the activities taking place in the school are represented and monitored. Each user must be able to view or change only the information that he/she is authorised to.

When developing software it is imperative that the design of the system allows future modification and interoperability to new platforms. Newly designed software must fulfil the needs of an increasing number of users and meet their demands, while not compromising performance. Furthermore, it must allow moderate change in configuration by the user to give the program a user-friendly feel.

Chapter 5: Design and Development

This stage has two important elements: the design of a multi-tenant database that caters for the needs of multiple schools and the design and implementation of a SaaS ready application that securely interacts with the multi-tenant database. This chapter discusses the database design. Chapter 6 looks at the application design.

5.1 Database design

A database forms the backbone of any dynamic software. It stores information and allows actions such as viewing, editing and deleting to be carried out. The design of a database ensures that these actions are carried out without causing anomalies such as loss of data during a database action. Anomalies can ruin the entire purpose of a database as well as the software that was developed based on it. To prevent anomalies, an action known as **normalising** should be carried out. Normalising is a set of rules that must be followed to ensure that datum is arranged in a way that anomalies are avoided and that data are saved, retrieved, changed and deleted, all without affecting other data. Normalising a database helps maintain data isolation and data integrity in a database.

Data isolation means that during multiple transactions by multiple users, the transaction that each user carries out is hidden from the other. Data isolation ensures that a user's transaction does not interfere with another user's transaction.

Data integrity means that during the updation of a field in a database table, only that field is subjected to change and not the entire table. It is mandatory that during design itself that a table be able to maintain data isolation and data integrity.

When selecting a database to store information, there are different options. Some types of databases are [9]:

- Entity relationship databases
- Object model
- Relational model
- Hierarchical model
- Network model

According to user requirements, the desired database model also changes. The system proposed in this study will use a relational database and data will be modelled using E-R models.The E-R model was a technique suggested by Peter Chen in 1976, and theories and ideas of this model have been traced back to before 1976. The E-R model is based on three building blocks, namely entities, attributes and relationship, which will be discussed below.

5.1.1 Entities

An entity is anything that is capable of independent existence; it is always identified and recognised as an individual entity. An entity may be anything from a house to a car, to an individual. An entity is represented by a rectangle

5.1.2 Relationships

Relationship describes how one entity is related to another and works to link them. A relationship is represented by a rhombus.

5.1.3 Attributes

Attributes refer to the characteristics of both an entity and or a relationship. For example, a student is an entity and has the attributes name, student number and so forth. A student studies in a school: hence, the relationship between student and school is studying. In order to study in a school, a student has to enrol. The date of enrolment is an attribute of the study relationship. An attribute is represented by an ellipse. Figure 21 is a pictorially representation of attributes, relationships and entities.

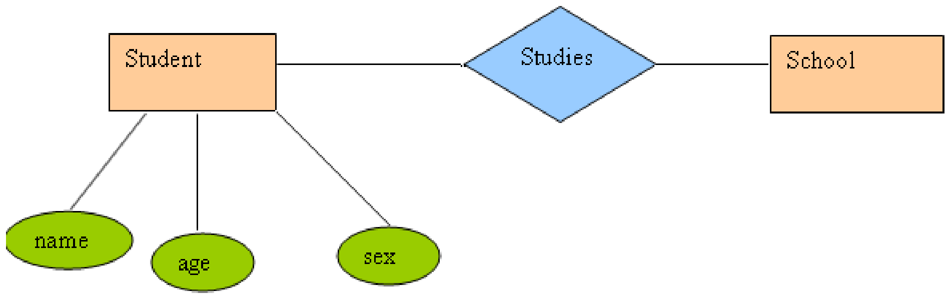


Figure 23 E-R diagram of relationship between a student and school

There are different ways of representing an entity and its relationships and attributes. One example is seen in figure 23. Another method is shown using the software MS Visio illustrated below: Figure 24 shows an E-R diagram constructed in MS-Visio.

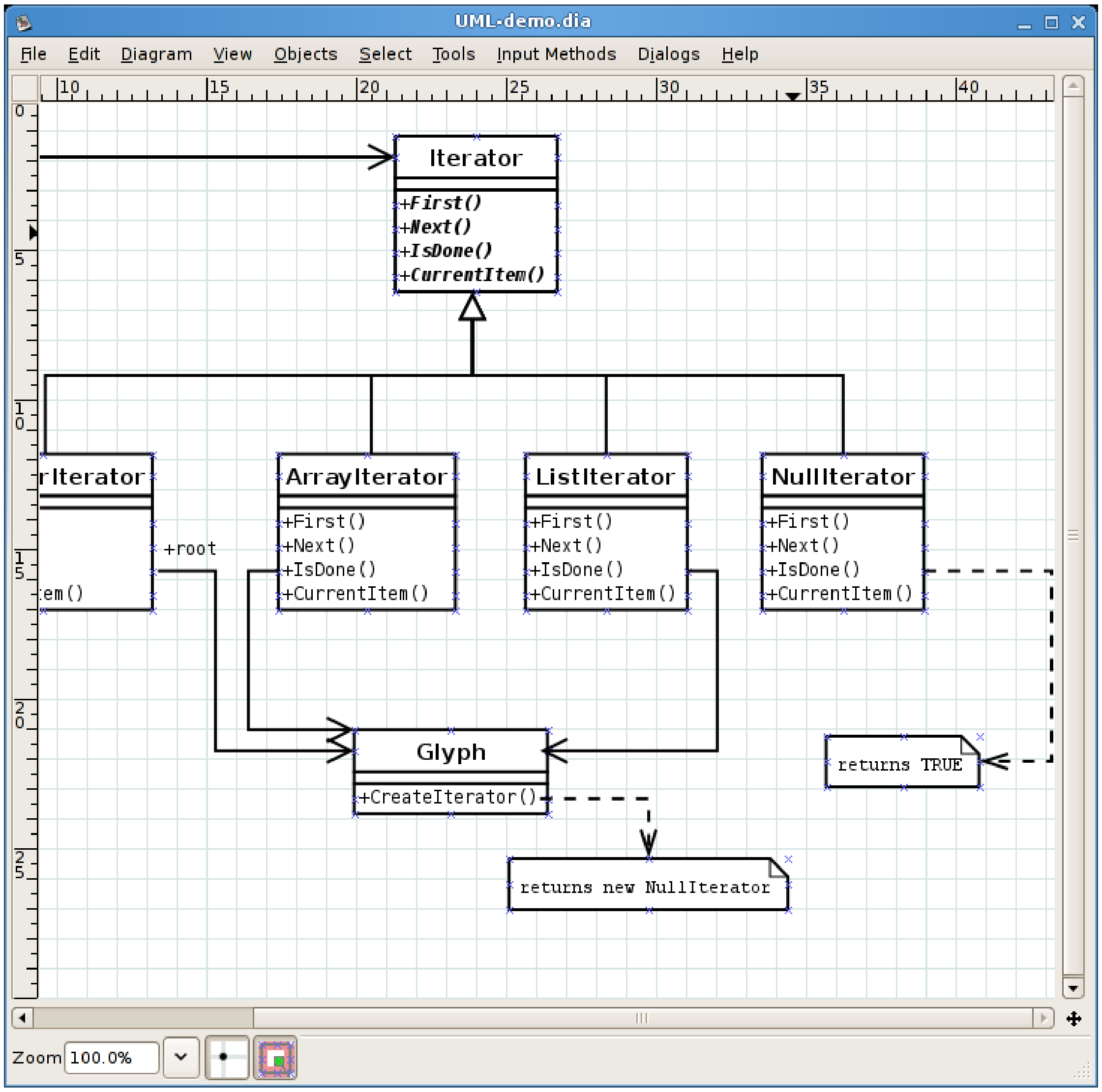


Figure 24 Use of MS Visio to represent relationship between entities [10]

In MS Visio, entities are represented as boxes, with the name or type of entity mentioned at the top. The attributes are separated from the entity name by two solid lines. Relationships are illustrated using arrows.

5.2 E-R representation of the modules of the system

The E-R representation of the modules of the system aims at explaining to the reader novice how the database of the system, has been designed. Through database design one will be able to understand how multitenancy has been implemented.

5.2.1 School creation and user login

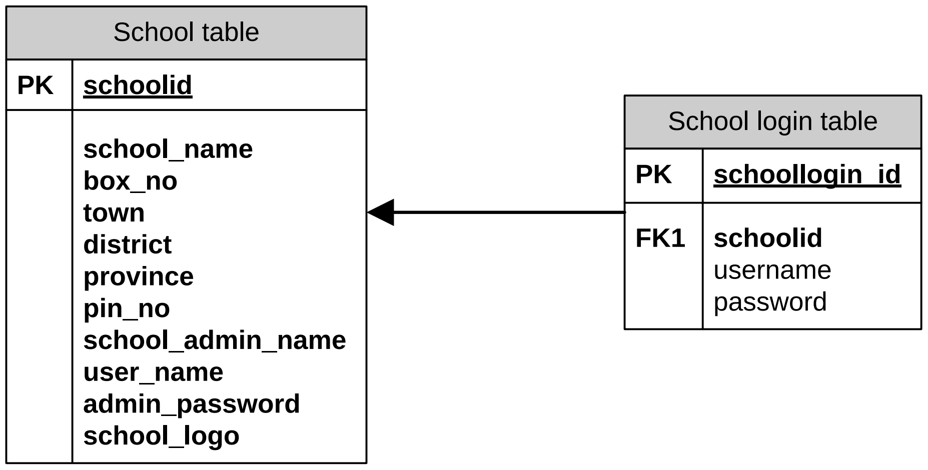


Figure 25 E-R diagram of the school login module

The above diagram consists of two tables, one being the school table and the other being the school login table. The school table contains information about the school: it has attributes such as school name, address, the name of the school administrator, the school logo, and the username and password assigned to the school admin. The school login table, on the other hand, contains attributes such as schoolid, username and password, with schoologin_id as the primary key. Both these tables are related due to the fact that schoolid in the school table is referenced as a foreign key in the school login table. When a school logs on to the system, verification takes place as to whether such a username and password exists in the school login table. When the site admin registers a school, its username and password are saved automatically into the school login table and the school becomes a member of the system. A pictorial explanantion of the activities explained is given in figure 25.

5.2.1.1 User login

Different users have different rights and are thus redirected to different pages. The user login helps with the navigation of different users to their respective points. The user type table is a pre-existing table that contains the type of users that the system was designed for, for example, principal, staff member, student, parent and guardian. During the registration of a new user, his/her user type, username and password are saved into the user creation table. The school_id attribute ensures security so that a user of a different school cannot, by accident, gain access to the records of other schools. A pictorial explanantion of the activities explained is given in figure 26.

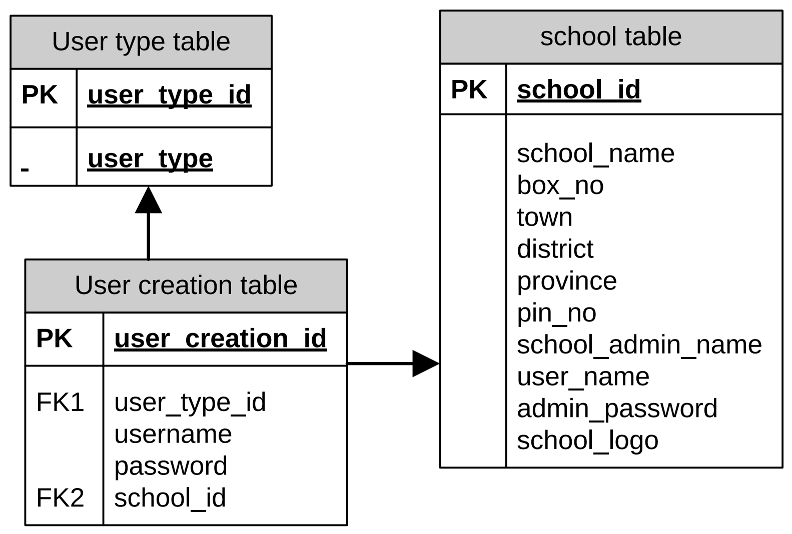


Figure 26 User login

5.2.2 School administrator module

The school admin mainly inputs information about the members of a school as well as some general information about the school. Member information refers to information about the students, their parents or guardians and staff members, whereas information about the school refers to information about the number of classes in the school and the subjects offered in the school. For better understanding, this section will be explained in more detail with respect to each user.

5.2.2.1 Representation of student information

As shown on the next page, the student details table captures the personal information of a student such as name, address, gender, date of birth (dob), enrolment date and so forth. One of the fields in the student details table is schoolid. This table is related to the school table. This helps to identify to which school a student belongs. The ailment table captures information about the ailments from which a student may suffer. The previous academic details table captures information about the student’s previous academic records, while the student transfer table records information about students who have left the school. The student class details contain information about the class, division and academic year in which the student is currently. Every table other than the school table contains an attribute known as studentid, which identifies the student to which all this information belongs. Once the student has been identified, the system can identify the school to which the student belongs. A pictorial explanantion of the activities explained is given in figure 27.

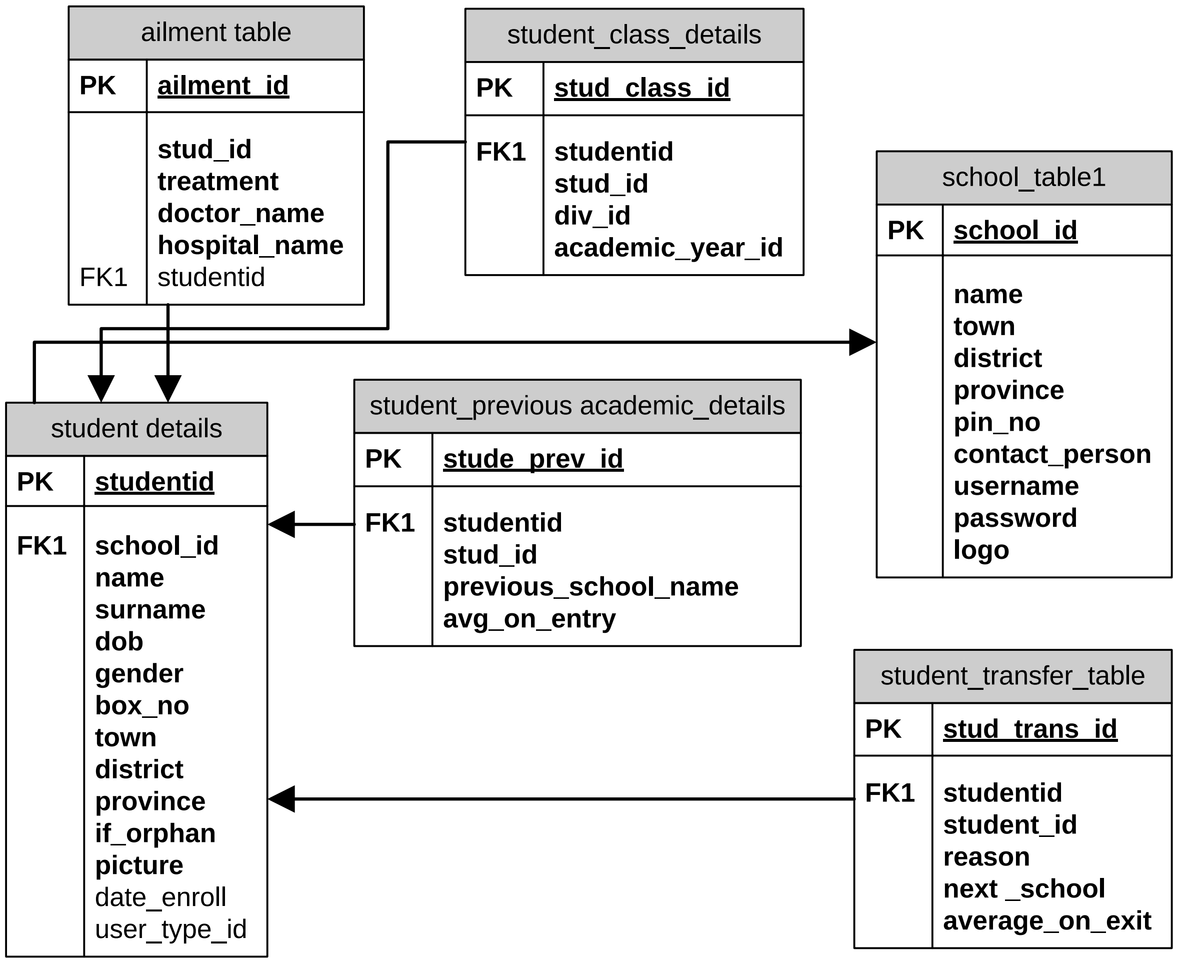


Figure 27 E-R representation of how school admin captures student information

5.2.2.2 Representation of staff information

The staff basic details table captures the personal information of a staff member such as name, gender, address, persal number and marital status. This table also contains the attribute schoolid, which is included to identify the school to which a particular staff member belongs. The staff education detail table captures a staff member's academic information. The staff_detail_id is included so that the system can identify which staff member these qualifications belong to and, in turn, at which school this staff member is employed. A pictorial explanantion of the activities explained is given in figure 28.

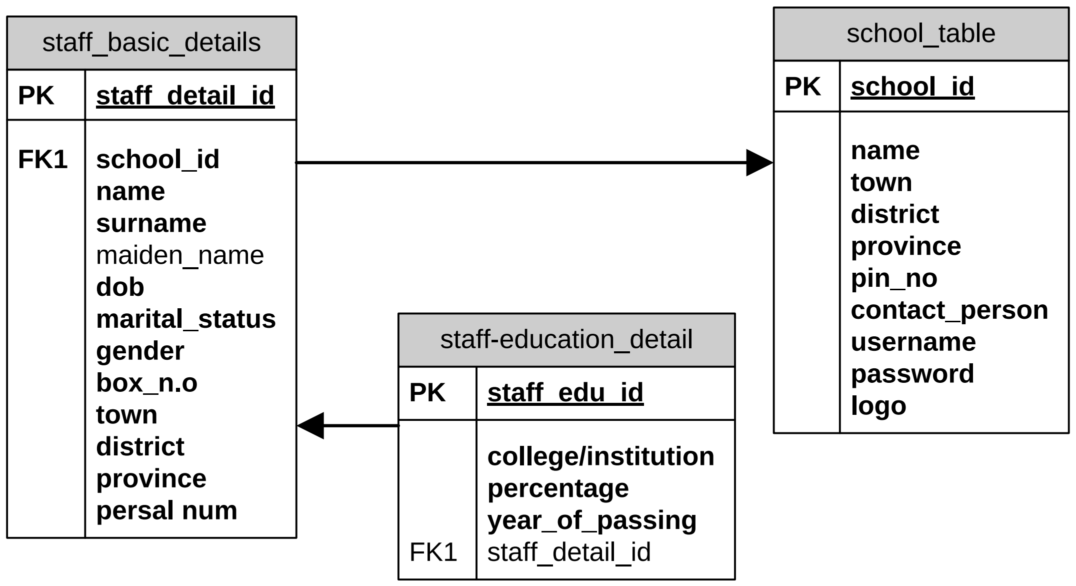


Figure 28 E-R representation of how school admin captures staff information

Each staff member is assigned a designation, which is to distinguish whether they are educators or the principal. The designation table captures the different designations in a school, while the designation master table records the staff members and their respective designations. A pictorial explanantion of the activities explained is given in figure 29.

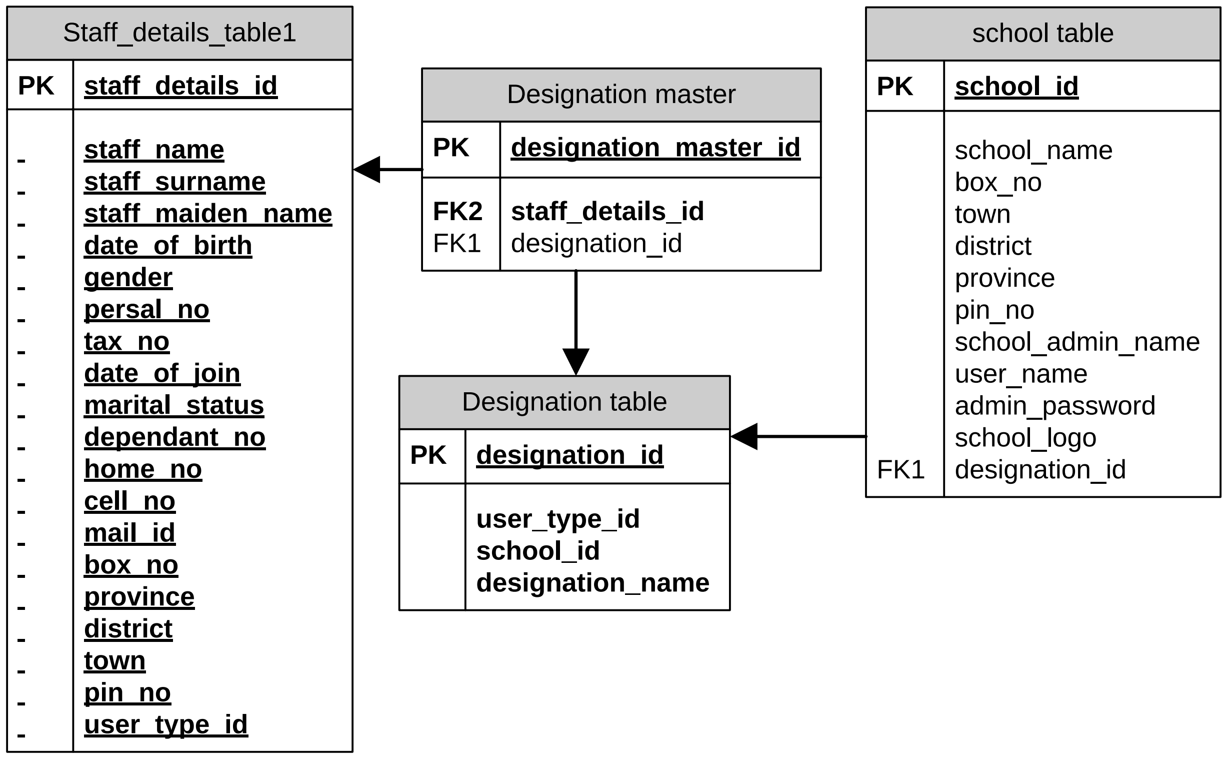


Figure 29 E-R representation of how school admin assigns designation

5.2.2.3 Representation of parent guardian information

The parent table gathers personal information about the parents such as name, address, occupation, identification number and cell phone number. The guardian table stores similar information than the parent table. Both the parent table and the guardian table contain the attribute student_id which recognises the child of the parents. Identifying the student can also help identify the school to which the student belongs. A pictorial explanantion of the activities explained is given in figure 30.

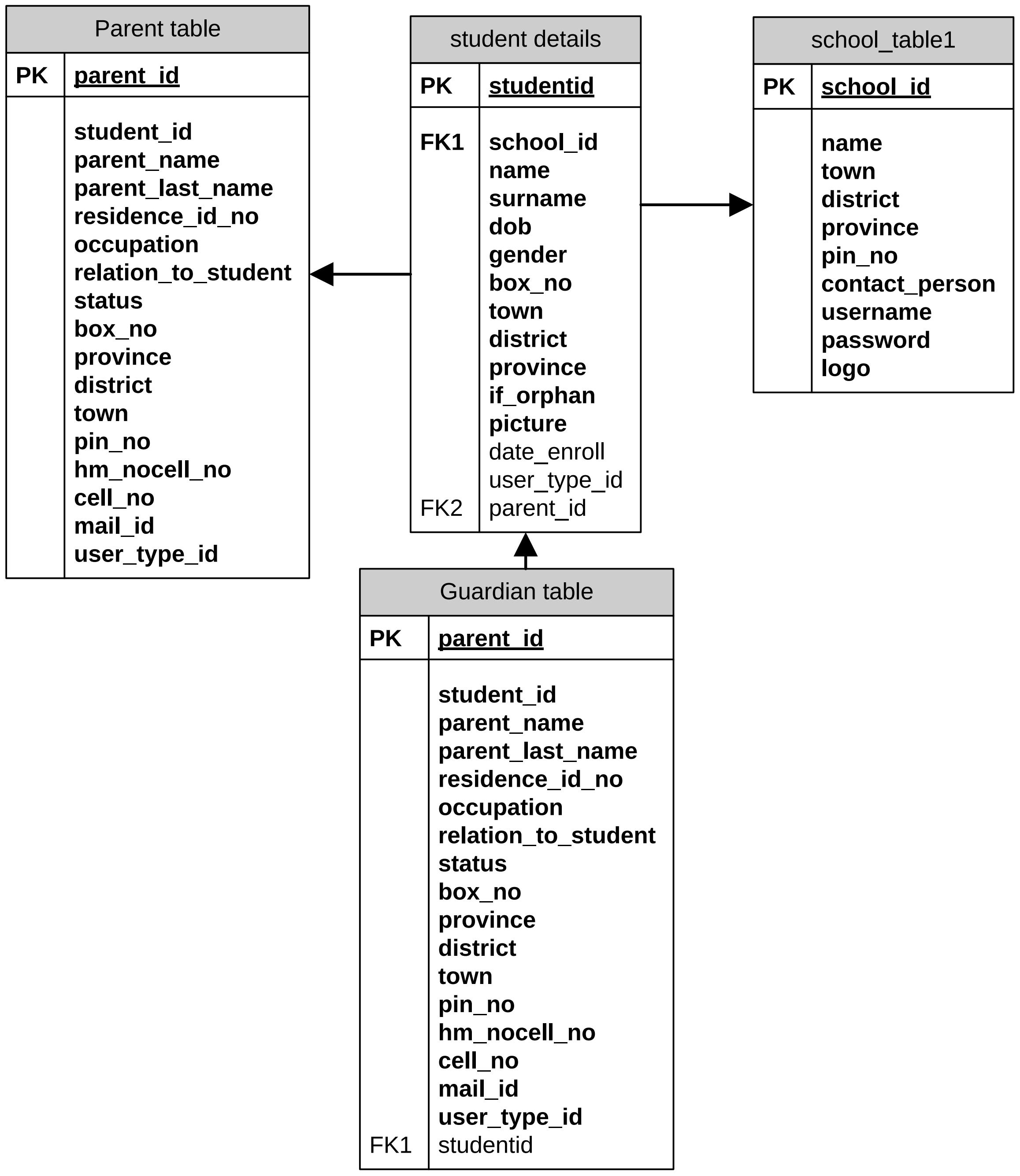


Figure 30 E-R representation of how school admin captures information

5.2.2.4 Representation of school information

Every school consists of classes and every class is divided into subdivisions or divisions. Every student is assigned to a class for the specific academic year, and a staff member is assigned to teach a particular subject in that class.

Firstly, a class creation table is created in MS SQL. A class creation table consists of class_creation_id as its primary key and school_id and class as its attributes. The school_id helps to identify to which school the class belongs. Next, a division table is created with student_div_id as its primary key and division, class_creation_id and stud _class_details as its attributes. This identifies the class to which the subdivision belongs and, from the stud_class_details foreign key, which students are present in that class. From the student_id attribute, one can find out more details of the student as well as which school the student attends. Hence, each student is assigned to a class and subdivision of the school to which he/she belongs. A scenario where a student is assigned to anther school does not occur in this software.

Every class is assigned subjects. Firstly, a subject table is created that contains the subjectid as the primary key and school_id and subject name as attributes, which identify the subjects taken in a particular school. Then a subject class table is created, which contains both subjectid and student_div_id as attributes, with subject_creation_id as the primary key. The subjectid and student_div_id help identify which subject is taken in which subdivision of which class. From the subject class table, one can write SQL queries to determine the subjects that the students of that class and its subdivisions take. The staff_class_table contains the attributes subjectid, student_div_id and the staff_detail_id. This table helps to identify which staff member is assigned to teach a particular subject in a particular class. A pictorial explanantion of the activities explained is given in figure 31.

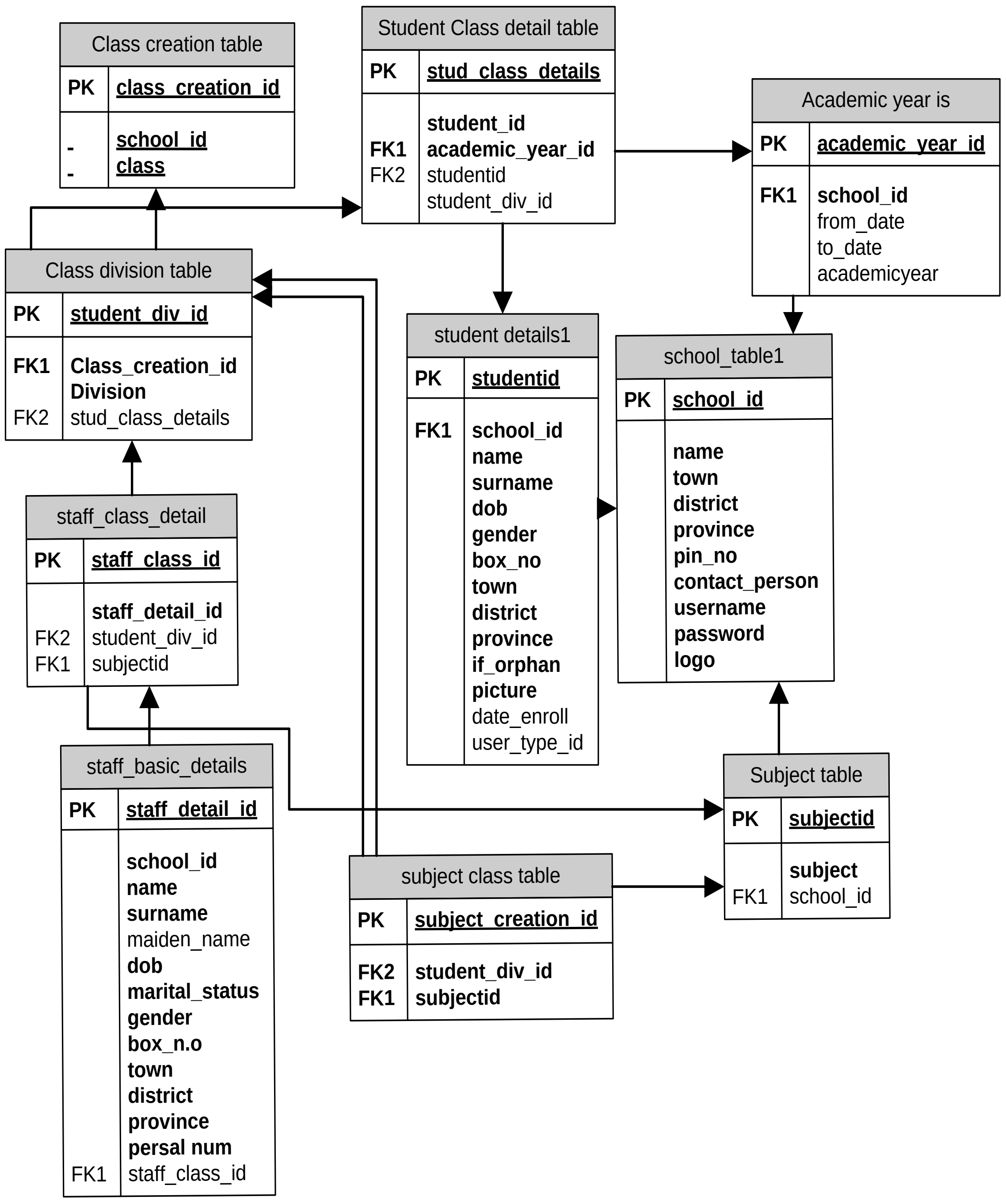

Class creation table
PK class_creation_id
school_id
class
Student Class detail table
PK stud_class_details
FK1 FK2 student_id academic_year_id studentid student_div_id
Academic year is
PK academic_year_id
FK1 school_id from_date to_date academicyear
Class division table
PK student_div_id
FK1 FK2 Class_creation_id Division stud_class_details
student details1
PK studentid
FK1 school_id name surname dob gender box_no town district province if_orphan picture date_enroll user_type_id
school_table1
PK school_id
name town district province pin_no contact_person username password logo
staff_class_detail
PK staff_class_id
FK2 FK1 staff_detail_id student_div_id subjectid
staff_basic_details
PK staff_detail_id
school_id name surname maiden_name dob marital_status gender box_n.o town district province persal num
FK1 staff_class_id
Subject table
PK subjectid
FK1 subject school_id
subject class table
PK subject_creation_id
FK2 FK1 student_div_id subjectid

Figure 31 E-R representation of how school admin captures school information

5.2.3 Principal

The principal should be able to view all the activities in a school and have access to information regarding all students and staff members.

5.2.3.1 Viewing staff member information

A principal enters into the system using his/her username and password. Once verified, he/she can view the personal details of the staff members in the school. The staff_basic_detail table contains the attribute school_id, which ensures that only members of his/her school can be viewed. Front-end coding ensures that school_id is saved during login and that it is referenced during all SQL queries. This helps maintain data integrity and data isolation. SQL queries can be written to access the staff class details. The staff class table contains information about the subjects taken as well as class subdivision in which the staff member teaches. A principal can also view the leave that the staff requested. A pictorial explananation of the activities explained is given in figure 32.

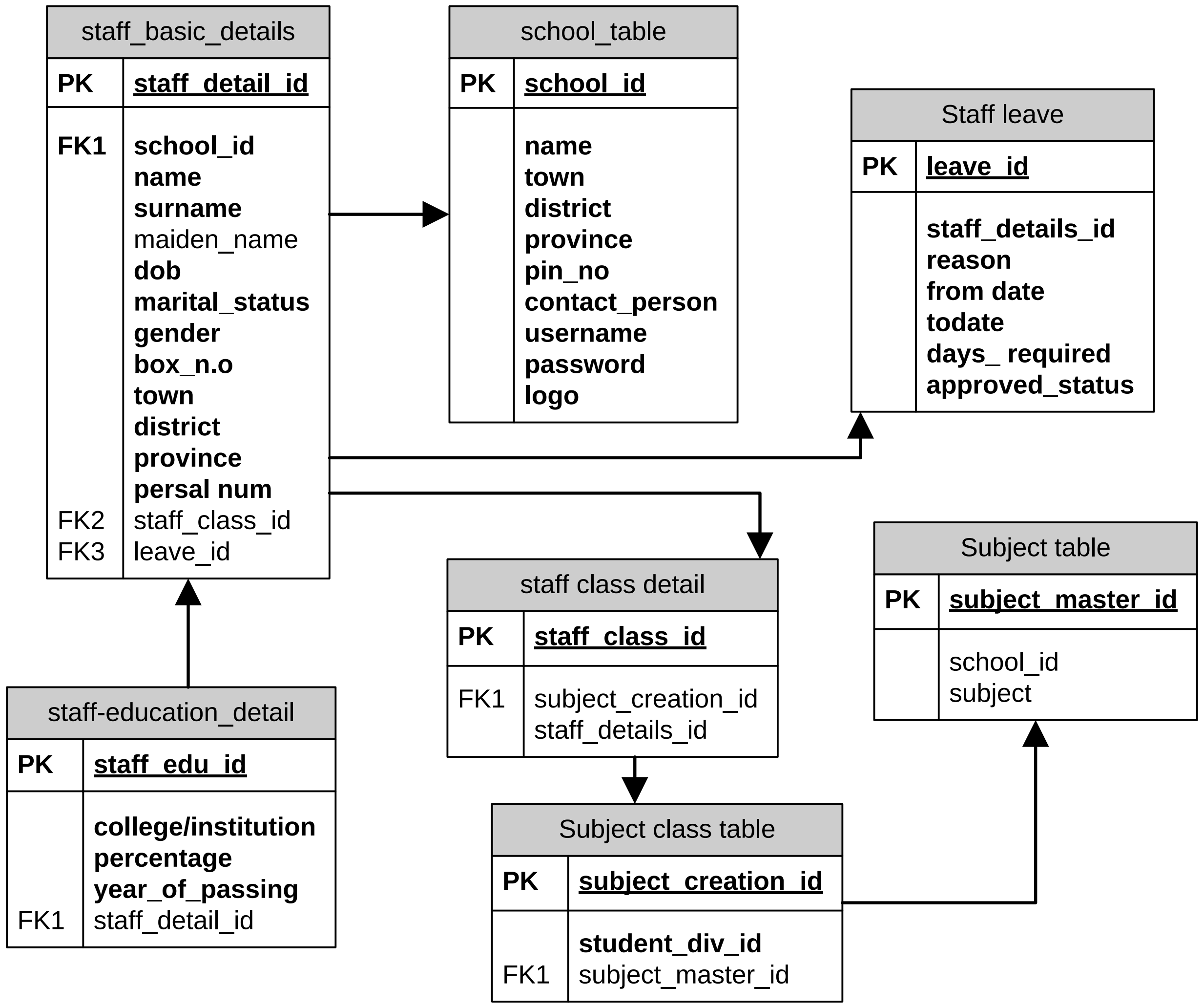


Figure 32 E-R representation of how principal accesses staff information

5.2.3.2 Viewing student information

To view information about a student, a principal will firstly select the class and subdivision to which the student belongs. This will display all the members of that class. The relevant student can be found from the tabulated list. During a staff login, the attribute schoolid is stored as a session variable – this schoolid attribute is a criterion when any information from the database is requested. It ensures that only data pertaining to the relevant school is displayed. With the aid of the studentid attribute, it is possible to search the database for a student’s ailment details, term-wise marks, previous academic details and disciplinary records. The staff_detail_id also shows which staff posted a disciplinary record of the student. A pictorial explanantion of the activities explained is given in figure 33.

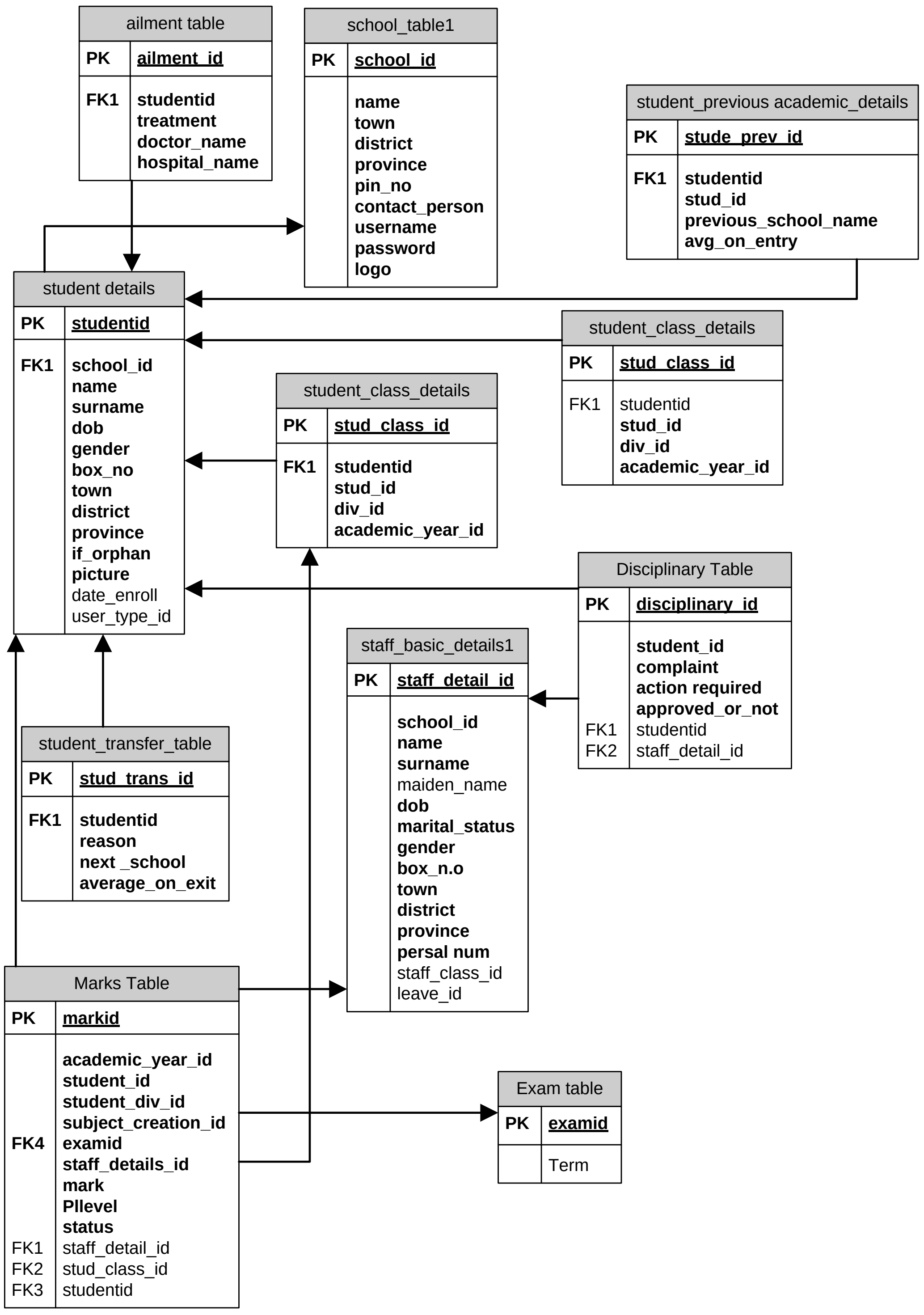


Figure 33 E-R representation of how principal views student information

5.2.3.3 Viewing parent or guardian information

The student_class_details table will help to identify the student in a particular class. By identifying the class and subdivision, the principal is able to obtain the attribute div_id and from there the student_id. The list of student_ids will provide the student in that class. By selecting a student in that class, the principal can access the details of his/her parents or guardian. Both parent and guardian tables contain the field studentid, which ensures that only information about the selected student is displayed. A pictorial explanantion of the activities explained is given in figure 34.

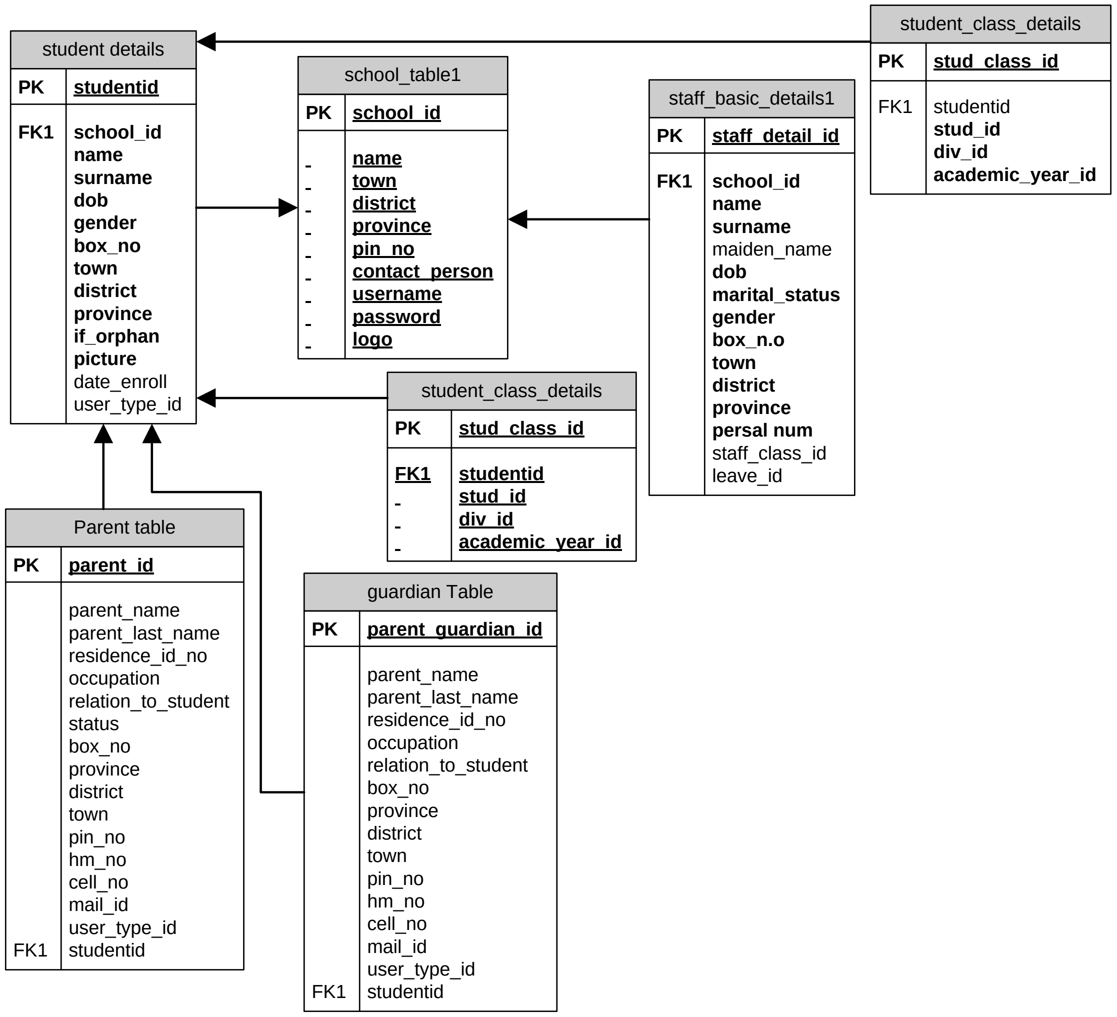


Figure 34 E-R representation of how principal views parent/ guardian information

5.2.3.4 Creating and writing forums

Only the principal has the right to create a forum. After verification, the principal can create a forum. He/she also has to approve the threads that educators post in the forum reply table so that the status attribute can change from not approved to approved. The principal can reply to the threads teachers post. A pictorial explanantion of the activities explained is given in figure 35.

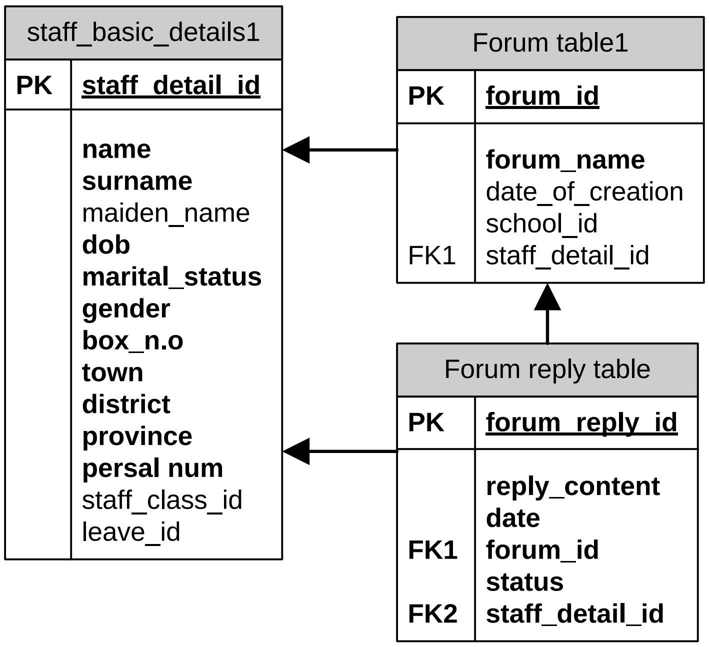


Figure 35 E-R representation of how principal views forum

5.2.3.5 Creating notifications

A principal creates notifications of upcoming events in the school in order to notify the staff members, students and the parents. Only a principal is authorised to create a notification, while others are allowed to view notifications. A pictorial explanantion of the activities explained is given in figure 36.

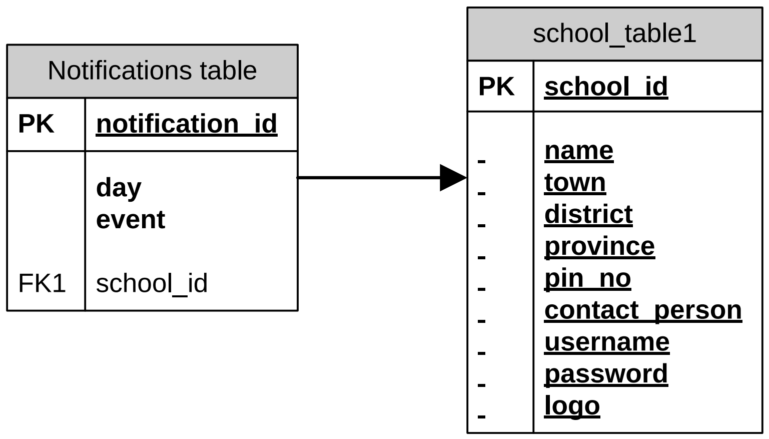


Figure 36 E-R representation of how principal views notifications

5.2.4 Staff members

A staff member should be able to view all information about a student. Furthermore, they must be able to write threads under a forum topic and view notifications.

5.2.4.1 Viewing student information

In order to view information about a student, an educator will firstly select the class and subdivision to which the student belongs. This will display all the members of that class. The relevant student can be found from the tabulated list. When a staff member logs in, the attribute schoolid is stored as a session variable. This schoolid attribute is a criterion when any information from the database is requested. It ensures that data pertaining only to the relevant school is displayed. Using the studentid attribute, it is possible to search the database for a student's ailment details, term-wise marks, previous academic details and disciplinary records. The staff_detail_id also shows which staff posted a disciplinary record of the student. A pictorial explanantion of the activities explained is given in figure 37.

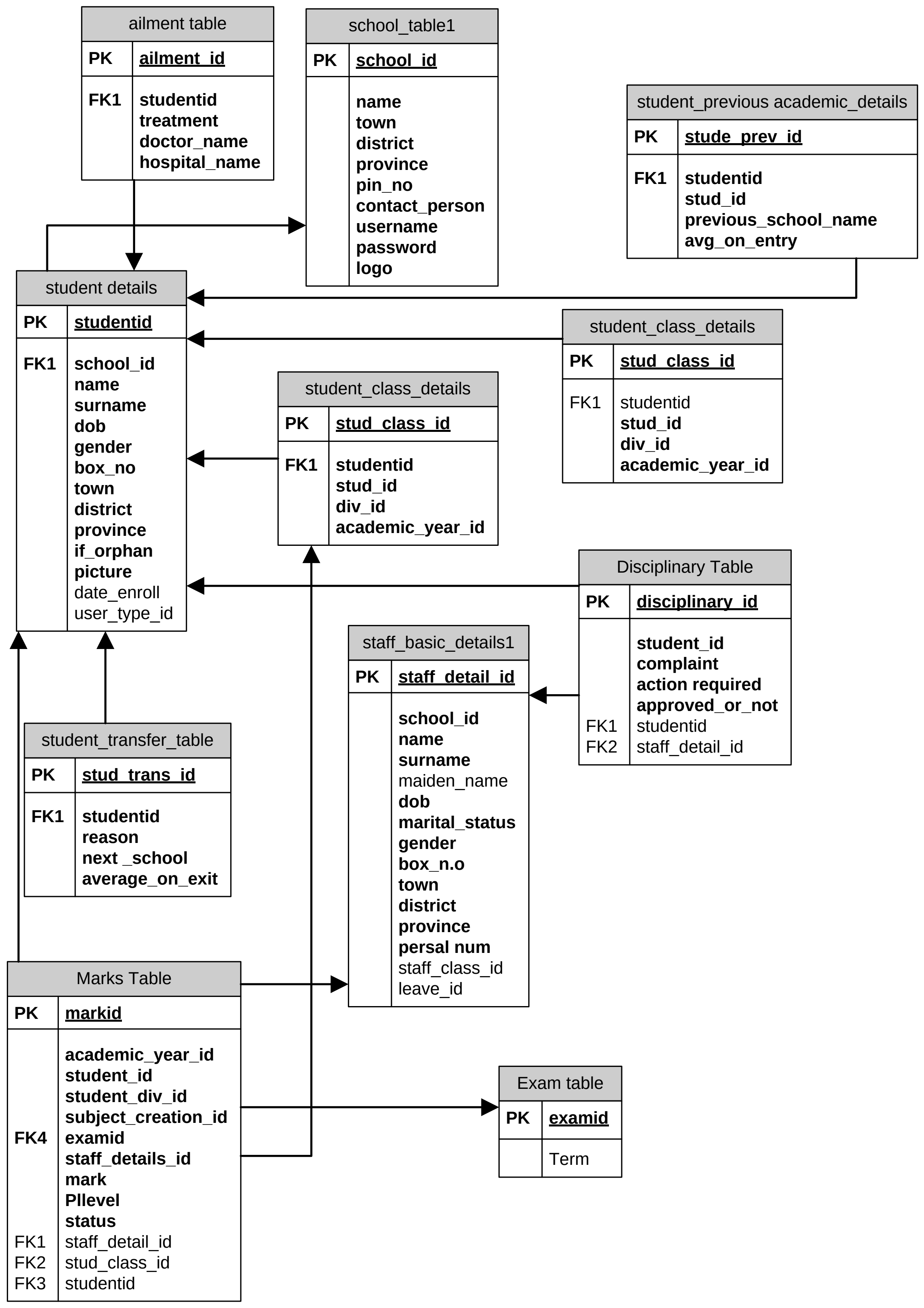


Figure 37 E-R representation of how educator views student information

5.2.4.2 Viewing parent/guardian information

The student_class_details table helps to identify a relevant student in a particular class. By identifying the class and subdivision, the attribute div_id is obtained, and from there the student_id. The list of student_ids will provide the list of students in that class. By selecting a student in that class, the educator can access the details of his/her parents or guardian. Both the parent and guardian tables contain the field studentid, which ensures that only information about the selected student is displayed. A pictorial explanantion of the activities explained is given in figure 38.

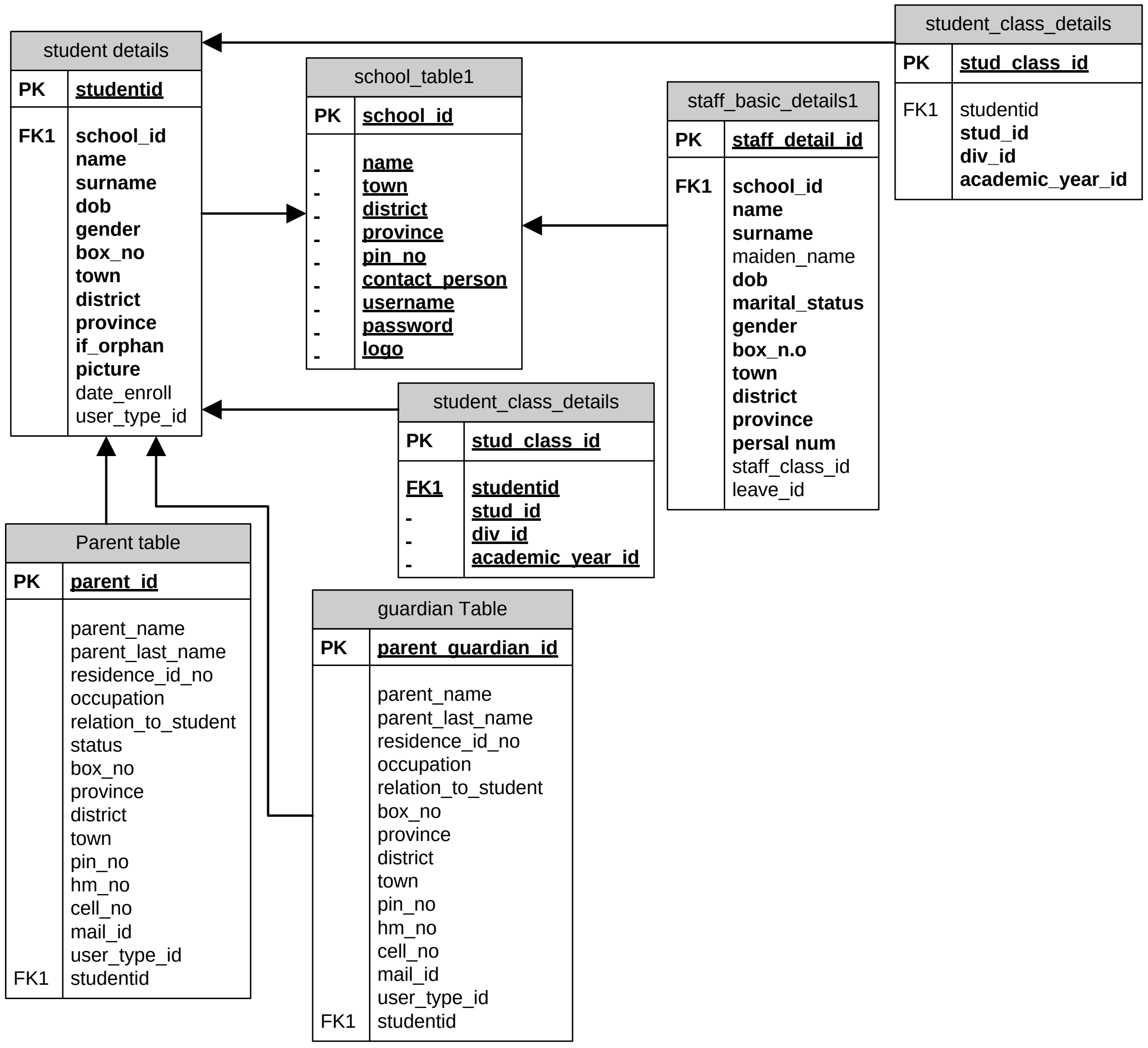


Figure 38 E-R representation of how educator views parent/guardian information

5.2.4.3 Viewing and writing into forums

Educators are able to view the forums created by the principal of the school. All forums created by the principal will be shown. The staff member can write into the forum. His/her replies will be saved in the forum reply table, but will be shown only after the principal has approved them. A pictorial explanantion of the activities explained is given in figure 39.

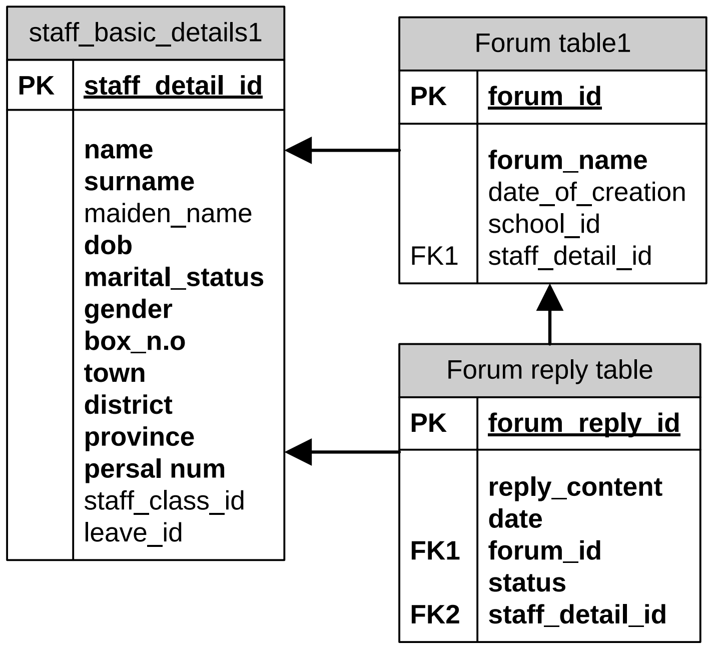


Figure 39 E-R representation of how educator views and write into forums

5.2.4.4 Viewing notification

An educator can view the notifications created for that particular school. During viewing, only notifications of the relevant school are selected, ensuring data isolation of other schools. A pictorial explanantion of the activities explained is given in figure 40.

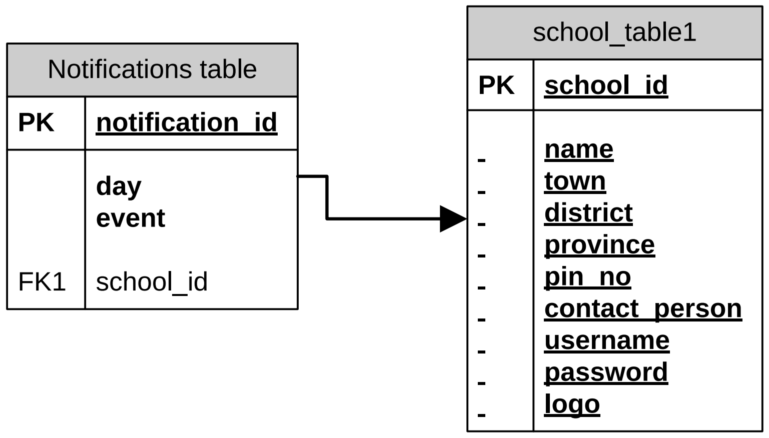


Figure 40 E-R representation of how educator views notifications

5.2.4.5 Uploading notes

An educator can upload notes for a subject that is taught in a particular class. The subject_creation_id ensures that the upload is unique to the relevant class and is not inserted into others classes. The staff_details_id identifies which educator uploaded the notes. A pictorial explanantion of the activities explained is given in figure 41.

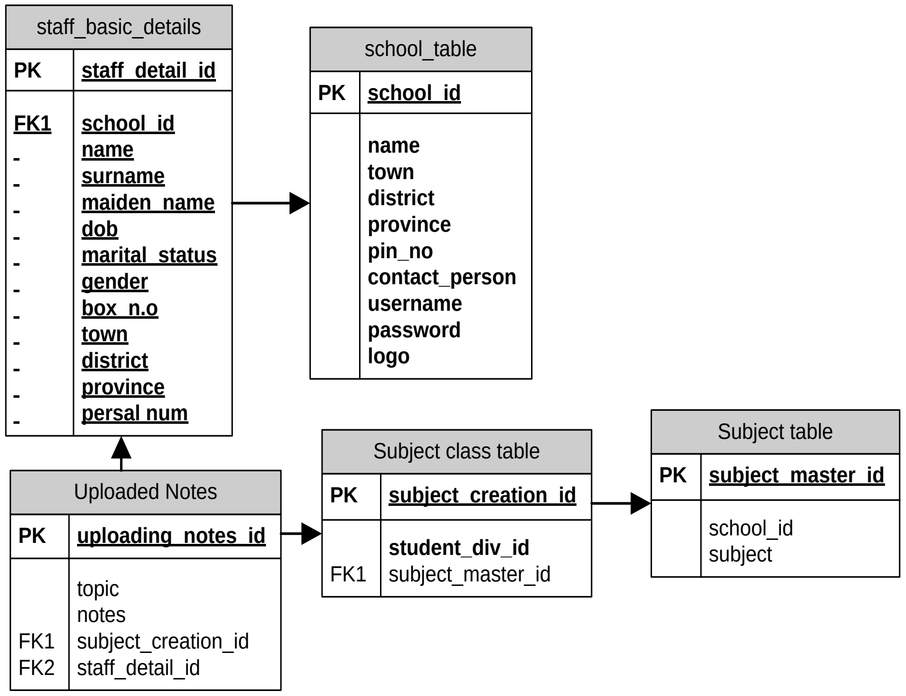


Figure 41 E-R representation of how educator uploads notes

5.2.5 Students

Students are able to view their examination results, uploaded notes and notifications.

5.2.5.1 Viewing results

The marks table captures the marks a student obtained for term-wise examinations. The marks table contains a student_id which helps to identify which student obtained the marks. It also contains staff_details_id identifying the educator who awarded the marks, as well as a status message indicating whether the student has passed or failed. The subject_creation_id identifies the subject taken and, if required, the class in which the student is. Finally, both the staff and the student tables contain the attribute school_id, which ensures that only students and staff of their respective school are able to award and obtain marks. A pictorial explanantion of the activities explained is given in figure 42.

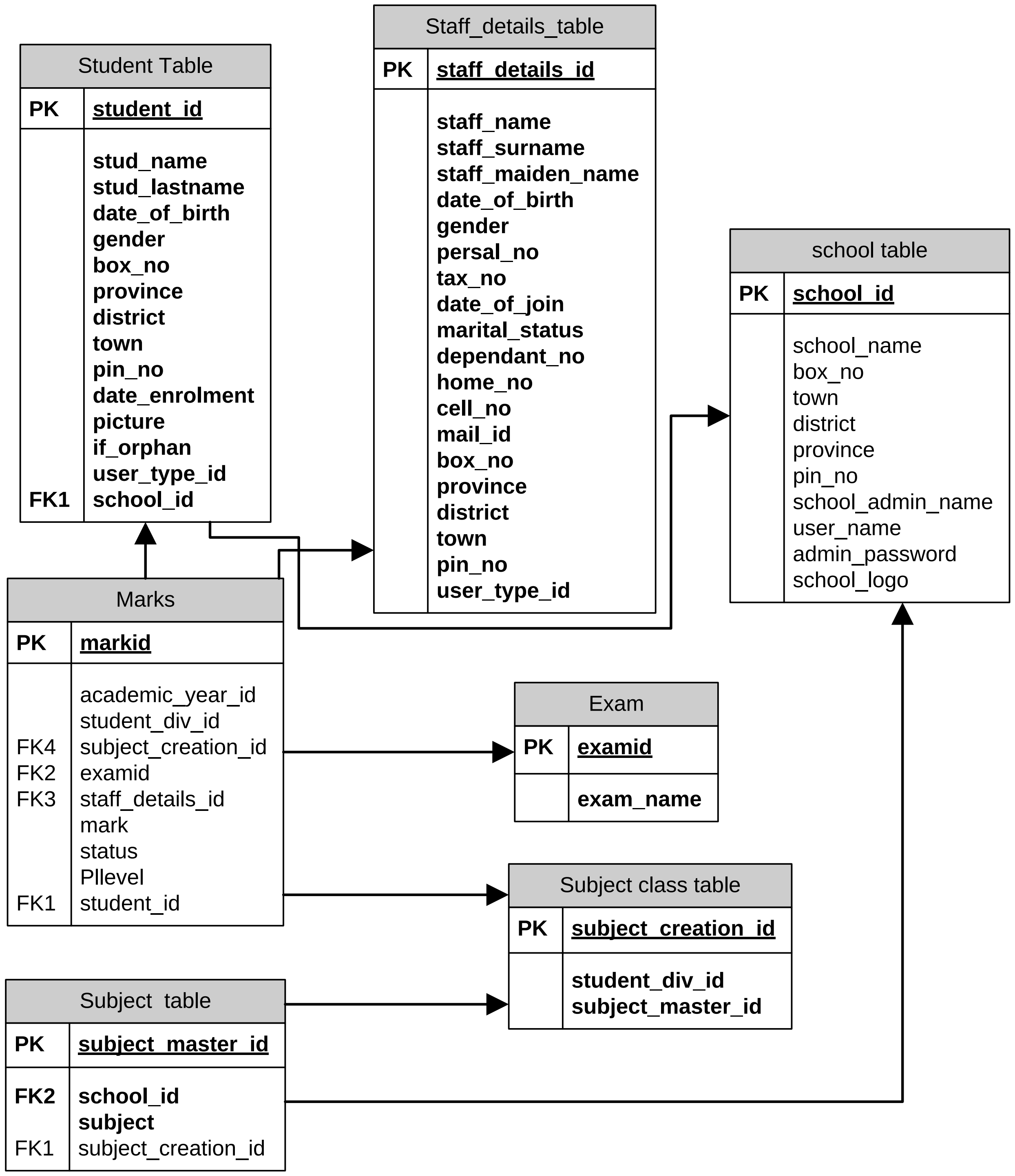


Figure 42 E-R representation of how student views exam result

5.2.5.2Viewing notes

A student is able to view the subject notes that an educator has uploaded. The subject creation id in the uploaded notes table ensures that the notes belong to a particular subject and to a particular class. The staff_details_id identifies the staff member who uploaded the notes. To display the notes, the student needs select only the class to which he/she belongs and the subject. A pictorial explanantion of the activities explained is given in figure 43.

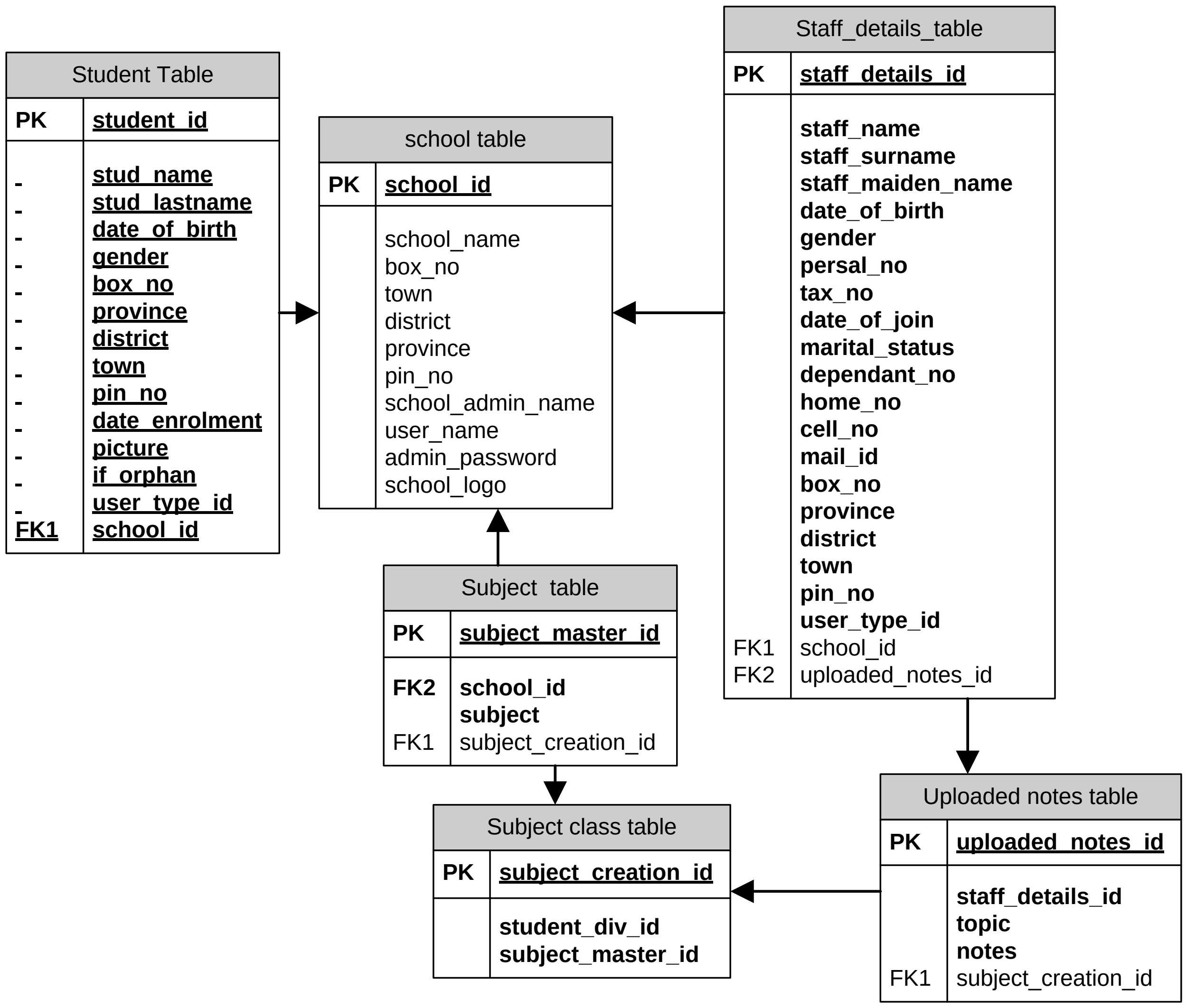


Figure 43 E-R representation of how student views subject notes

5.2.5.3 Viewing notifications

A student can view the notifications created for his/her particular school. During viewing, only notifications of the relevant school are selected in order to ensure data isolation. A pictorial explananation of the activities explained is given in figure 44.

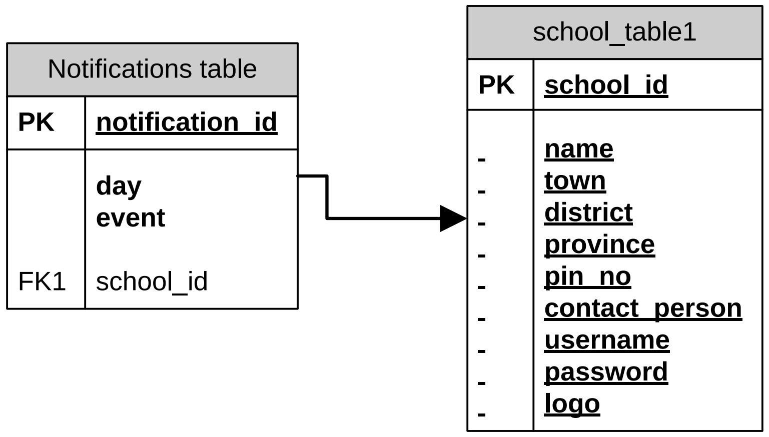


Figure 44 representation of how student views notifications

5.2.6 Parents and guardians

Parents and guardians are able to view their child's marks, disciplinary issues and notifications.

5.2.6.1 Viewing results

Parents can view the marks table. The student_id attribute ensures that the parents can only view their child's results and not those of any other student. Every parent table and guardian table contains the attribute student_id, which identifies the parents' child or children. Thus, when a parent enters the system based on this student_id, the results of that child will be displayed.

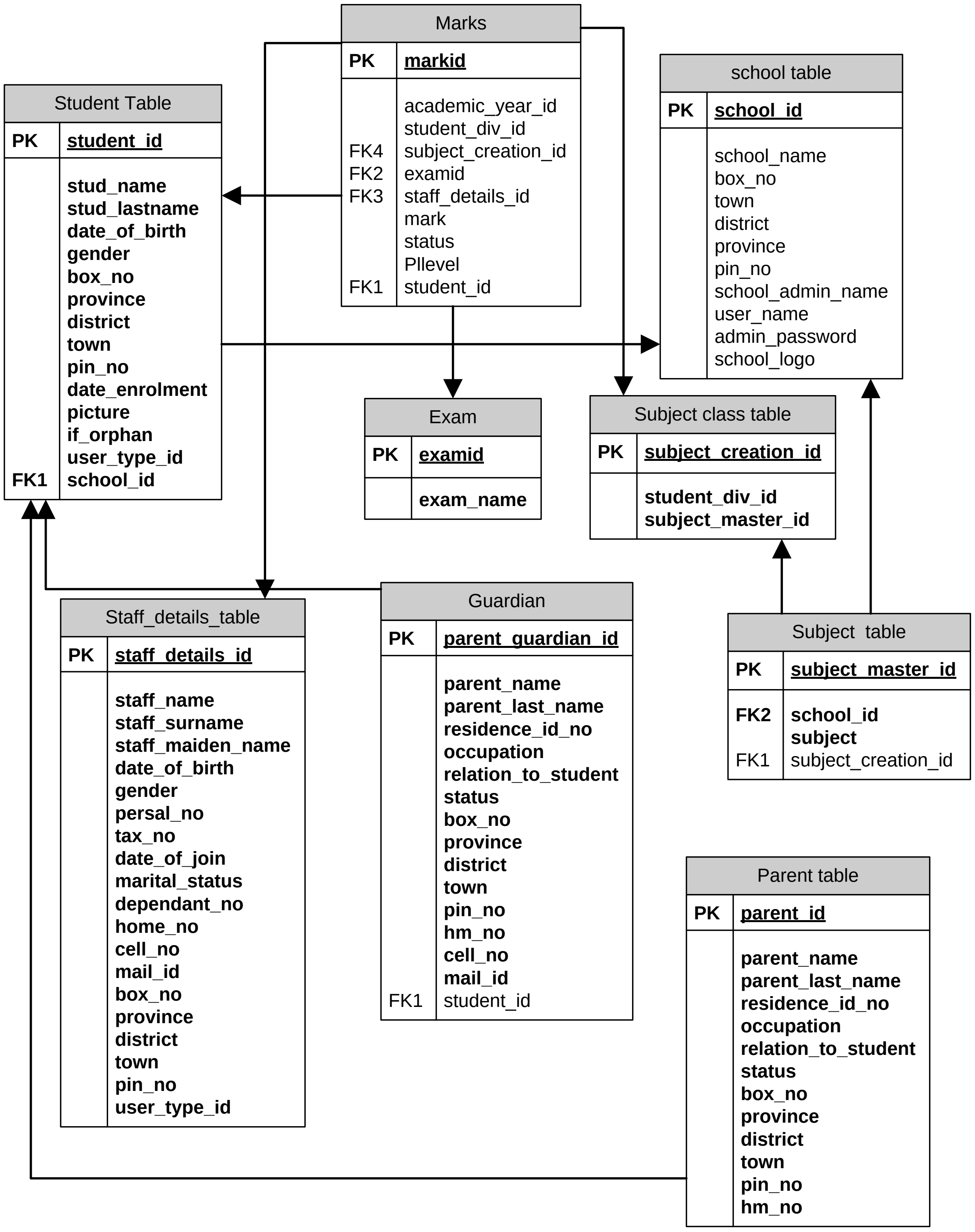


Figure 45 E-R representation of how parents view exam results

A pictorial explanantion of the activities explained is given in figure 45.

5.2.6.2 Viewing disciplinary action

Parents or guardians can log on to the system and on to the disciplinary action section. The disciplinary table, which is the back end of this section, contains the student_id. If this corresponds with the student_id in the parents or guardian tables, they can view the disciplinary measures taken against their child as well as the reasons. A pictorial explanantion of the activities explained is given in figure 46.

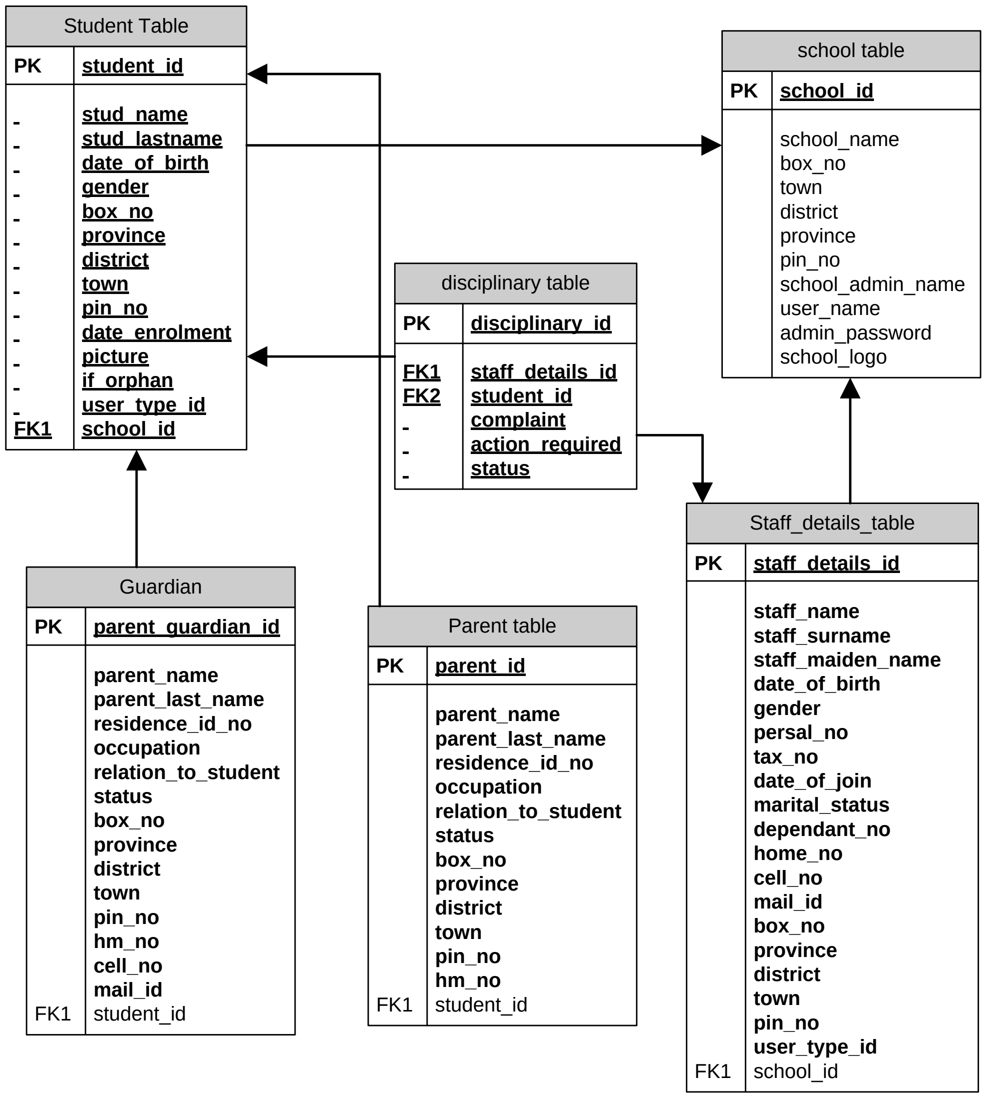


Figure 46 E-R representation of how parents view disciplinary action

5.2.6.3 Viewing notifications

Parents and guardians can view the notifications created for their child's school. During viewing, only notifications of the relevant school are selected. A pictorial explanantion of the activities explained is given in figure 47.

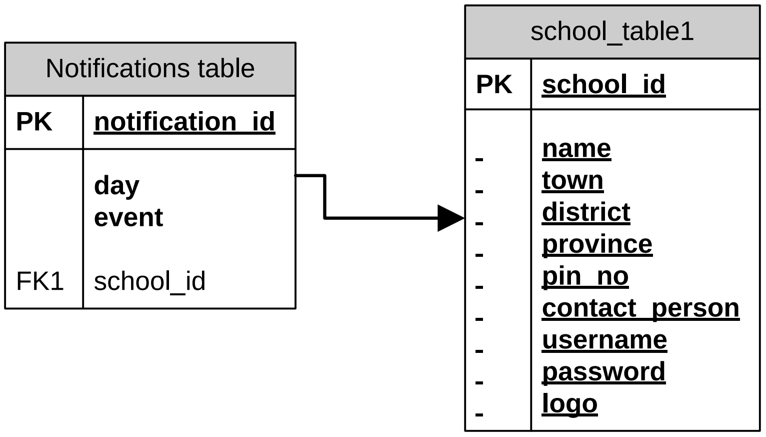


Figure 47 E-R representation of how parents/guardian view notifications

5.3 Summary

Database design is the foundation upon which software is built, A sound database design is mandatory in order to build a resilient and working software. Chapter 4 aims at providing an in-depth view of the database design, the table attributes ad the join queries between tables that help carry out the functions of the software being developed.

Chapter 6: Application User Interfaces

6.1 Explanation of coding

Each module in this software has been programmed in ASP.NET. Three-tier programming has been applied for the development of this project. This type of programming splits the variables used, the methods that manipulate the data, and the connection string into three separate classes. When designing large projects, it allows data to be more organised, and assists with error detection and correction.

The coding used in the connection class will be explained in the next chapter, where the concepts of 'connection' and 'connection object' will been explained in detail. The business layer contains variables, which hold data until they are stored in the database or manipulated to produce the needed result. The data layer contains the methods that execute the function required.

In the previous chapter, it was illustrated how data integrity and data isolation were maintained in the database level. This chapter aims at illustrating how they are maintained in the front end as well.

6.2 Authentication

During authentication, a user is verified and redirected to his/her homepage in accordance with the school he/she belongs to. This is what happens during authentication. Firstly, during authentication, the user name and password is saved into a business layer variable.

```
                b_obj.uname = txtuname.Text;

                b_obj.pwd = txtpwd.Text;
```

b_obj is the business layer object, and uname and pwd signify the variable that holds the user name and password respectively.

```
              DataTable dC = d_obj.loginchek(b_obj);

                 DataTable dB = new DataTable();

If

   (dC.Rows.Count == 0)

       {

       ScriptManager.RegisterStartupScript(Page,

       this.GetType(), "write",


      "alert('UserName and Password Does not exist');", true);

       }

else

       {

           dB = d_obj.checkuserlogin(b_obj);

           Session["schoolid"] = dC.Rows[0][1];

       }
```

Secondly, two data tables are created, dC and dB. dC contains the username and password data from the user creation table, which is compared with the entered username and password. If not found in the user creation table, a message is displayed ‘UserName and Password Does not exist’. If such a username and password do exist, they are compared with a data table dB. dB contains information such as usertypeid, and the schoolid. Usertypeid represents the primary key that determines the usertype. The latter is the type of user that is logging in, for example, staff, student or principal. Schoolid is the primary key of the school to which the user belongs.

```
    Session["usertype"] = dB.Rows[0][1].ToString();
```

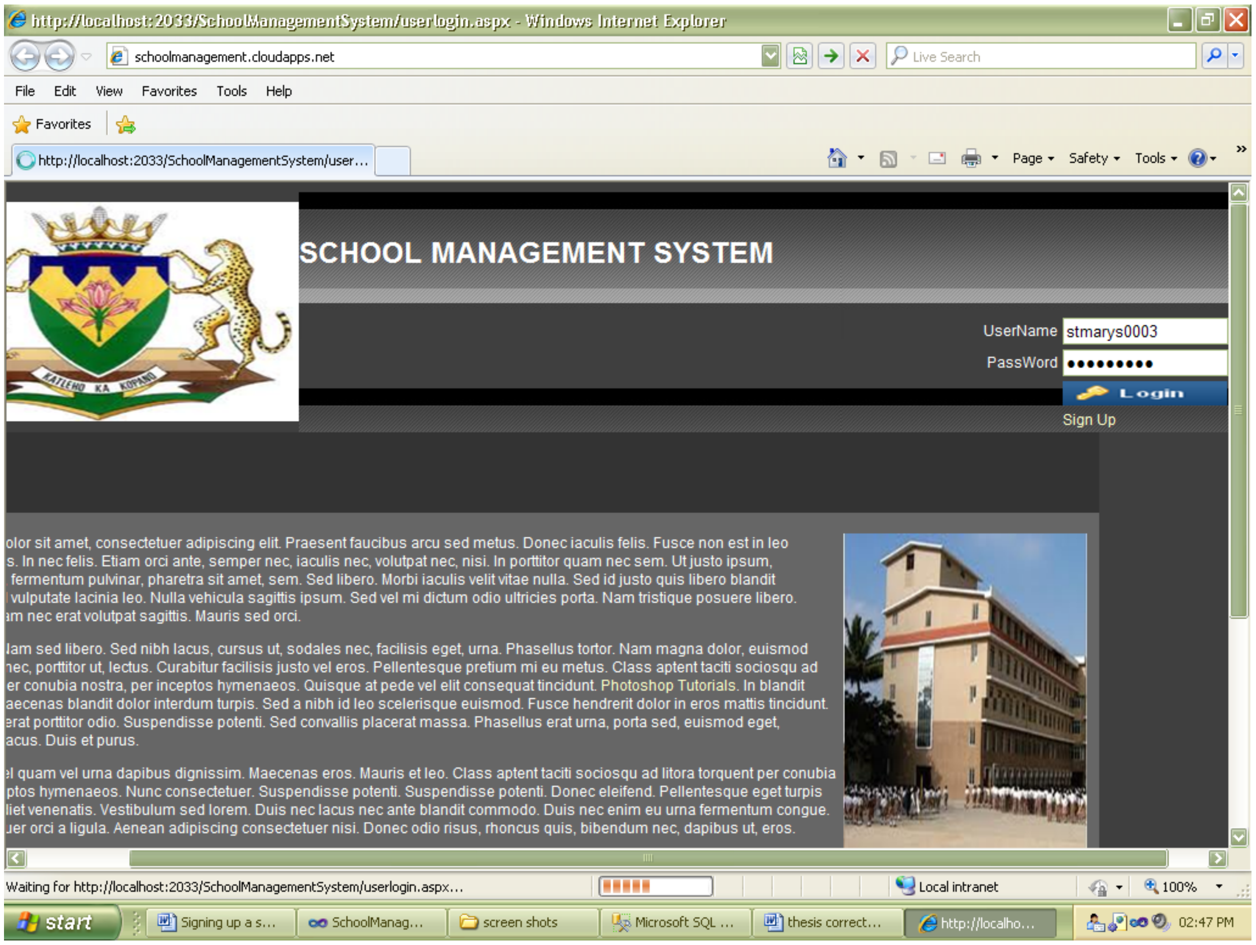
The usertype, which represents the type of user that just logged in, is then saved in another session variable named:

```
          Session["usertype"]
 if
{
(Session["usertype"].ToString() == "SCHOOLADMIN")


Response.Redirect("Schoolhome.aspx");
}
```

```
Session["usertype"]

}

 if

(Session["usertype"].ToString() == "SCHOOLADMIN")

{

 Response.Redirect("Schoolhome.aspx");

}
```

Depending on the usertype, the user is directed to his/her homepage.

Figure 48 is a screen shot of the login page. It is only through the login page that a user enters the system. Figure 49 shows the school admin home page.


http://localhost:2033/SchoolManagementSystem/userlogin.aspx - Windows Internet Explorer
schoolmanagement.cloudapps.net
Live Search
File
Edit
View
Favorites
Tools
Help
Favorites
http://localhost:2033/SchoolManagementSystem/user...
Page
Safety
Tools
SCHOOL MANAGEMENT SYSTEM
Login
Sign Up
Waiting for http://localhost:2033/SchoolManagementSystem/userlogin.aspx...
Local intranet
100%
start
Signing up a s...
SchoolManag...
screen shots
Microsoft SQL ...
thesis correct...
http://localho...
02:47 PM

UserName stmarys0003

PassWord ●●●●●●●●●

olor sit amet, consectetuer adipiscing elit. Praesent faucibus arcu sed metus. Donec iaculis felis. Fusce non est in leo s. In nec felis. Etiam orci ante, semper nec, iaculis nec, volutpat nec, nisi. In porttitor quam nec sem. Ut justo ipsum, fermentum pulvinar, pharetra sit amet, sem. Sed libero. Morbi iaculis velit vitae nulla. Sed id justo quis libero blandit vulputate lacinia leo. Nulla vehicula sagittis ipsum. Sed vel mi dictum odio ultricies porta. Nam tristique posuere libero. am nec erat volutpat sagittis. Mauris sed orci.

lam sed libero. Sed nibh lacus, cursus ut, sodales nec, facilisis eget, urna. Phasellus tortor. Nam magna dolor, euismod nec, porttitor ut, lectus. Curabitur facilisis justo vel eros. Pellentesque pretium mi eu metus. Class aptent taciti sociosqu ad er conubia nostra, per inceptos hymenaeos. Quisque at pede vel elit consequat tincidunt. Photoshop Tutorials. In blandit aecenas blandit dolor interdum turpis. Sed a nibh id leo scelerisque euismod. Fusce hendrerit dolor in eros mattis tincidunt. erat porttitor odio. Suspendisse potenti. Sed convallis placerat massa. Phasellus erat urna, porta sed, euismod eget, acus. Duis et purus.

el quam vel urna dapibus dignissim. Maecenas eros. Mauris et leo. Class aptent taciti sociosqu ad litora torquent per conubia ptos hymenaeos. Nunc consectetuer. Suspendisse potenti. Suspendisse potenti. Donec eleifend. Pellentesque eget turpis liet venenatis. Vestibulum sed lorem. Duis nec lacus nec ante blandit commodo. Duis nec enim eu urna fermentum congue. uer orci a ligula. Aenean adipiscing consectetuer nisi. Donec odio risus, rhoncus quis, bibendum nec, dapibus ut, eros.

Figure 48 Login

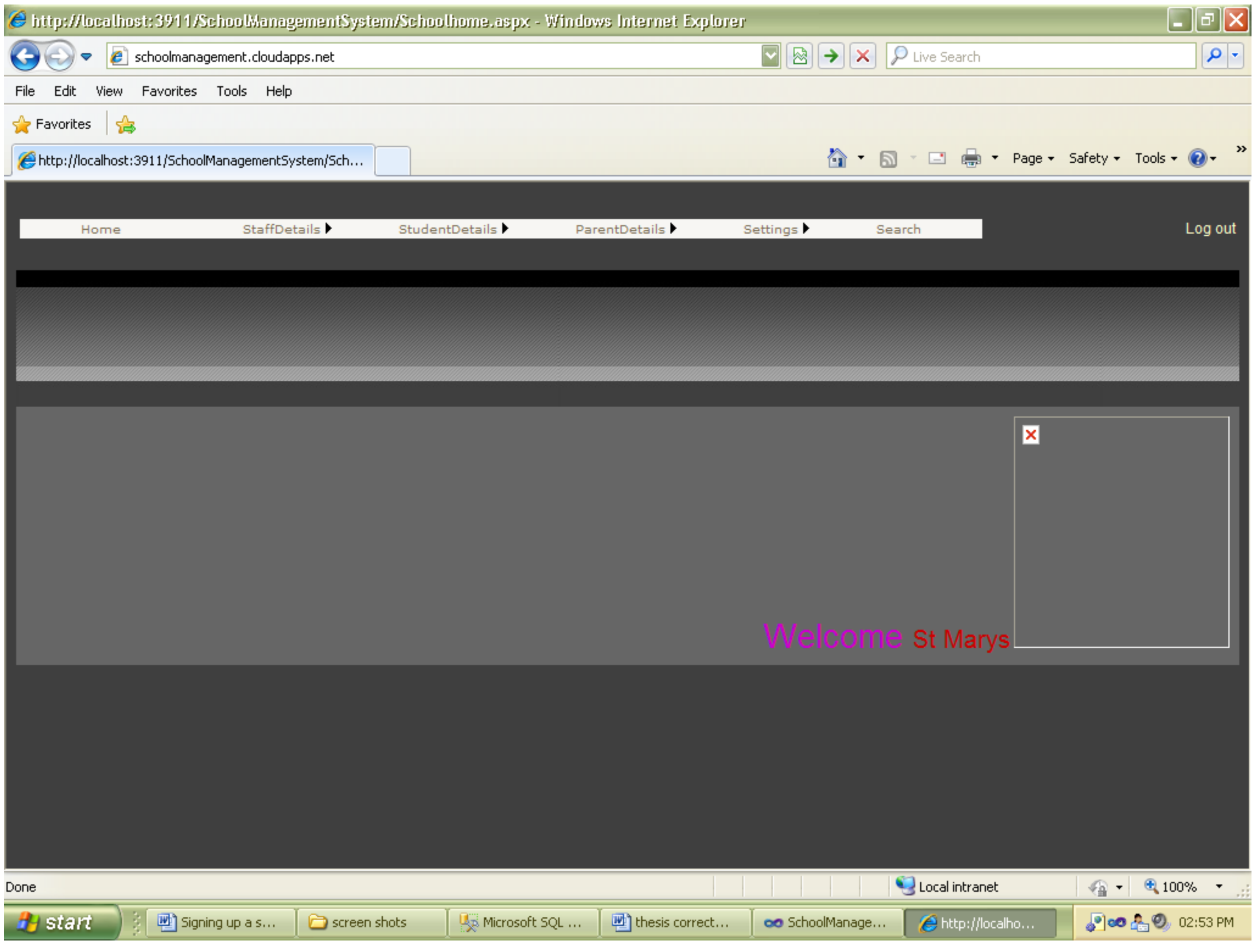

http://localhost:3911/SchoolManagementSystem/Schoolhome.aspx - Windows Internet Explorer
schoolmanagement.cloudapps.net
Home
StaffDetails
StudentDetails
ParentDetails
Settings
Search
Log out
Welcome St Marys

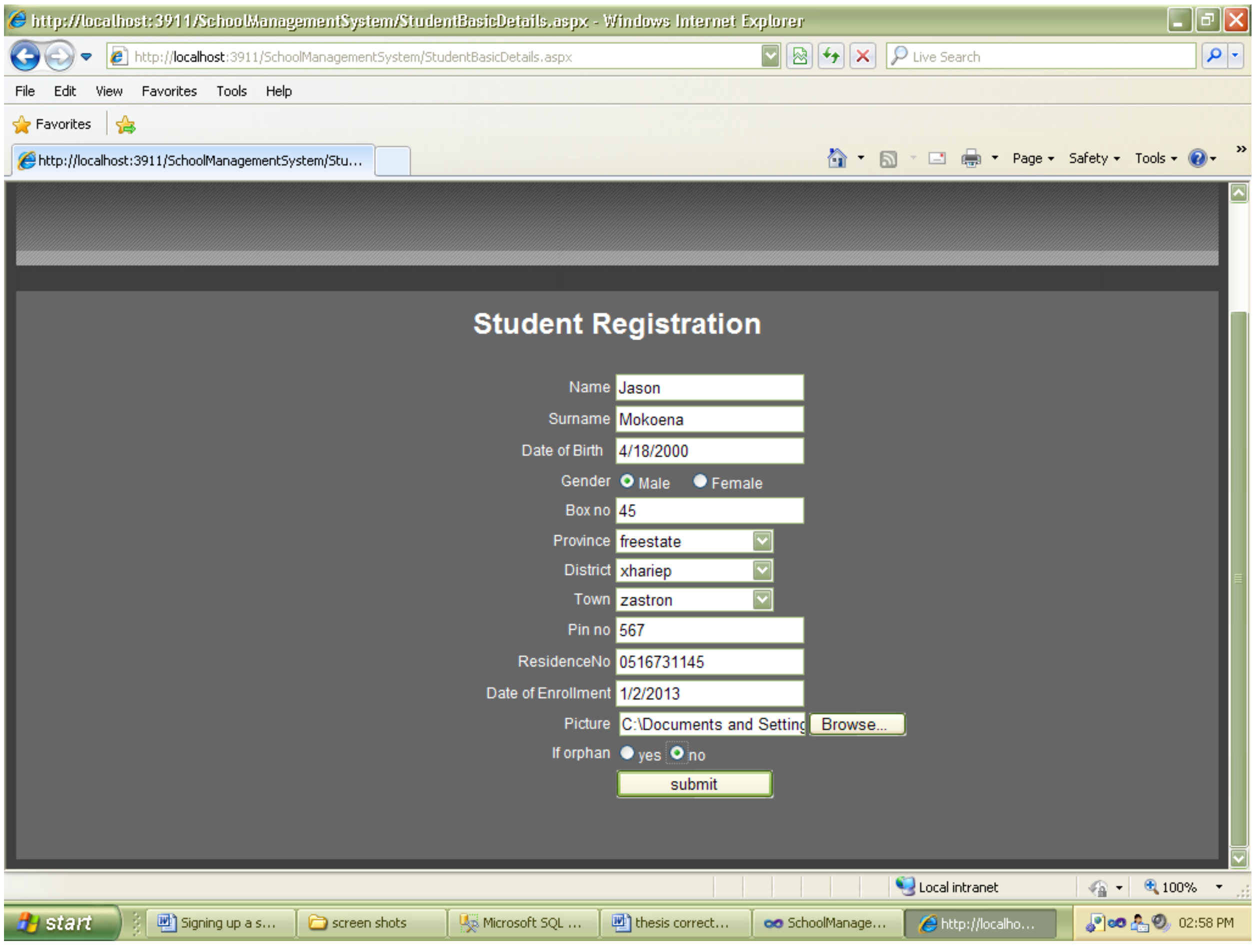
Figure 49 School admin home page

6.3 Information querying

Information querying refers to the entering, editing and deletion of data. It is important that, during the querying of data, data are isolated and data integrity is maintained. Data isolation means that the data are stored under the correct user, and school and data integrity means that no information from other schools should be viewed during data retrieval.

6.3.1 Entering information

When entering information, for example, a student's personal details, one has to ensure that the data are stored under the school to which the student belongs. This is accomplished by the school admin entering the admin homepage first, and then entering the relevant data and clicking submit. Figure 50 illustrates how information is collected.


http://localhost:3911/SchoolManagementSystem/StudentBasicDetails.aspx - Windows Internet Explorer
http://localhost:3911/SchoolManagementSystem/StudentBasicDetails.aspx
Live Search
File Edit View Favorites Tools Help
Favorites
Page Safety Tools
Local intranet
100%
start
Signing up a s...
screen shots
Microsoft SQL ...
thesis correct...
SchoolManage...
http://localho...
02:58 PM

Student Registration

Name	Jason
Surname	Mokoena
Date of Birth	4/18/2000
Gender	[✓ Male] Female
Box no	45
Province	freestate
District	xhariep
Town	zastron
Pin no	567
ResidenceNo	0516731145
Date of Enrollment	1/2/2013
Picture	C:\Documents and Setting Browse...
If orphan	yes [✓ no]
	submit

Figure 50 Screen shot of student registration

Variables for each field in the above form have been created in the business layer. A method to insert these variables into the database has been created in the data layer as well.

```
 protected void sumbitstuddetail_Click(object sender, EventArgs
e)
   {
     b_obj.School_id = Convert.ToInt16(Session["schoolid "]);
        byte[] img = fileupload1.FileBytes;
        Session["imag"] = img;
        b_obj.School_id = Convert.ToInt16(Session["s_id"]);
        b_obj.student_name = txtstudentname.Text;
        b_obj.student_lastname = txt_surname.Text;
        b_obj.student_dob = txtdob.Text;

   if (rbtmale.Checked == true)
        {
            b_obj.student_gender = rbtmale.Text;
        }
   else
        {
            b_obj.student_gender = rbtfemale.Text;
        }
        b_obj.student_boxno = Convert.ToInt32(txt_box.Text);
             b_obj.student_province =
     Convert.ToInt32(ddlprovince.SelectedValue);
```

```
                  b_obj.student_district =
     Convert.ToInt32(ddldistrict.SelectedValue);
              b_obj.student_town =
     Convert.ToInt32(ddltown.SelectedValue);
         b_obj.student_pinno = Convert.ToInt32(txt_pinno.Text);
              b_obj.student_residenceno =
     Convert.ToInt32(txtresidence.Text);
         b_obj.student_dateofenrollment =
txt_dateofenrollment.Text;
         b_obj.student_image = img;
         if (rbtorphanyes.Checked == true)
         {
             b_obj.student_iforphan = rbtorphanyes.Text;
         }
         else
         {
             b_obj.student_iforphan = rbtorphano.Text;
         }

         b_obj.user_typeid = Convert.ToInt32 (3);
         Session["usertypeid"] = b_obj.user_typeid;
         object studid = new object();
```

```
    studid =d_obj.insert_studentregistrationdetails(b_obj);
             Session["studentid"] = studid    ;
    ScriptManager.RegisterStartupScript(Page, this.GetType(),
    "write", "alert('Student Registered Successfully');",
    true);
       Panel1.Visible = true;
       txtusername.Text  = admissionno;
      }
```

txt_dateofenrollment.Text is the name of the text box that contains the date of enrolment.

```
b_obj.student_dateofenrollment = txt_dateofenrollment.Text;
```

The above line of code assigns the value in the text box to a business layer variable. Similarly, the value in each of the text boxes is assigned to its corresponding business layer variable.

```
b_obj.School_id = Convert.ToInt16(Session["schoolid "]);
```

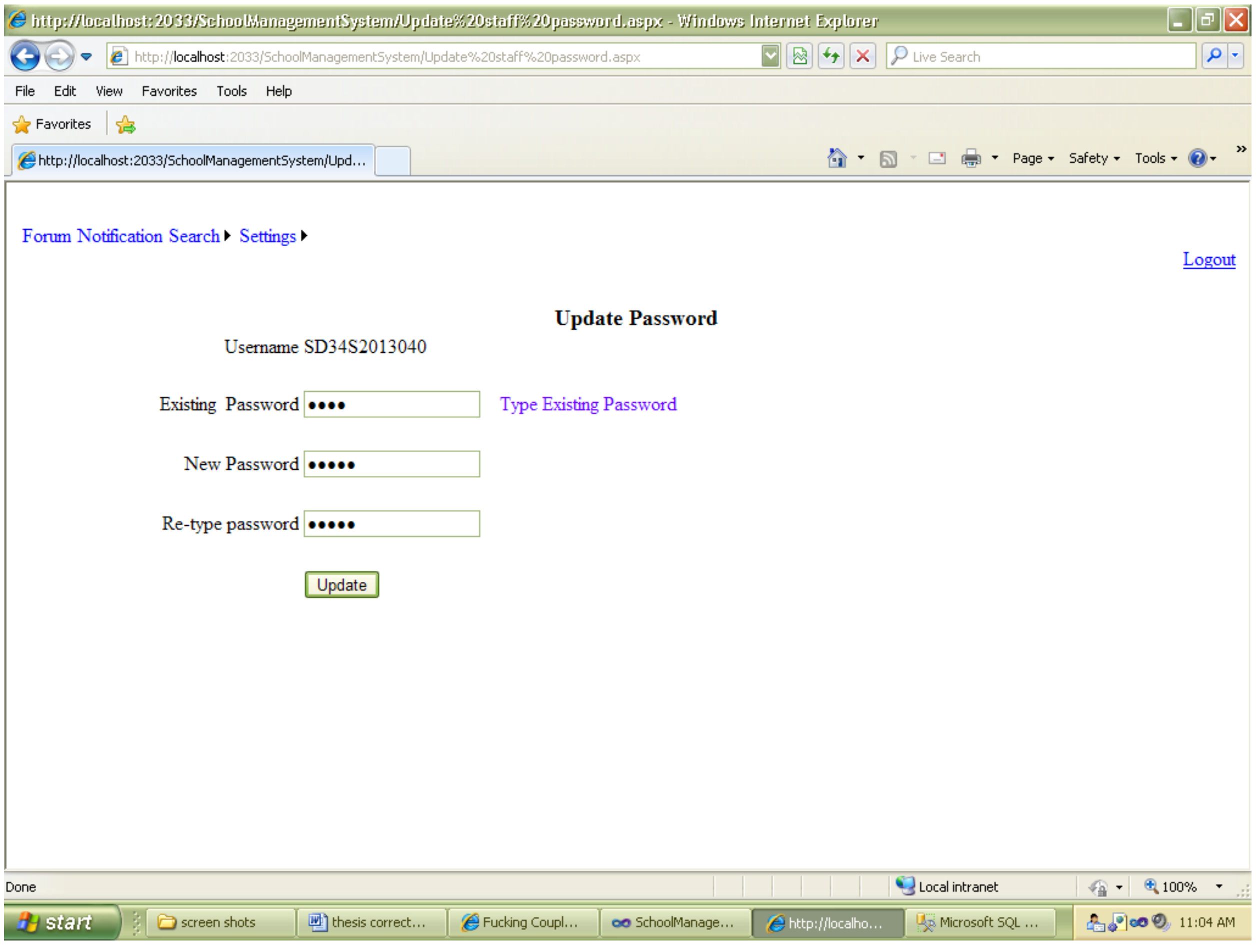
The value of the school primary key id is saved inside a session variable called **Session["schoolid "])**. This session variable is converted into an integer value and assigned to the business layer variable. This value is stored in the student details table and helps to distinguish students in different schools.

The above line of code will invoke the data layer method that will insert all the required information into the database.

```
d_obj.insert_studentregistrationdetails(b_obj);
```

6.3.2 Updating existing information

During the editing of information, it is mandatory that information belonging only to the correct user is edited and no other information. In the following example, changing the password of a staff member is illustrated. The illustrated code below will explain how only the staff member’s password is edited and no information of other users is affected. Figure 51 is a screen shot of the password updation page


http://localhost:2033/SchoolManagementSystem/Update%20staff%20password.aspx - Windows Internet Explorer
File Edit View Favorites Tools Help
Forum Notification Search Settings
Logout

Update Password

Username SD34S2013040

Existing Password ●●●● Type Existing Password

New Password ●●●●●

Re-type password ●●●●●

Update
Done
Local intranet
100%
start
screen shots
thesis correct...
Fucking Coupl...
SchoolManage...
http://localho...
Microsoft SQL ...
11:04 AM

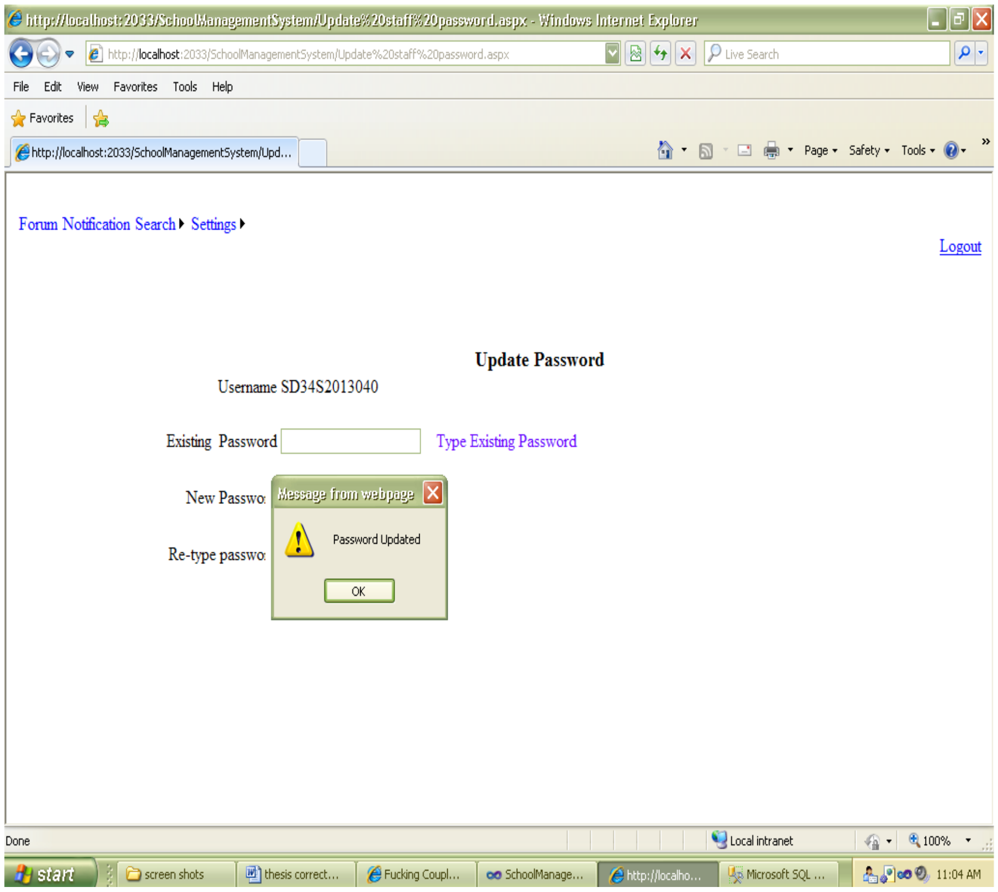
Figure 51 Updation of information

```
b_obj.user_username = Label3.Text;

b_obj.user_password =  TextBox1.Text;

b_obj.pwd =TextBox3.Text;


d_obj.updatestaffpassword(b_obj);


ScriptManager.RegisterStartupScript(Page,
this.GetType(), "write", "alert('Password Updated');", true);
```

The user creation table contains the login details of each user. Every user has a unique user name and password combination. Because of this unique combination, data integrity is maintained during updation. The query that is used to update the password is as follows.

```
UPDATE UserLogin SET UserLogin.Password=@pwd

WHERE  UserLogin.Username=@username and
UserLogin.Password=@password.
```

The new password is received from the text box. It is assigned to business layer variables, after which the data layer method is invoked. Figure 52 is a screen shot of after a password has been updated successfully.


http://localhost:2033/SchoolManagementSystem/Update%20staff%20password.aspx - Windows Internet Explorer
File Edit View Favorites Tools Help
Forum Notification Search Settings
Logout

Update Password

Username SD34S2013040

Existing Password [ ] Type Existing Password

New Passwo

Re-type passwo

Message from webpage

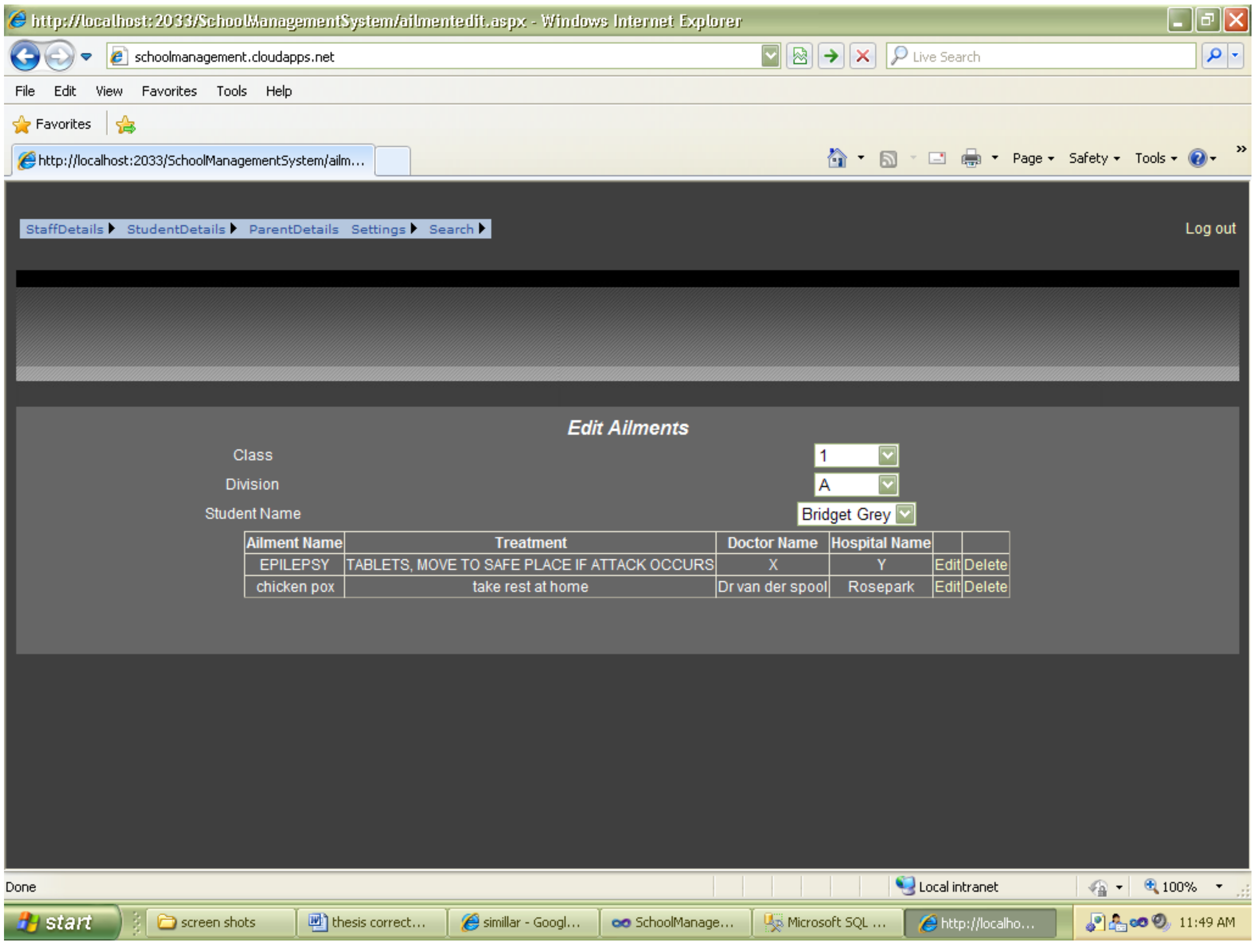
Password Updated

OK

Figure 52 Updated staff password

6.3.3 Deleting existing information

During deletion, it is important that only a single row is deleted and not the entire table. The following illustrates the example of deleting an ailment entry. Figure 53 shows the ailment records before deletion


http://localhost:2033/SchoolManagementSystem/ailmentedit.aspx - Windows Internet Explorer
StaffDetails
StudentDetails
ParentDetails
Settings
Search
Log out

Edit Ailments

Class	1
Division	A
Student Name	Bridget Grey

Ailment Name	Treatment	Doctor Name	Hospital Name		
EPILEPSY	TABLETS, MOVE TO SAFE PLACE IF ATTACK OCCURS	X	Y	Edit	Delete
chicken pox	take rest at home	Dr van der spool	Rosepark	Edit	Delete

Figure 53 Ailment table before deleting entry

The medical records of this particular student are contained in a grid view. The primary key of these rows is also held in the grid. For an aesthetic and meaningful view, their viewing of these rows is disabled.

```
int id =
Convert.ToInt32(GridView1.DataKeys[e.RowIndex].Values[0]);
```

```
         b_obj.AILMENT_ID = id;

        d_obj.deleteailment(b_obj);


        ScriptManager.RegisterStartupScript(Page,
this.GetType(), "write", "alert('Deleted');", true);


        b_obj.student_id =
Convert.ToInt32(ddstud_name.SelectedValue);

        DataTable dm = d_obj.ailment_validatee(b_obj);

        GridView1.DataSource = dm;

        GridView1.DataBind();
```

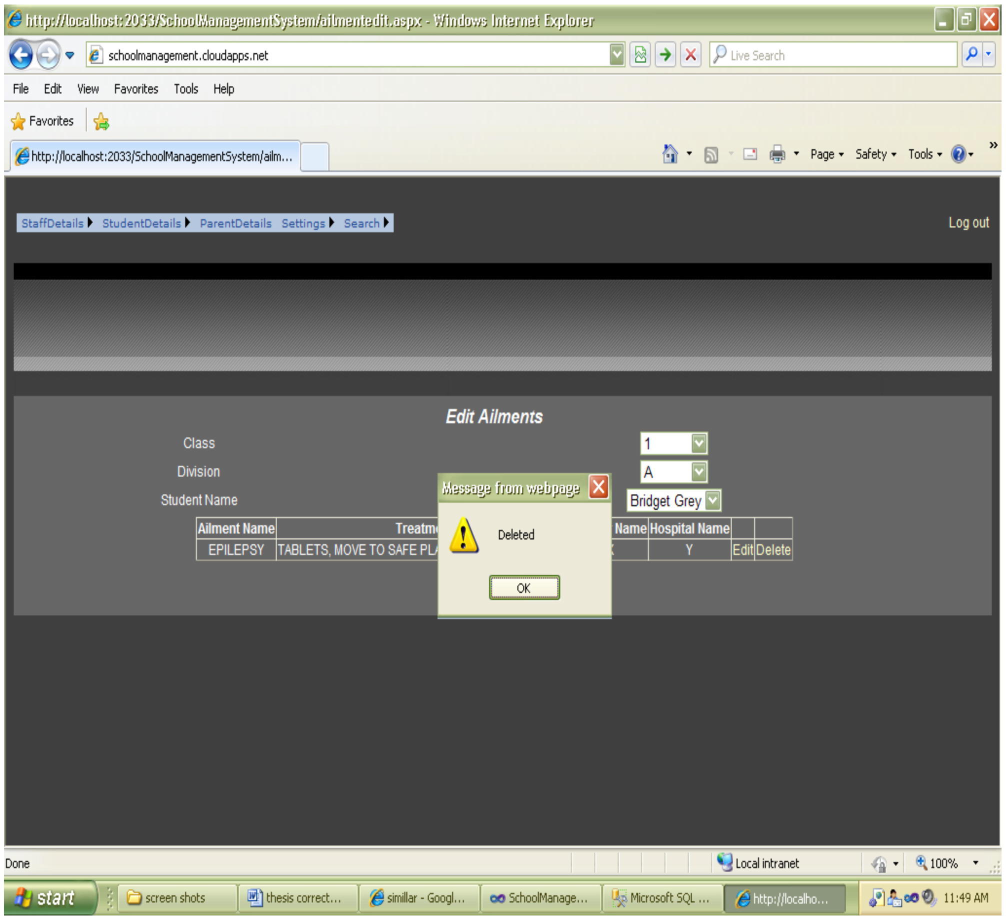
The primary key is held at the zeroth position of the row. The following code helps obtain the primary key:

```
int id =
Convert.ToInt32(GridView1.DataKeys[e.RowIndex].Values[0]);
```

Once the primary key has been obtained, it is assigned to the business layer variable. Subsequently, the data layer method is invoked and deletion is carried out. After deletion, the page is reloaded to show the updated table entry, as indicated by the code below:

```
obj.student_id = Convert.ToInt32(ddstud_name.SelectedValue);
      DataTable dm = d_obj.ailment_validatee(b_obj);
      GridView1.DataSource = dm;
      GridView1.DataBind();
```

Figure 54 show the ailment records table after deletion of an entry.


http://localhost:2033/SchoolManagementSystem/ailmentedit.aspx - Windows Internet Explorer
schoolmanagement.cloudapps.net
File Edit View Favorites Tools Help
StaffDetails StudentDetails ParentDetails Settings Search
Log out
Message from webpage
OK
Done
Local intranet
start
11:49 AM

Edit Ailments

Class	1
Division	A
Student Name	Bridget Grey

Deleted

Ailment Name	Treatm	Name	Hospital Name		
EPILEPSY	TABLETS, MOVE TO SAFE PL		Y	Edit	Delete

Figure 54 Ailment records after deletion

6.4 Summary

Chapter 6 is an explanation of the coding behind the web forms that users access, how database operations are carried out by codings. How entries are added, deleted and updated through the codings written.

Chapter 7: Overview of Development Tools and Technologies used

7.1 Asp.Net

ASP.NET is a programming language developed by Microsoft in 2002, which allows programmers to develop dynamic web pages [11]. With the common language runtime (CLR) as its foundation, ASP.NET can be supported by any .NET languages.

7.1.1 Characteristics

Some of the characteristics of an ASP.Net page are as follows

7.1.1.1 Directives

Directives are instructions to a page parser on how an ASP.NET is to process a page [11].

7.1.1.2 User controls

Certain sections of pages are encapsulated; these are then registered and used as controls in ASP.NET [11].

7.1.1.3 Custom controls

A programmer can build his/her own controls to manipulate the software they are creating using cutom controls; the .ASCX markup file is not available to the programmer, unlike the user controls. All coding created by the user is compiled into the dynamic link library (DLL). These controls can be used with multiple web applications and visual studio projects.

7.1.1.4 Rendering techniques

During compilation, the template (.aspx) file is compiled into initialisation code, which builds a control tree (the composite) representing the original template. Literal text goes into instances of the literal control class, and server controls are represented by instances of a specific control class. The initialisation code is combined with user-written code (usually by the assembly of multiple partial classes) and results in a class that is specific for the page. The page doubles as the root of the control tree [11].

7.1.1.5 State management

ASP.NET applications are hosted by a web server and are accessed using the stateless HTTP protocol. As such, if an application uses stateful interaction, it has to implement state management on its own. ASP.NET provides various functions for state management. Conceptually, Microsoft treats ‘state’ as GUI state. Problems may arise if an application needs to keep track of ‘data state’, for example, a finite-state machine which may be in a transient state between requests (lazy evaluation) or which takes a long time to initialise. State management in ASP.NET pages with authentication can make web scraping difficult or impossible [11].

7.1.1.6 Session state

Server-side session state is held by a collection of user-defined session variables that are persistent during a user session. These variables, accessed through the session collection, are unique to each session instance. The variables can be set to be automatically destroyed after a defined time of inactivity even if the session does not end. Client-side user session is maintained by either a cookie or by encoding the session ID in the URL itself [11].

ASP.NET supports three modes of persistence for server-side session variables [15]:

7.1.1.7 SQL server mode

State variables are stored in a database, allowing session variables to be persisted across ASP.NET process shutdowns. The main advantage of this mode is that it allows the application to balance load on a server cluster, sharing sessions between servers. This is the slowest method of session state management in ASP.NET [11].

7.1.1.8 Application

Applications are set and initialised when the Application_OnStart event fires on the loading of the first instance of the application and are available until the last instance exits. Application state variables are accessed using the applications collection, which provides a wrapper for the application state. Application state variables are identified by name [11].

7.1.1.9 View state

ASP.NET applications emit HTML pages. HTML utilizes a state management mechanism known as view state. View state is a page level state management mechanism that is used to control and manage web form controls and widgets. The state of the controls is encoded and sent to the server at every form submission in a hidden field known as __VIEWSTATE. The server then sends back the variable so as to re-render the controls render at their last state.

At the server side the application may change the viewstate, if the processing requires a change of state of any control. The viewstate collection makes the states of individual controls decoded at the server available for use in the ASP.NET pages.

7.1.1.10 Server-side caching

ASP.NET offers a 'Cache' object that is shared across the application and can also be used to store various objects. The 'Cache' object holds the data for a specified amount of time only and is automatically cleaned after the session time-limit elapses [11].

7.1.2 Net framework

Before the invention of the framework, programmers saved codes that would be of use in the future. When the concept of a framework was being developed, it was agreed that reused codes would be saved in a library and accessed when needed. This also further implemented the reusability paradigm of object-oriented analysis and design.
The framework performs memory management, code execution, etc., and that is why .NET applications are called 'managed applications'.

A .NET application or service always works within the .NET framework. A .NET application does not access the operating system directly, instead it first accesses the .NET framework, and the .NET framework, in turn, accesses the operating system and hardware. The framework consists of two parts, namely Common language runtime and the .NET framework class library.

7.1.3 Advantages of using ASP.NET

ASP.NET is a language that has been created specifically for the purpose of web application development while, at the same time, enabling developers to do so with ease and comfort. ASP.NET improves performance by utilising the strengths of JIT compilation, automatic resource optimisation, runtime profiling, automatic memory management, early binding, better caching and exception handling. It allows dragging and dropping of controls; thus, making programming much simpler. [12]

ASP applications are actually an HTML pages contained within tags. Running an ASP pages requires it to be placed in a directory on a server and the user to be authenticated and authorised to run it. Programming in ASP.NET allows flexibility. A novice programmer can use VB script and Jscript to program in ASP.NET. ASP.NET also contains modules to support those who program in python and Perl [12].

7.1.4 Disadvantages of ASP.NET

Even though often referred to as the easiest way to build a web application, ASP.NET does have some disadvantages. In order to run an ASP.NET page, Internet information services, or IIS, will have to be installed. MS SQL will have to be installed in order to create databases. Many of the in-built features, such as those found in PHP, are not found in ASP.NET.

ASP is based on the component object model; this is an overhead to the server. It is only ompatible on a Windows platform, whereas PHP has more compatibility.
Prior VB knowledge is an advantage to those who wish to program in ASP.NET, since much of the ASP.NET coding is similar to VB.NET.

7.2 MySQL

MySQL is the world's most used open source relational database management system. The term MySQL was coined by co-founder of MYSQL AB company Michael Widenuis, and consists of 'My', his daughter's name, and SQL, which stands for structured query language.

MySQL is popular among many web-based technologies such as Linux, PHP and PERL [17], and many popular websites such as Facebook, Twitter, Flickr, Wikipedia and YouTube have used it as their database.

7.2.1 SQL Server Manangement Studio

For the development of the system explained in this thesis, the SQL server management studio was used.

The object explorer in the SQL server management studion allows users to select, browse and edit objects within the server [24]. SQL server management has been written in the

7.3 ADO.NET

ADO stands for access data object. It is a part of the .NET framework that consists of a set of classes which are used to handle data access. ADO.NET is based fully on the Xtra Markup Language (XML), but differs from ADO in that it has no recordset object.

In order to connect to a database in ASP.NET the following namespaces (header files) are required:

```
using System.Data;

using System.Data.SqlClient;
```

In order to establish a connection with SQL Server, the SqlConnection object is used:

```
SqlConnection sqlConn = new SqlConnection(connectionString)

                      sqlConn.Open();
                      sqlConn.Close();
```

In order to reduce overhead, once a connection is opened, it is left open for the entire process. This process is referred to as connection pooling. Pooling database connections can significantly enhance the performance and scalability of an application. In order to take maximum advantage of these functions, one should note the following [13]:

- Connections are pooled only if they have the same connection string; so ensure this is always constant.
- When finished with an SqlConnection object, call Dispose() or Close() to release the connection back to the pool.
- In order to keep the maximum number of connections available, keep connections open for as short a period as possible – remembering that, due to connection pooling, re-opening a database connection will incur little overhead.

7.4 Summary

Chapter 7 aimed to familiarise the reader with the technologies used in the development of cost-effective school management software. ASP.NET is by far one of the easiest technologies in which to develop a web application. Any user who is familiar with object-oriented programming can easily program in ASP.NET. Using ADO.NET helps to establish a link between the database and coding, meaning that ADO.NET aids in connecting ASP.NET to the SQL server. Using SQL as the back end of the application, is advantageous because SQL is compatible with almost all types of platforms. The ease of programming with ASP.NET and the compatibility it shares with SQL were the reasons why these technologies were selected for the development of the school management system in this study.

Chapter 8: Cloud Deployment

8.1 Cloud Deployment Platforms

Once the application was ready, it needs to be deployed on the cloud, from where it will be made available to multiple schools. As illustrated earlier, schools access the system by subscribing to the cloud based SAMs through a simple registration form. Currently, there are several public cloud platforms in existence including the following

- MS Azure
- Oracle public cloud
- Amazon Web Services
- Rackspace
- Google App Engine

The method of moving a developed application onto a cloud platform is called migration. Migration of an application onto cloud will be explained later in this section. First, a brief description of the various cloud platforms is presented.

8.2 MS Azure

MS Azure is a cloud computing platform provided by Microsoft, which aids in the development, deployment and management of cloud applications. Azure allows the development of applications and also makes it possible for developers to integrate their existing IT environment into public cloud applications. Various data centres that hold databases and application code are placed around the world.

8.2.1 Implementation

The Windows Azure services use a specialised operating system known as the Windows Azure to run its fabric layer. A cluster hosted at Microsoft's data centres manages computing and storage resources of the computers and provisions the resources (or a subset of them) to applications running on top of Windows Azure. Windows Azure has been described as a 'cloud layer' on top of a number of Windows Server systems, which uses Windows Server 2008 and a customised version of Hyper-V, known as the Windows Azure Hypervisor, to provide virtualisation of services. Scaling and reliability are controlled by the Windows Azure Fabric Controller, so the services and environment do not crash if one of the servers crashes within the Microsoft data centre. It also provides the management of the user's web application, for instance memory resources and load balancing [2].

Azure has data centres in the following regions:

North America

- North-central US – Chicago, IL
- South-central US – San Antonio, TX
- West US – California
- East US – Virginia

Asia

- East Asia – Hong Kong, China
- South East Asia – Singapore

Europe

- West Europe – Amsterdam, Netherlands
- North Europe – Dublin, Ireland

8.2.2 Oracle public cloud

Oracle cloud aims to provide software to clients using the SaaS principle. Services are rented to clients on a subscription or pay-as-you-go basis.

8.2.3 Amazon Web Services

Amazon Web Services (AWS) is a collection of remote computing services (also called web services) that, together, make up a cloud computing platform, offered over the Internet by Amazon.com. The most central and well-known of these services are Amazon Elastic Compute Cloud (EC2) and Amazon simple storage (S3) [15].

8.2.4 Rackspace Cloud

The Rackspace Cloud is a set of cloud-related products and services that is billed on a utility computing basis, or pay as you go method. It includes web application hosting, cloud storage, Virtual Private Server Cloud Load Balancers, Cloud Databases, Cloud Backup and Cloud Monitoring. It was one of the first commercial cloud computing services [16].

8.2.5 Google Cloud Connect

Google Cloud Connect is a free cloud computing plug-in for Windows Microsoft Office 2003, 2007 and 2010 and can automatically store and synchronise any Microsoft Word document, PowerPoint presentation, or Excel spreadsheet to Google Docs in Google Docs or Microsoft Office formats. The Google Doc copy is automatically updated each time the Microsoft Office document is saved. Microsoft Office documents can be edited offline and synchronised later when online. Google Cloud Sync maintains previous Microsoft Office document versions and allows multiple users to collaborate, working on the same document at the same time [6].

8.2.6 Why MS Azure?

When developing an application with the intent of hosting it in cloud, one is given a variety of choices on the cloud platform. The following is a list of reasons that encouraged the deployment of the cost-effective school management system in MS Azure.

1. MS Azure provides a 90-day trial period that allows users to deploy and test the programs. In this way, a user can gauge the platform as well as the way in which the hosted services work. If fully satisfied, the user can later upgrade to the subscription package and utilise its full functionality.

2. Azure is based on Windows; thus, one can write applications in the same programming languages one uses for Windows apps: Visual Basic, C++, C#, etc. One can also use familiar tools such as Visual Studio, along with ASP.NET and other Windows technologies. This makes it easy for organisations to find developers who already have the skills to create applications for the Azure platform. Also, because the Azure environment is much like the standard Windows environment, it is easier to create a cloud version of an existing Windows application [22].

3. Applications running on Azure run in virtual machines, with each instance of the application running in its own VM on the 64-bit Windows Server 2008 operating system. The hypervisor on which they run is designed specifically for the cloud. There is no need to supply one’s own VMs

or deal with managing and maintaining the OS, because applications are developed using web role instances or worker role instances, which run in their own VMs [22].

4. Microsoft provides the Windows Azure Software Development Kit (SDK), which includes a version of the Azure environment that can run on one's own computer. It is called the Windows Azure Development Fabric and includes the Azure agent and storage. One can work locally when developing and debugging an application and then move it to the cloud [22].

5. Using Azure, one can easily create applications that run reliably and scale from 10 to 10 thousand or even 10 million users, without any additional coding. Azure Storage provides scalable, secure, performance-efficient storage services in the cloud [22].

6. Taking advantage of resources in the cloud allows one to decrease your costs for building and expanding your on-premises resources. One can also reduce the cost of IT administration because the hardware is being taken care of, off-premises [22].

7. SQL Azure provides organisations with all the benefits of an enterprise-class data centre without the hassles and cost of maintaining such an entity. One gets high availability and reliability with redundant copies of one's data and automatic failover, without concern about backing up data [22].

8. With Azure, one can develop hybrid applications that allow your on-premises applications to use cloud services, such as the cloud database and storage services. Communications services work between on-premises applications and the cloud, as well as mobile devices [22].

9. Knowing that security is one of the biggest concerns for companies considering a move to the cloud, Microsoft designed Azure with security in mind. The .NET Access Control Service provides a way to integrate identities, and Security Assertion Markup Language (SAML) tokens are used by applications to determine whether a user is allowed access. Microsoft has designed its compliance framework to meet regulatory requirements [22].

10. Windows Azure can benefit hosting providers, ISVs, systems integrators, and custom software developers. Hosting providers can expand their services to areas where they do not have existing infrastructure, and add new services without more infrastructure investment. ISVs can use Azure to create, deploy and manage Web apps and SaaS without large capital expenditures, and they can scale those applications more quickly and cost effectively [22].

8.3 Deploying/migrating a project onto MS Azure

Migrating is the process of deploying an existing ASP.NET project onto the MS Azure cloud. Before deploying onto MS Azure, the website has to be converted into a web application, after which a Windows Azure subscription has to be created, and the servers which are closest to the developer have to be identified and selected. The supporting database has to be migrated onto SQL Azure first.

8.3.1 SQL database migration

The SQL database is migrated with the help of SQL Azure Migration Wizard. A software that can be downloaded from the web free of cost. A pre-requisite of using the Migration Wizard is that an exception for port 1433 in Windows firewall should be created. This allows connection to a remote database from the desktop. SQL Azure Migration Wizard can be executed from the folder it is saved in.

A step-by-step migration procedure is explained as follows:

- Select local SQL server name and select database.
- Select database, select the stored procedures and tables associated with the database, and click next. This will result in the generation of script for the database.
- The next step is connecting the database with the SQL Azure. This requires information such as server name, and username and password which were generated when the subscription to be inputted was created. Then, click connect.
- Afterwards the setup target server connection window is opened. Here we create a database onto which our database will be loaded. After creating the database, we have to execute the script created against the newly created database. After this, Migration Wizard can be closed and the Azure portal is refreshed.
- Select the newly created database in the portal and click manage.
- This opens a login window. Once login has been done, the tables and stored procedures that were created can be viewed.
- The newly created database contains no data. The next step is to migrate the data.

- Using the SQL management studio, the local database that contains all the data is selected by right-clicking on the database, selecting task and selecting export data.
- The export data wizard helps to migrate all the data onto SQL Azure.

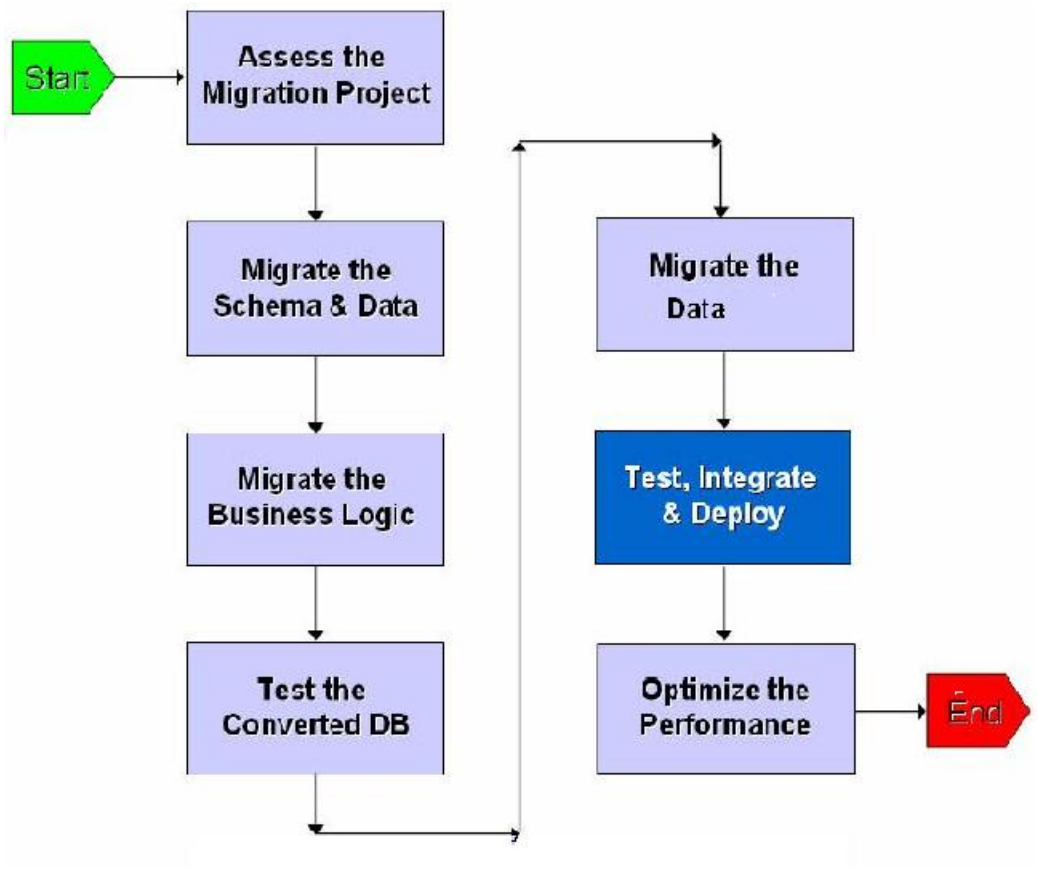


Figure 55 Migrating a SQL database [20]

Figure 55 is a flow chart showing the migration of a SQL database.

8.3.2 Connecting existing application onto SQL Azure

- Select the SQL Data Source Wizard in the web form.
- Select new connection, and enter the fully qualified dns name given in the MS Azure portal.

- Enter the username and password, select the created Azure database and click database.
- Copy the connection string and paste it in the connection class.

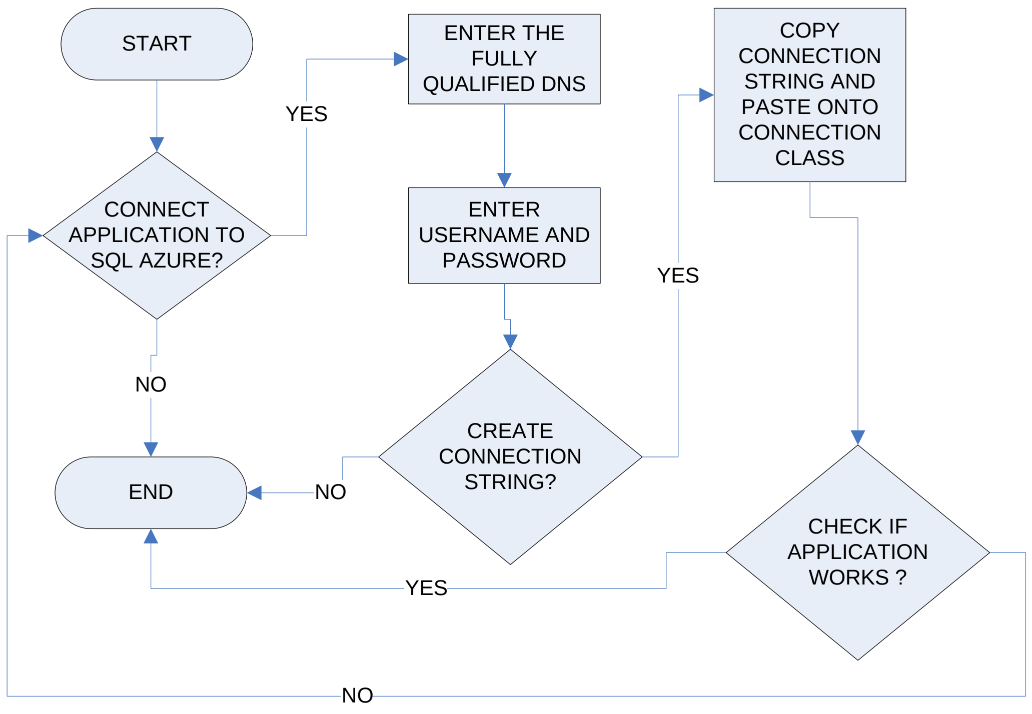

START
CONNECT APPLICATION TO SQL AZURE?
YES
ENTER THE FULLY QUALIFIED DNS
ENTER USERNAME AND PASSWORD
CREATE CONNECTION STRING?
YES
COPY CONNECTION STRING AND PASTE ONTO CONNECTION CLASS
CHECK IF APPLICATION WORKS ?
NO
NO
END
YES
NO

Figure 56 Connecting an existing application with SQL Azure

Figure 56 depicts a flow chart representation of connecting an existing application onto SQL azure.

8.3.3 Publishing a website in MS Azure

- Create a new cloud project.
- Right click roles under the cloud project created and add existing web role. Web role is the web application we have created.
- Right click the existing web role and select properties. From the Azure portal, select the primary connection string and input it under settings.

- Now information regarding the Azure account will have to be inputted, namely account key and account name.
- Next, right click the cloud project and select publish.
- A new certificate will have to be created and the path to that certificate will have to be copied and uploaded in MS Azure.
- Back in the publishing wizard, the subscription id will have to be entered, which helps to authenticate the website onto which the application will be published.
- Click ok. This will authenticate all the information and populates data on staging and deployment.
- The website is now published onto Azure.

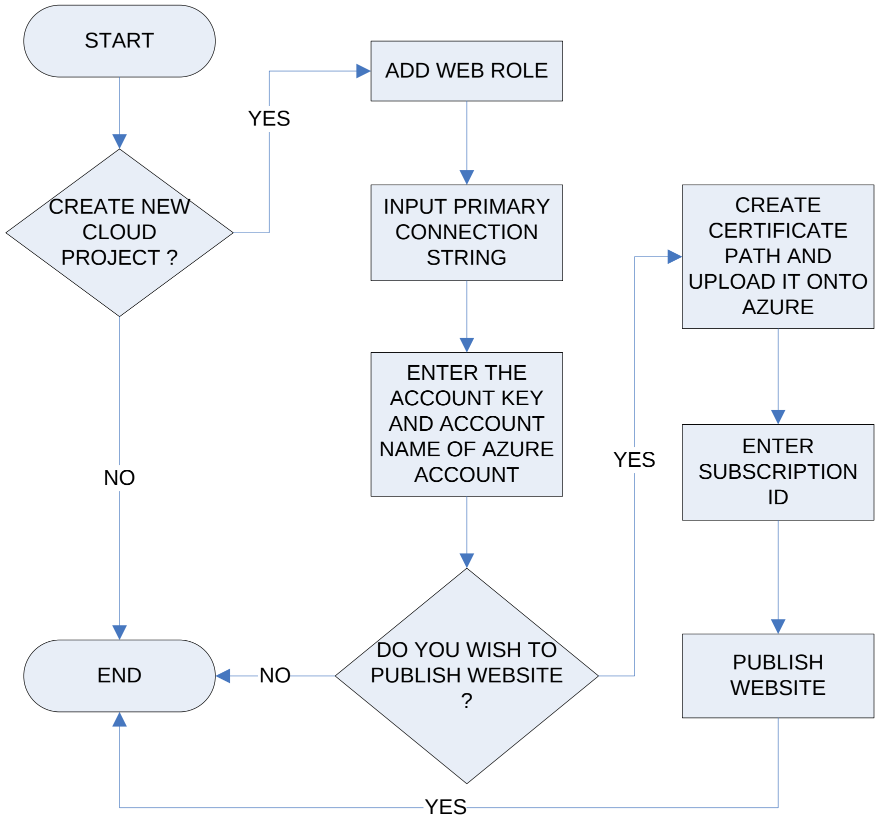

START
CREATE NEW CLOUD PROJECT ?
YES
ADD WEB ROLE
INPUT PRIMARY CONNECTION STRING
ENTER THE ACCOUNT KEY AND ACCOUNT NAME OF AZURE ACCOUNT
DO YOU WISH TO PUBLISH WEBSITE ?
YES
CREATE CERTIFICATE PATH AND UPLOAD IT ONTO AZURE
ENTER SUBSCRIPTION ID
PUBLISH WEBSITE
NO
NO
END
YES

Figure 57 publishing a website in MS-Azure

8.4 Summary

In this study, MS Azure was selected to deploy the system being developed, because it allows migration of existing applications and databases onto it. Being an ASP application, which requires frequent debugging and improvements in its features, it was essential that hosting be possible only after thorough validation. MS Azure also allows a free trial of 90 days of hosting of the application, which allows users to observe, use and suggest changes to the system before official release of the system.

Chapter 9: Demonstration and Evaluation

9.1 System Evaluation and Testing

Software testing is an investigation conducted to provide the stake holder with information about the quality of the product or service developed [17]. Testing also aids in measuring not only the behaviour of a system when different numbers of people access the software, but also how the system reacts when different people use the system simultaneously. An integral part of software testing is studying how a system reacts to inputs from different users as well as how a system reacts when different numbers of users access the system simultaneously.

Different types of testing are available for different requirements. Common testing methods include performance testing, usability testing, security testing, and user acceptance testing. These methods will be discussed below.

9.2 Performance testing

Software performance testing is used to measure how well the software performs and how resistant it is to failure. Software performance testing by itself can be divided into subsets, namely load tests, stress tests, spike tests, component tests and capacity tests [17]. The key test that has been carried out in this project is load testing, which will be explained in more detail.

9.2.1 Load testing

A load test determines the way in which an application behaves under different numbers of users, users that access a particular system at a given time, or the number of transactions that are performed at a given time. Load testing gives the response time that is required for critical transaction. Load testing overall helps gauge the efficiency of a system during intervals.

9.2.2 Pre-requisite for performance testing

The pre-requisite for performance testing is an error-free system that conforms to requirements.

9.2.3 Test conditions

Performance testing, in an ideal scenario, would be carried out in an environment resembling its targeted environment. This is, however, not fully possible in practice. The reason is that the workloads of production systems have a random nature and, while the test workloads do their best to mimic what may happen in the production environment, it is impossible to exactly replicate this workload variability, except in the simplest system [17].

9.2.4 Timing

Performance testing of software begins from the very inception of the software. The later a performance defect is detected, the higher the cost of remediation. It is always crucial for a performance test team to be involved as early in the process as possible [17].

9.3 Security testing

Software threats increase every day; hence, the software that is developed should safeguard the integrity of the computers it runs on as well as the data it holds [18]. Security testing is carried out to ensure that neither the software nor the data it holds is compromised under attack.

9.4 Usability testing

In usability testing, a system is evaluated by people who have no prior experience working on the system. Usability testing is conducted to get an idea of how novice users feel about the software, how user friendly they find it, and how well it serves its intended purpose. This is in contrast with usability inspection methods where experts use different methods to evaluate a user interface without involving users.

9.5 User Acceptance Testing (UAT)

Testing performed by the customer is known as User Acceptance Testing (UAT). User acceptance helps determine the success of developed software.

9.6 Test results

The cost-effective school management system for disadvantaged schools in the Free State was subjected to usability testing by a group of first-time users with the aim to measure how they felt about the system. About 20 users participated in this survey. The average of their findings is illustrated in figure 53 below. These users are part of the targeted community which the software hopes to serve in future.

A questionnaire was designed for the user to complete after testing each module (see appendix 3). The following main modules were tested:

- School admin
- Teacher
- Principal

- Student

The users were asked to rate the efficiency of the developed system on a scale of 1-5. The following bar charts were calculated and represent the data that were analysed.

9.6.1 Usability testing of school admin module

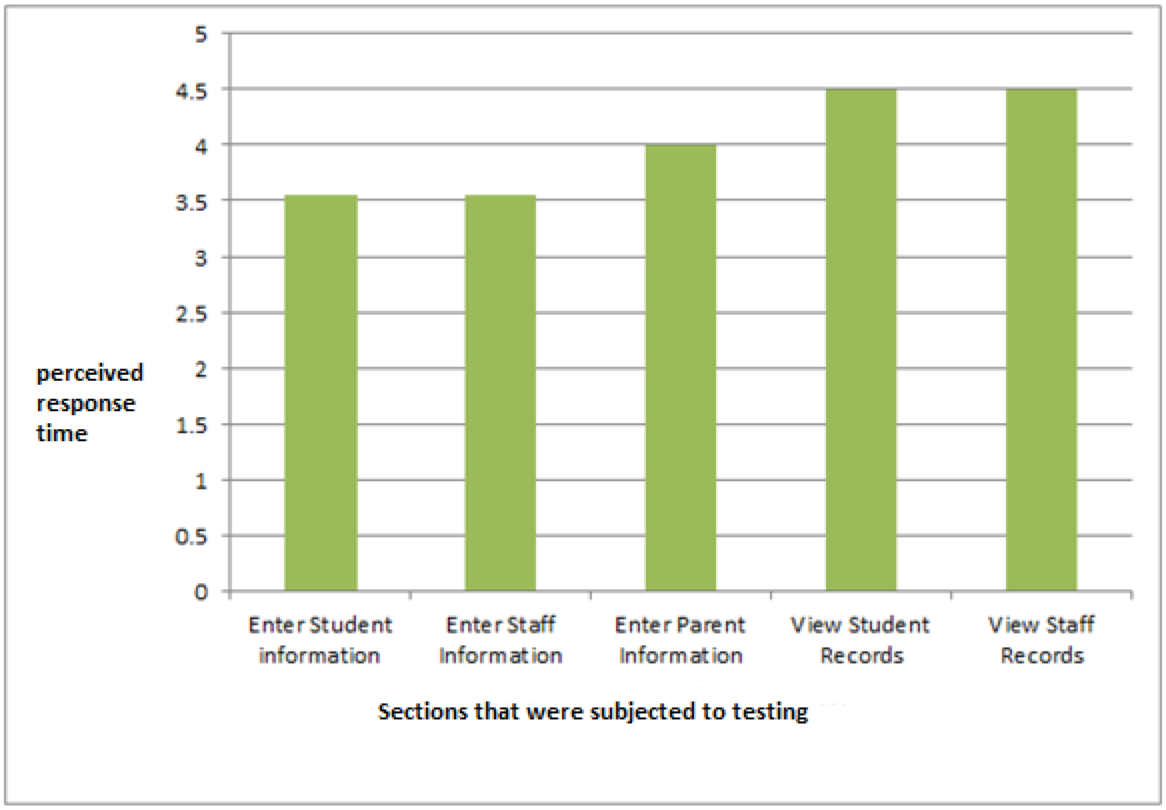


Figure 58 Usability testing of school admin module

The x-axis contains the sections that were tested in the school admin module, and the y-axis contains the users' rating of perceived response time. Each module's perceived response times is measured on a scale of 1-5, 1 being extremely slow and 5 being exceptionally fast. It is clear that most user's agree that, during peak load, the system does have a reasonable speed in recording information while, at the same time, it demonstrates above an average speed in retrieving data from the system.

9.6.2 Usability testing of teacher module

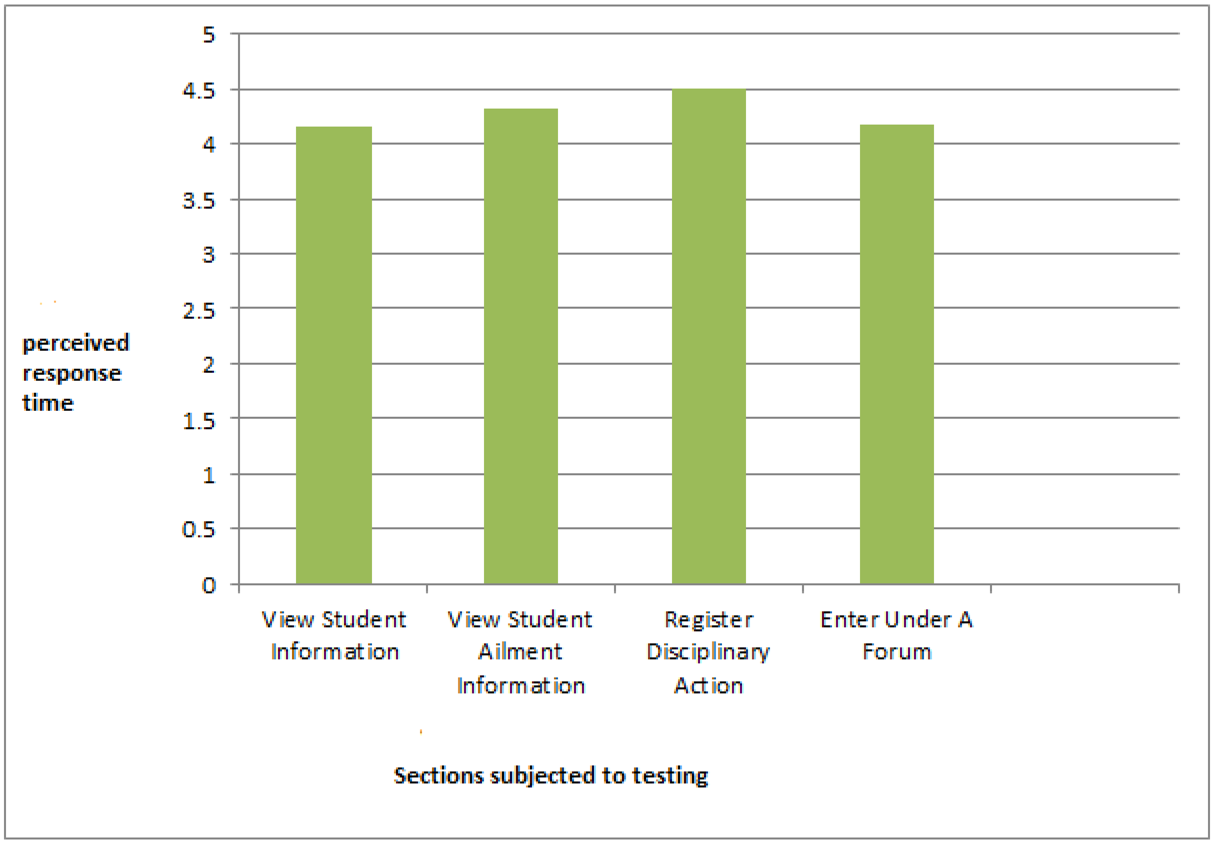


Figure 59 Usability testing of the teacher module

The x-axis contains the sections that were tested in the teacher module, and the y-axis contains the users' ratings of perceived response times for each section on a scale of 1-5, 1 being extremely slow, and 5 being exceptionally fast. During the testing of the teacher module, it was reported that the viewing of information, and writing in the forum displayed above average speeds, whereas registering disciplinary action demonstrated exceptionally fast speed.

9.6.3 Usability testing of principal module

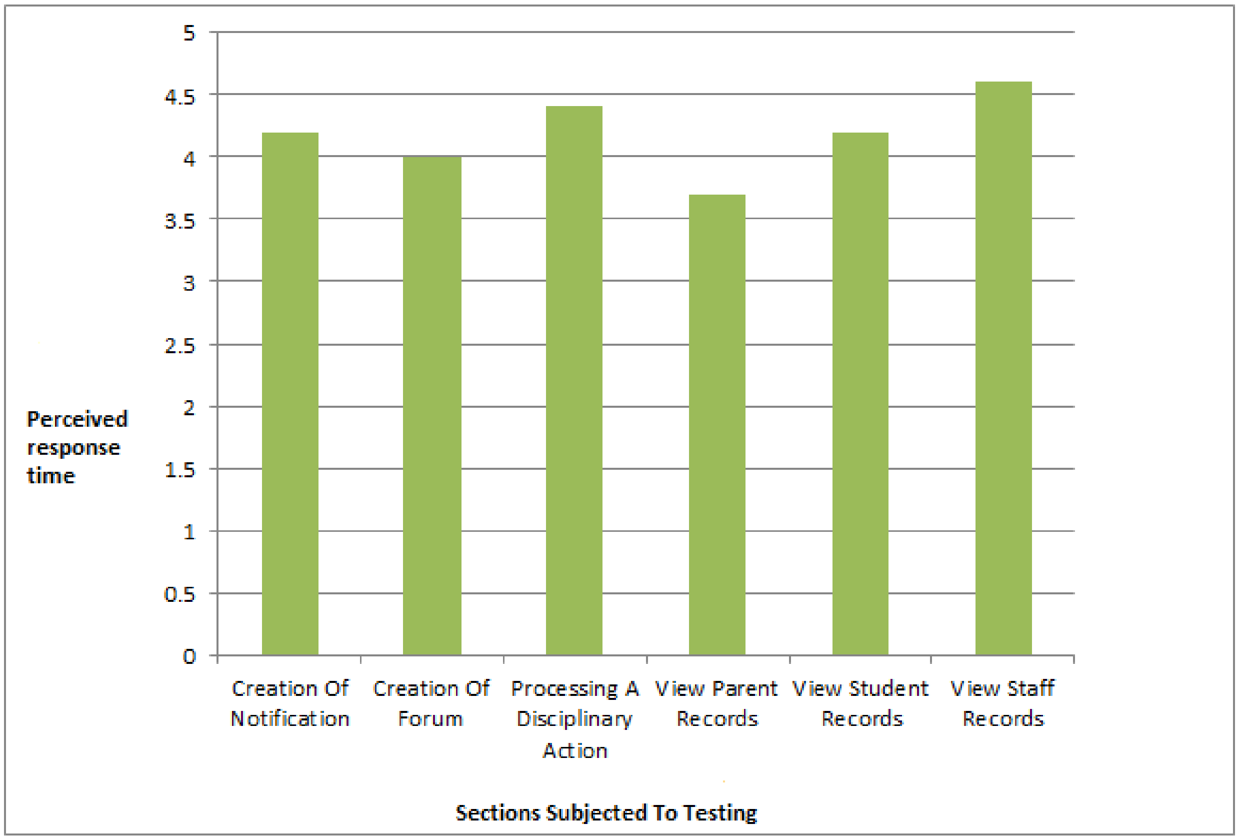


Figure 60 Usability testing of the principal module

The x-axis contains the sections that were tested in the principal module, and the y-axis contains the users' rating of perceived response time for each section on a scale of 1-5, 1 being extremely slow, and 5 being exceptionally fast. The time it takes in this module to enter data is longer than average, whereas the time it takes to view saved data is exceptionally fast.

9.6.4 Usability testing of student module

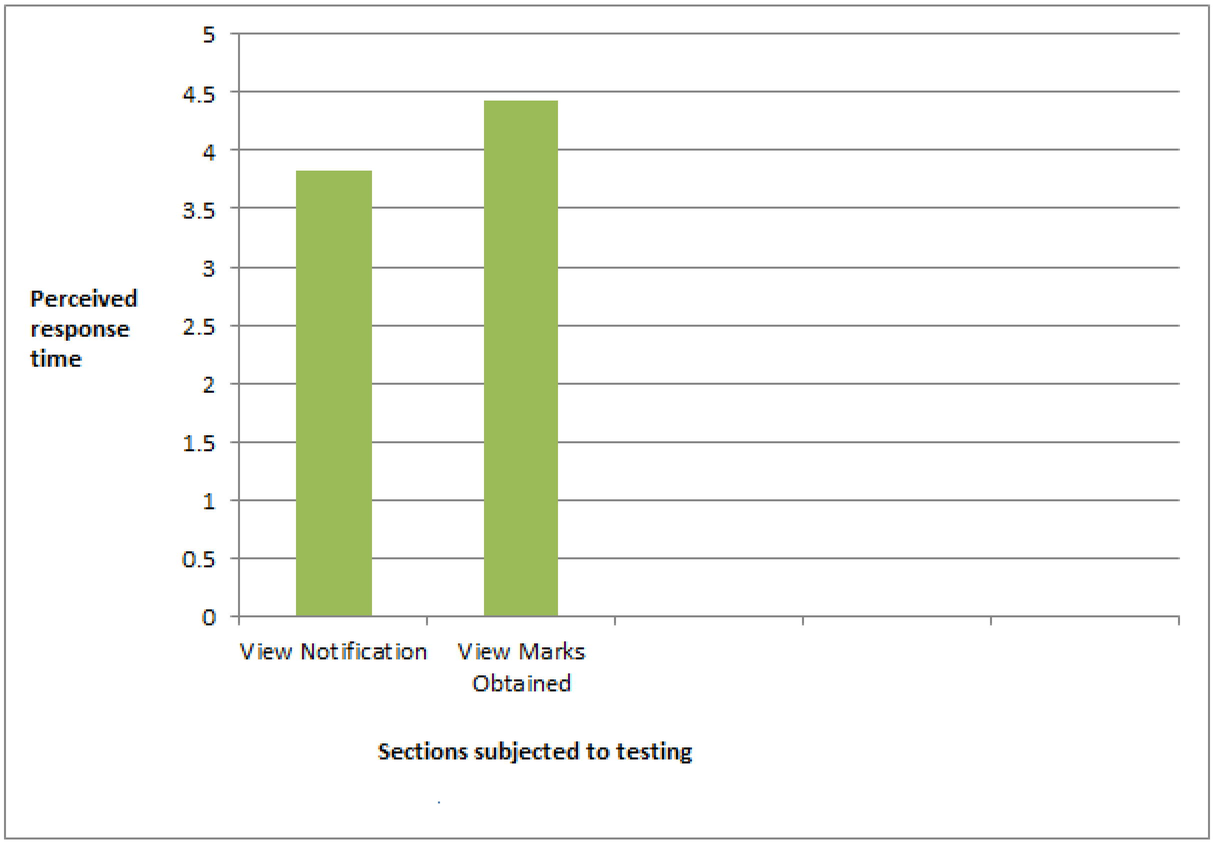


Figure 61 Usability testing of the student module

The x-axis contains the sections that were tested in the student module, and the y-axis contains the users' rating of each section on the scale of 1-5, 1 being extremely slow, and 5 being exceptionally fast. The time it takes to view the notification was reported as average, whereas the time it takes to view the marks were recorded as longer than average.

9.6.5 Usability testing of site admin module

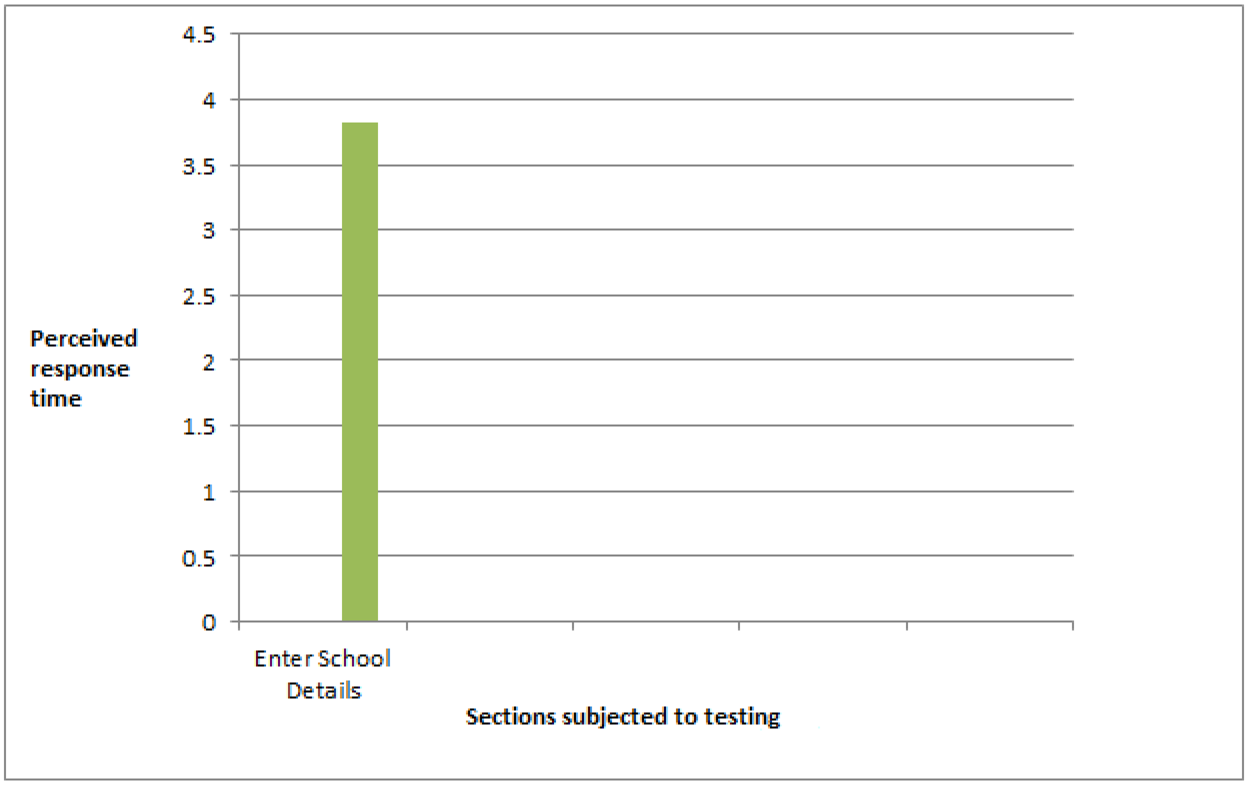


Figure 62 Usability testing of the site admin module

The x-axis contains the sections that were tested in the site admin module, and the y-axis contains the users' rating of the perceived response time. Each section is measured on a scale of 1-5, 1 being extremely slow, and 5 being exceptionally fast. Registering information about a school in order to become a member of the system displayed an average speed.

9.7 Results

This chapter presented results of a usability testing for the system developed in this study. The results suggest that users found the system usable, with perceived response times all falling within acceptable range. The only module that displayed an average response time during peak hours was the notification module.

Chapter 10: Conclusions and Recommendations

10.1 Conclusion

The system discussed in this thesis was developed with the sole objective of giving educators deeper insight into the students and their background. All the users who tested this software agreed that, through its usage, the goal of acquiring better knowledge of students and their background is possible as well as informing parents about their children's activities and progress. Through this software, the principal can judge the overall pulse of the school as well the difficulties faced by teachers.

A school, being a non-profit organisation, has limited funds. This is where SaaS fulfils a significant role in making software more affordable. SaaS enables the renting out of the software, unlike conventional software which needs to be bought and licenses renewed yearly. Furthermore, cloud deployment assists in reaching a wider audience at the same time due to multitenancy and facilitates upgrading and solving errors within.

Overall, the software succeeds in uniting different users in the school system while simultaneously maintaining affordability. The hope is expressed that better academic results can be produced in future through the usage of this system.

10.2 Future recommendations

It is recommended that a software metric system be designed to measure and calculate the cost of software usage. The forum module may also be expanded to include parents as well. In addition, the forum may be designed to unite different schools and their users so as to understand and solve problems faced by schools in different parts of the country.

Chapter 11: References

[1]SA-SAMS Available:
http://www.thutong.doe.gov.za/administration//Administration//GeneralInformation//SASAMS/tabid/3346/Default.aspx [Accessed January, 21. 2011]

[2]M. Rouse (December 2010) Available:
http://searchcloudcomputing.techtarget.com/definition/cloud-computing [Accessed May 5, 2012]

[3] K. Benedict. (2011, Oct.27). "Six Types of Cloud Computing" Available:
http://cloudcomputing.sys-con.com/node/2038121 [retreived April, 15. 2011]

[4] S. Sehlhorst. "The Economics of Software as a Service (SaaS) vs. Software as a Product" Available: http://www.pragmaticmarketing.com/publications/magazine/6/5/the-economics-of-software-as-a-service-saas-vs-software-as-a-product [Accessed May. 9, 2012].

[5] B.Violino (2009, May, 19.). "What Are the Hottest Areas for SaaS Apps?" Available http://www.ciozone.com/index.php/Management/What-Are-the-Hottest-Areas-for-SaaS-Appsu.html [Accessed May,15. 2012].

[6] F. Chong et al. (2006, Jun.). "Multi-Tenant Data Architecture, Microsoft Corporation [Online]." Available: http://msdn.microsoft.com/en-us/library/aa479086.aspx [Accessed May. 22, 2012].

[7] C. J. Kothari (2007, Sept. 25) “Meeting Security Requirements of Software as a Service (SaaS) Applications” Available http://www.ibm.com/developerworks/library/ar-saassec/ [Accessed August. 2, 2012].

[8] S. Raman. “Non-Functional Requirements for SaaS ~spark~ in the cloud available”, http://sultanate.blogspot.com/2010/09/non-functional-requirements-for-saas.html [Accessed August. 11, 2012]

[9] J. R Groff et al., “The Complete Reference SQL”. United States of America: Mcgraw Hill, 2010.

[10] “Free and Open Source Software (FOSS)” available http://gadgeteer.co.za/opensourcesoftware#Databasesen.html&docid=S_b9V9n2PG1SOM&imgurl=http://manu.chao.free.fr/freesofts4win/dia-screenshot1.png&w=730&h=724&ei=j1UFUMLfI4OIhQeot_XNBw&zoom=1&iact=hc&vpx=188&vpy=324&dur=125&hovh=224&hovw=225&tx=125&ty=125&sig=109504721973332583016&page=1&tbnh=163&tbnw=164&start=0&ndsp=18&ved=1t:429,r:6,s:0,i:89 [Accessed Nov, 25. 2012]

[11] B. Evjen et al., “Professional ASP.NET 3.5” In C# and VB. Indianapolis: Wiley publishing, Inc (2008).

[12] ASP 8211 “Its Advantages And Disadvantages” (2009, May. 27) Available: http://www.techyshell.com/internet/asp-its-advantages-and-disadvantages/ [Accessed July, 15. 2012].

[13] J. Crowley (2011, April. 24) available http://www.developerfusion.com/article/4278/using-adonet-with-sql-server/ [Accessed July. 15, 2012]

[14] R. Brunetti, “Windows Azure Step By Step”. California: O’Reily Media, Inc (2011).

[15] “What is Cloud Computing?” available http://aws.amazon.com/what-is-cloud computing/[Accessed August. 20, 2012]

[16] Wikipedia “Rackspace Cloud” available: http://en.wikipedia.org/wiki/Rackspace_Cloud [Accessed September, 21. 2012]

[17] J.D. Meier, et al., (2007 September) available: http://msdn.microsoft.com/en-us/library/bb924357.aspx [Accessed August. 20, 2012]

[18] T. Winograd, (2008, October.1). “Enhancing the development life cycle to produce secure software”. Available https://sw.thecsiac.com/techs/abstract/358844 [retrieved August. 31, 2012]

[19] B.Violino (2009,May.19) "Cost and Agility Driving Demand for SaaS Apps" available http://www.ciozone.com/index.php/Management/Cost-and-Agility-Driving-Demand-for-SaaS-Apps.html [retrieved September 1 2012]

[20] D. L .Fisherman "Database Migration: The Key to Unlocking your Business Assets" available
http://vyaskn.tripod.com/database_migration_white_paper.htm [Accessed September. 1, 2012]

[21]"CRM Software-as-a-Service (SaaS) Reality" available http://www.online-crm.com/saas_advantages_disadvantages.htm [Accessed October. 1, 2012]

[22] D. Shinder (2010, January 6.) "10 reasons to use Azure for your cloud apps available" http://www.techrepublic.com/blog/10things/10-reasons-to-use-azure-for-your-cloud-apps/1282 [Accessed September. 9, 2012]

[23] Timothy Grance, Peter Mell, The NIST definition of cloud computing. Special Publication 800-145, September 2011.

[24]"Object Explorer" available http://msdn.microsoft.com/en-us/library/ms173849.aspx [retreived October 10 2012]

[25] A. R. Hevner, et.al," Design science in information systems research" MIS Quartely, vol 28, N.o. 1, pp75-105, March 2008.

[26] K. Peffers, et.al, “A design science research methodology in information systems research” Journal of management information systems, Vol 24, N.o. 3, pp 45-74, winter 2007-8.

[27] M. Amburst, et.al, “Above the clouds a berkely view of cloud computing” Communications of the ACM, volume 52, n.o. 4, pp 50-58, April 2010

[28] F.Chong , G. Carraro. (April. 2006). “Architecture Strategies for Catching the Long Tail.” Available: http://msdn.microsoft.com/enus/library/aa479069.aspx#docume_topic4 [Accessed july. 29, 2012].

[29] F. Hoch, et.al. (2004 July) “Software as a service: Changing the paradigm in the software industry, SIIA and Tripletree Industry analysis series.” Available: www.siia.net [Accessed on Oct. 10. 2012].

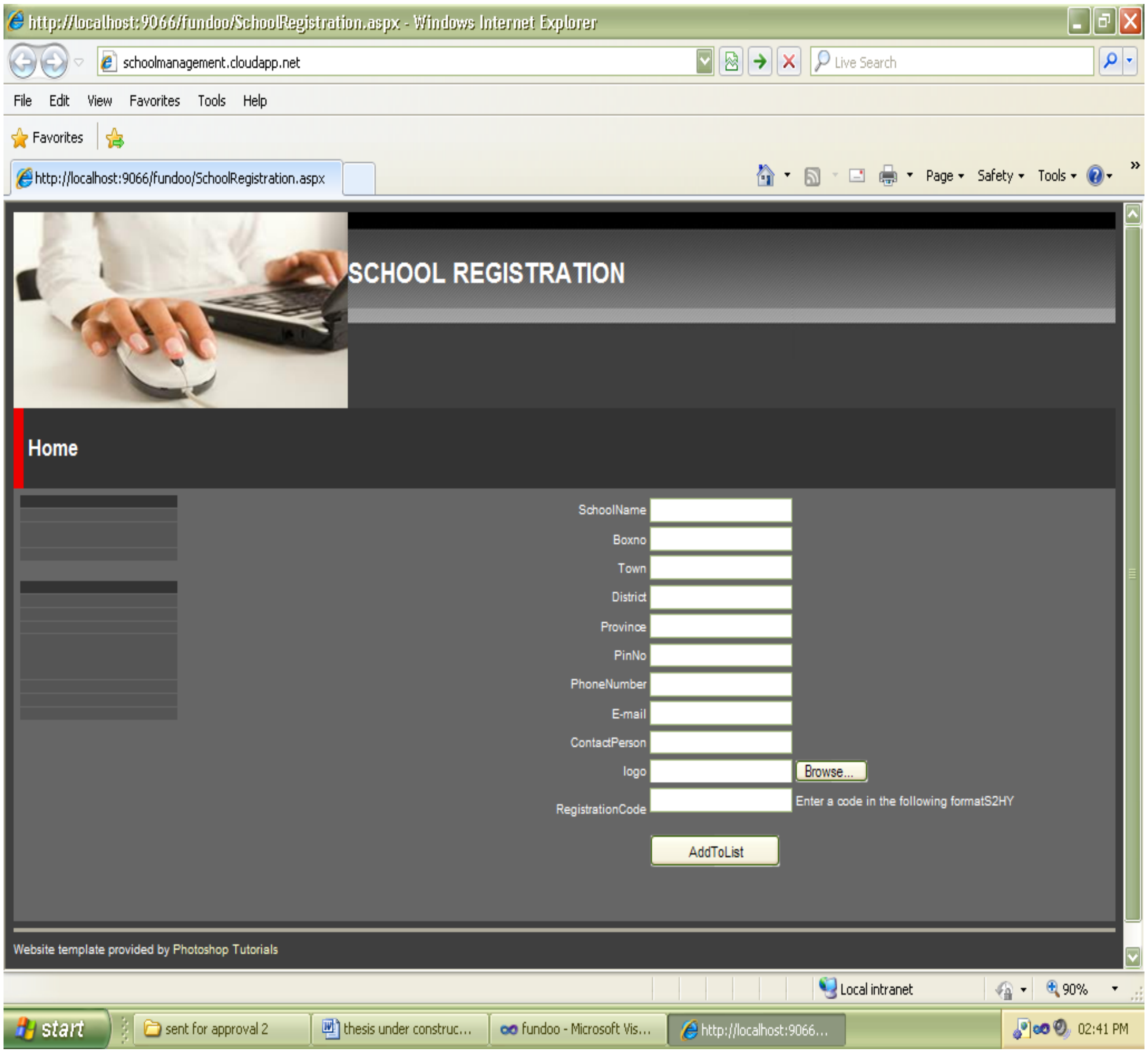
[30] J. Sysmans et.al “Software as a service: a comprehensive look at the cost of ownership of software applications” Available: www.siia.net [Accessed on Oct. 10. 2013].

[31] A. M. Basal, A.L Steenkamp (2010) “A Saas-Based Approach in an E-Learning System” Available at http://ijism.ricest.ac.ir/ojs/index.php/ijism/article/view/134 [Accessed on Oct 15 2013]

[32] N. Sakamoto “Construction of SaaS based E-learning system in Japan” Fujitsu Science and technological journal vol 45, No 3, pp.290-298, July 2009.

APPENDIX 1- SCREEN SHOTS

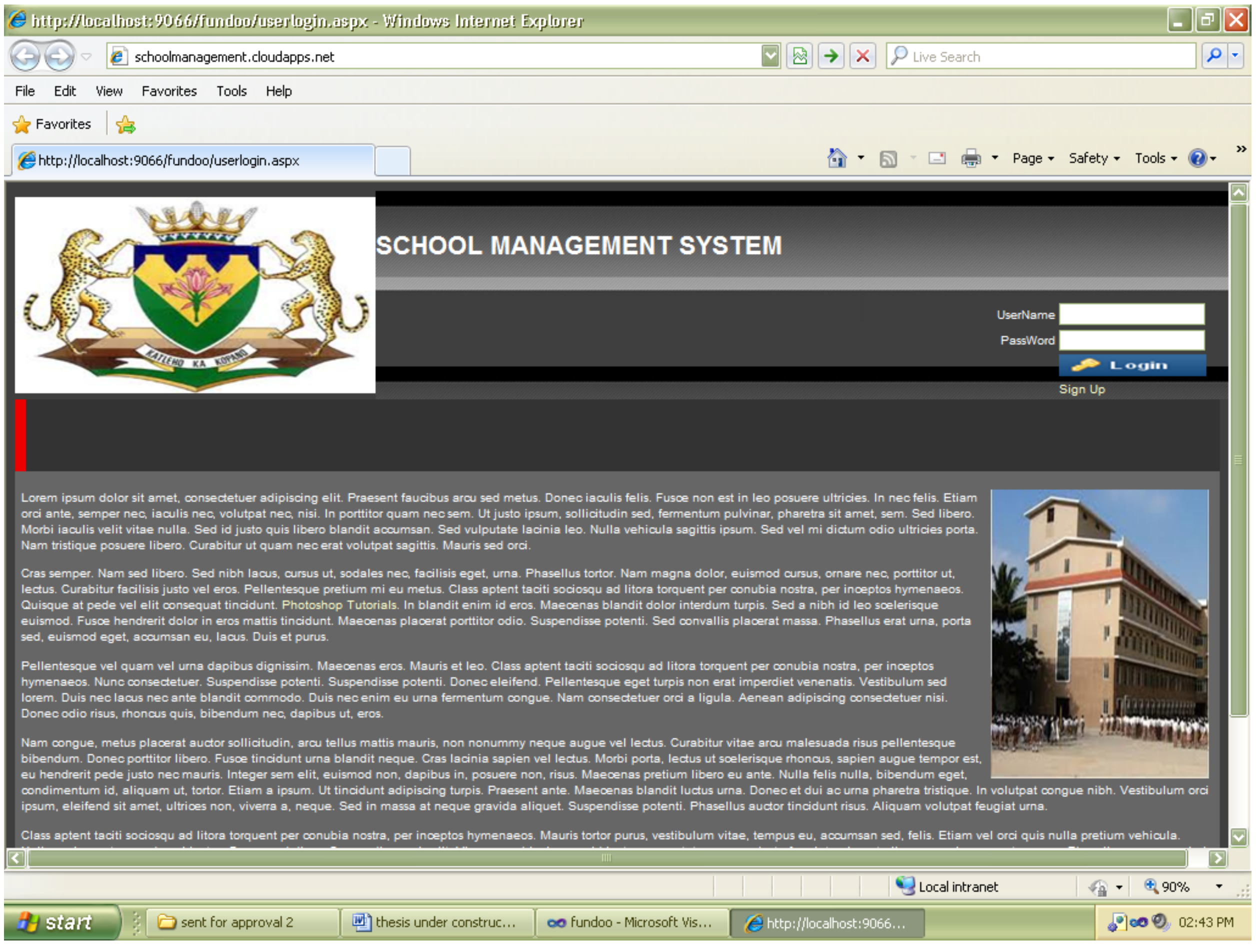
School registration


http://localhost:9066/fundoo/SchoolRegistration.aspx - Windows Internet Explorer
schoolmanagement.cloudapp.net
File Edit View Favorites Tools Help
Favorites
Page Safety Tools
SCHOOL REGISTRATION
Home
Browse...
AddToList
Website template provided by Photoshop Tutorials
Local intranet
90%
start
sent for approval 2
thesis under construc...
fundoo - Microsoft Vis...
http://localhost:9066...
02:41 PM

SchoolName	
Boxno	
Town	
District	
Province	
PinNo	
PhoneNumber	
E-mail	
ContactPerson	
logo	
RegistrationCode	

Enter a code in the following formatS2HY

Login form



http://localhost:9066/fundoo/userlogin.aspx - Windows Internet Explorer
schoolmanagement.cloudapps.net
Live Search
File Edit View Favorites Tools Help
Favorites
http://localhost:9066/fundoo/userlogin.aspx
Page Safety Tools
SCHOOL MANAGEMENT SYSTEM
UserName
PassWord
Login
Sign Up
Local intranet
90%
start
sent for approval 2
thesis under construc...
fundoo - Microsoft Vis...
http://localhost:9066...
02:43 PM

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Praesent faucibus arcu sed metus. Donec iaculis felis. Fusce non est in leo posuere ultricies. In nec felis. Etiam orci ante, semper nec, iaculis nec, volutpat nec, nisi. In porttitor quam nec sem. Ut justo ipsum, sollicitudin sed, fermentum pulvinar, pharetra sit amet, sem. Sed libero. Morbi iaculis velit vitae nulla. Sed id justo quis libero blandit accumsan. Sed vulputate lacinia leo. Nulla vehicula sagittis ipsum. Sed vel mi dictum odio ultricies porta. Nam tristique posuere libero. Curabitur ut quam nec erat volutpat sagittis. Mauris sed orci.

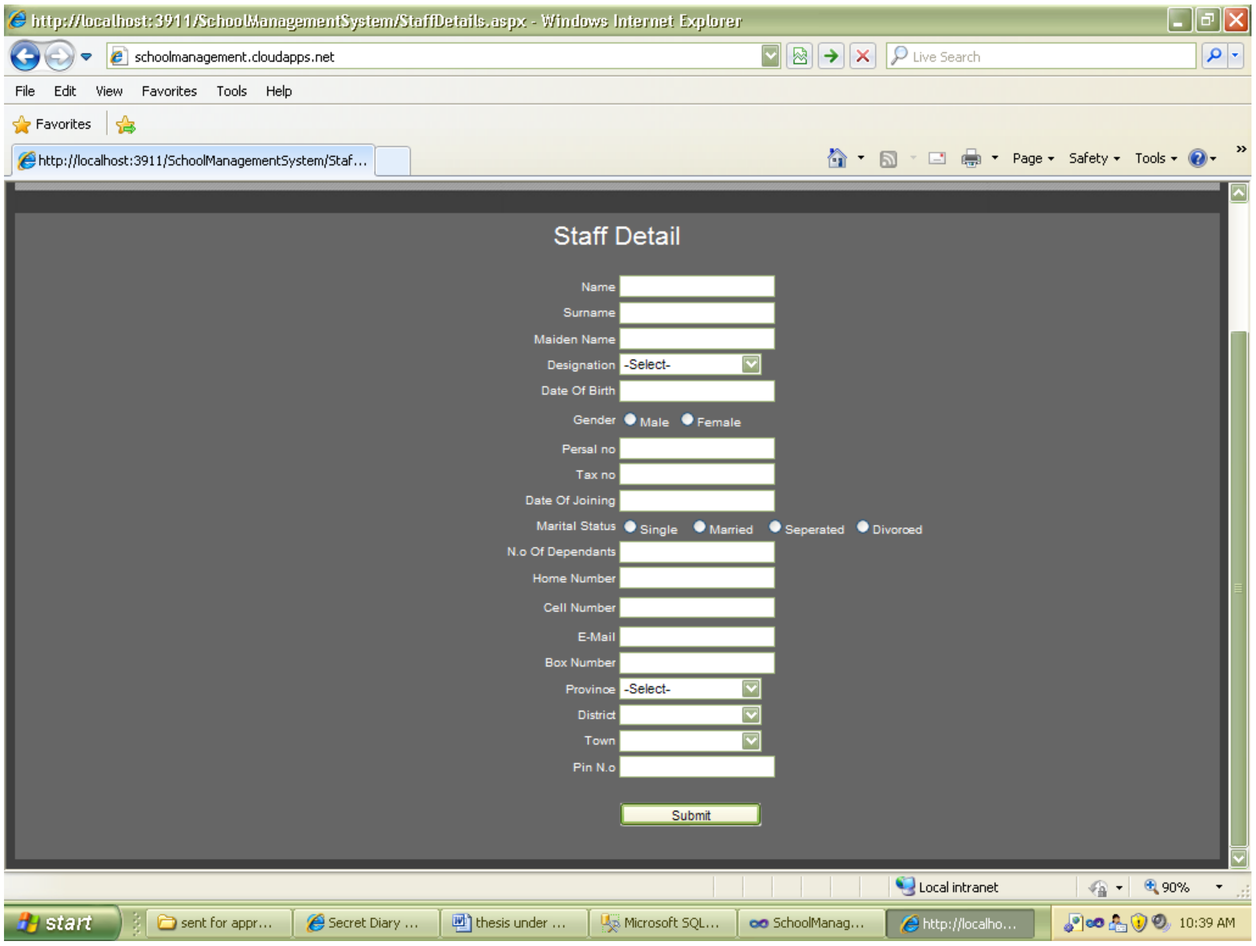
Cras semper. Nam sed libero. Sed nibh lacus, cursus ut, sodales nec, facilisis eget, urna. Phasellus tortor. Nam magna dolor, euismod cursus, ornare nec, porttitor ut, lectus. Curabitur facilisis justo vel eros. Pellentesque pretium mi eu metus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Quisque at pede vel elit consequat tincidunt. Photoshop Tutorials. In blandit enim id eros. Maecenas blandit dolor interdum turpis. Sed a nibh id leo scelerisque euismod. Fusce hendrerit dolor in eros mattis tincidunt. Maecenas placerat porttitor odio. Suspendisse potenti. Sed convallis placerat massa. Phasellus erat urna, porta sed, euismod eget, accumsan eu, lacus. Duis et purus.

Pellentesque vel quam vel urna dapibus dignissim. Maecenas eros. Mauris et leo. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Nunc consectetuer. Suspendisse potenti. Suspendisse potenti. Donec eleifend. Pellentesque eget turpis non erat imperdiet venenatis. Vestibulum sed lorem. Duis nec lacus nec ante blandit commodo. Duis nec enim eu urna fermentum congue. Nam consectetuer orci a ligula. Aenean adipiscing consectetuer nisi. Donec odio risus, rhoncus quis, bibendum nec, dapibus ut, eros.

Nam congue, metus placerat auctor sollicitudin, arcu tellus mattis mauris, non nonummy neque augue vel lectus. Curabitur vitae arcu malesuada risus pellentesque bibendum. Donec porttitor libero. Fusce tincidunt urna blandit neque. Cras lacinia sapien vel lectus. Morbi porta, lectus ut scelerisque rhoncus, sapien augue tempor est, eu hendrerit pede justo nec mauris. Integer sem elit, euismod non, dapibus in, posuere non, risus. Maecenas pretium libero eu ante. Nulla felis nulla, bibendum eget, condimentum id, aliquam ut, tortor. Etiam a ipsum. Ut tincidunt adipiscing turpis. Praesent ante. Maecenas blandit luctus urna. Donec et dui ac urna pharetra tristique. In volutpat congue nibh. Vestibulum orci ipsum, eleifend sit amet, ultrices non, viverra a, neque. Sed in massa at neque gravida aliquet. Suspendisse potenti. Phasellus auctor tincidunt risus. Aliquam volutpat feugiat urna.

Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Mauris tortor purus, vestibulum vitae, tempus eu, accumsan sed, felis. Etiam vel orci quis nulla pretium vehicula.

Entering a teacher into the database

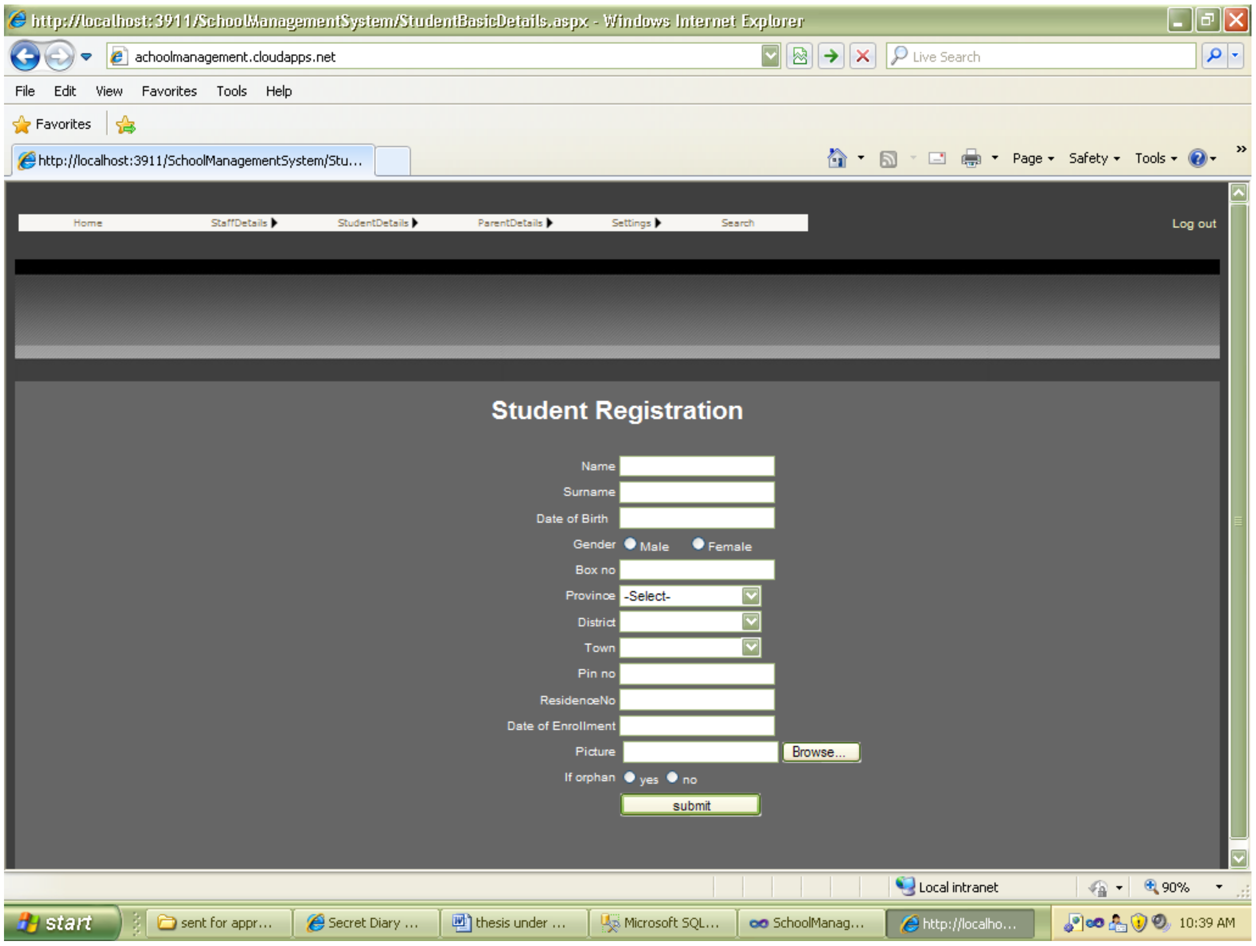

http://localhost:3911/SchoolManagementSystem/StaffDetails.aspx - Windows Internet Explorer
schoolmanagement.cloudapps.net
Live Search
File Edit View Favorites Tools Help
Favorites
http://localhost:3911/SchoolManagementSystem/Staf...
Page Safety Tools
Local intranet
90%
start
sent for appr...
Secret Diary ...
thesis under ...
Microsoft SQL...
SchoolManag...
http://localho...
10:39 AM

Staff Detail

Name	
Surname	
Maiden Name	
Designation	-Select-
Date Of Birth	
Gender	Male Female
Persal no	
Tax no	
Date Of Joining	
Marital Status	Single Married Seperated Divorced
N.o Of Dependants	
Home Number	
Cell Number	
E-Mail	
Box Number	
Province	-Select-
District	
Town	
Pin N.o	

Submit

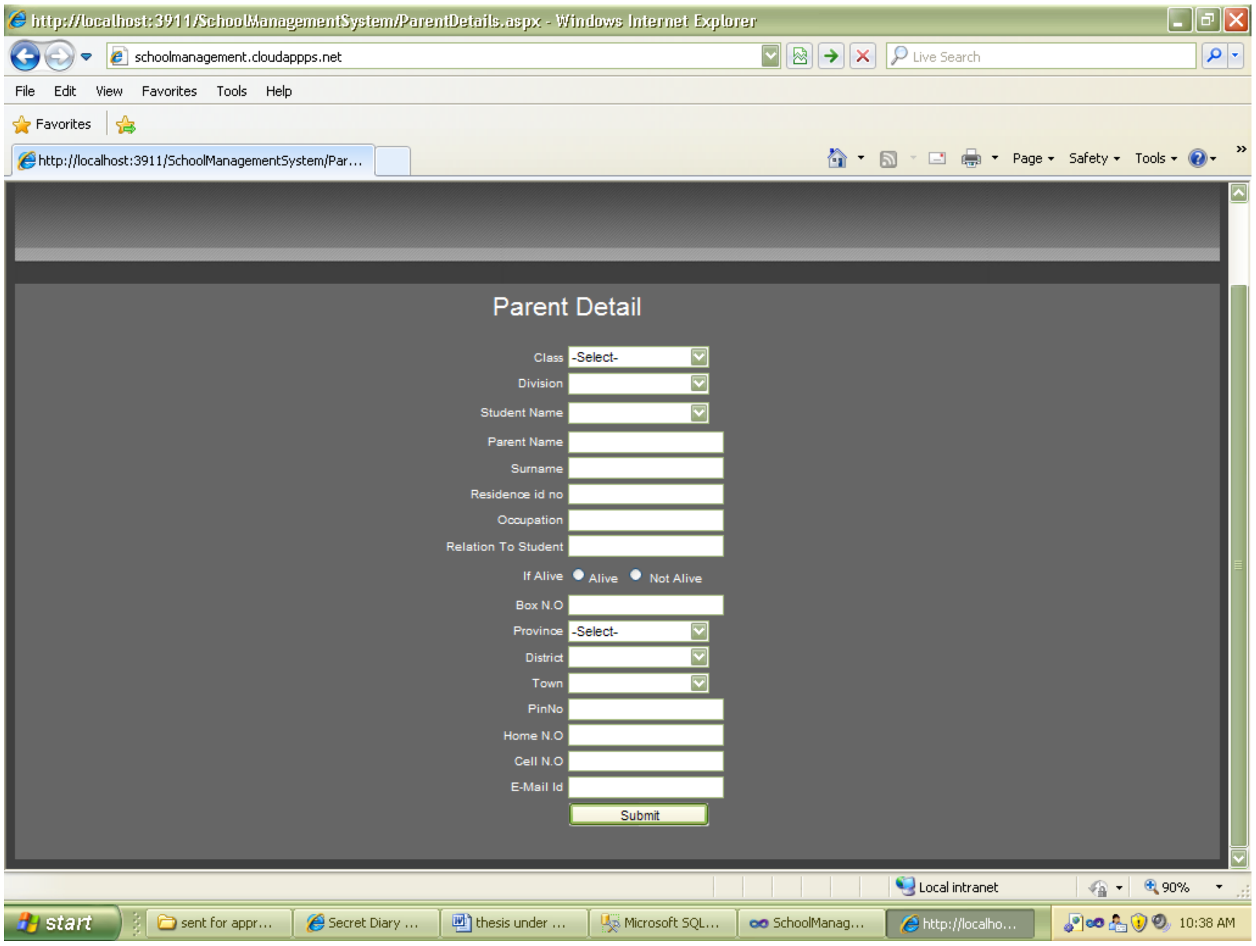
Entering a student into the database


http://localhost:3911/SchoolManagementSystem/StudentBasicDetails.aspx - Windows Internet Explorer
achoolmanagement.cloudapps.net
Home
StaffDetails
StudentDetails
ParentDetails
Settings
Search
Log out

Student Registration

Name	
Surname	
Date of Birth	
Gender	Male Female
Box no	
Province	-Select-
District	
Town	
Pin no	
ResidenceNo	
Date of Enrollment	
Picture	Browse...
If orphan	yes no
	submit

Entering a parent into the database

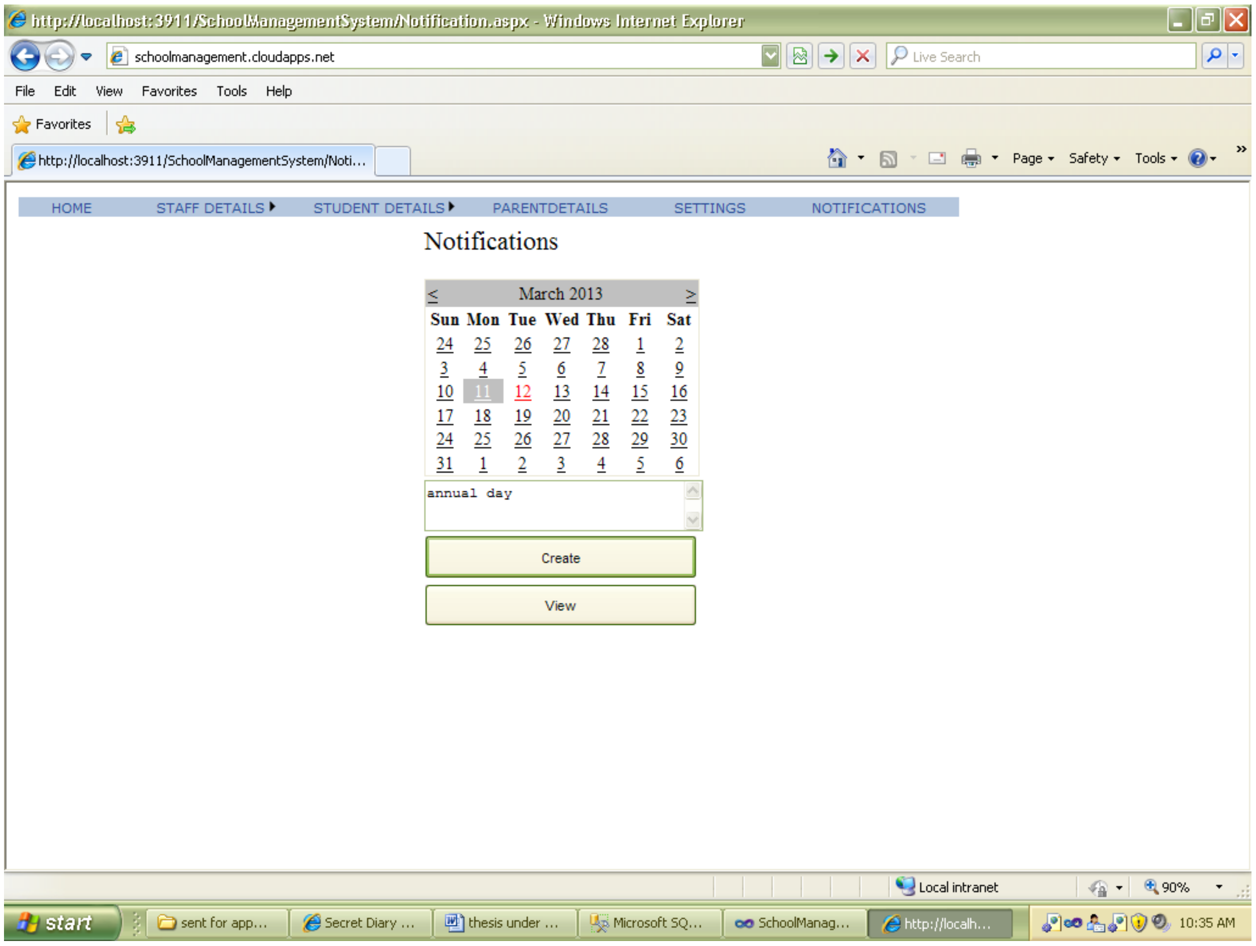

http://localhost:3911/SchoolManagementSystem/ParentDetails.aspx - Windows Internet Explorer

schoolmanagement.cloudappps.net

Parent Detail

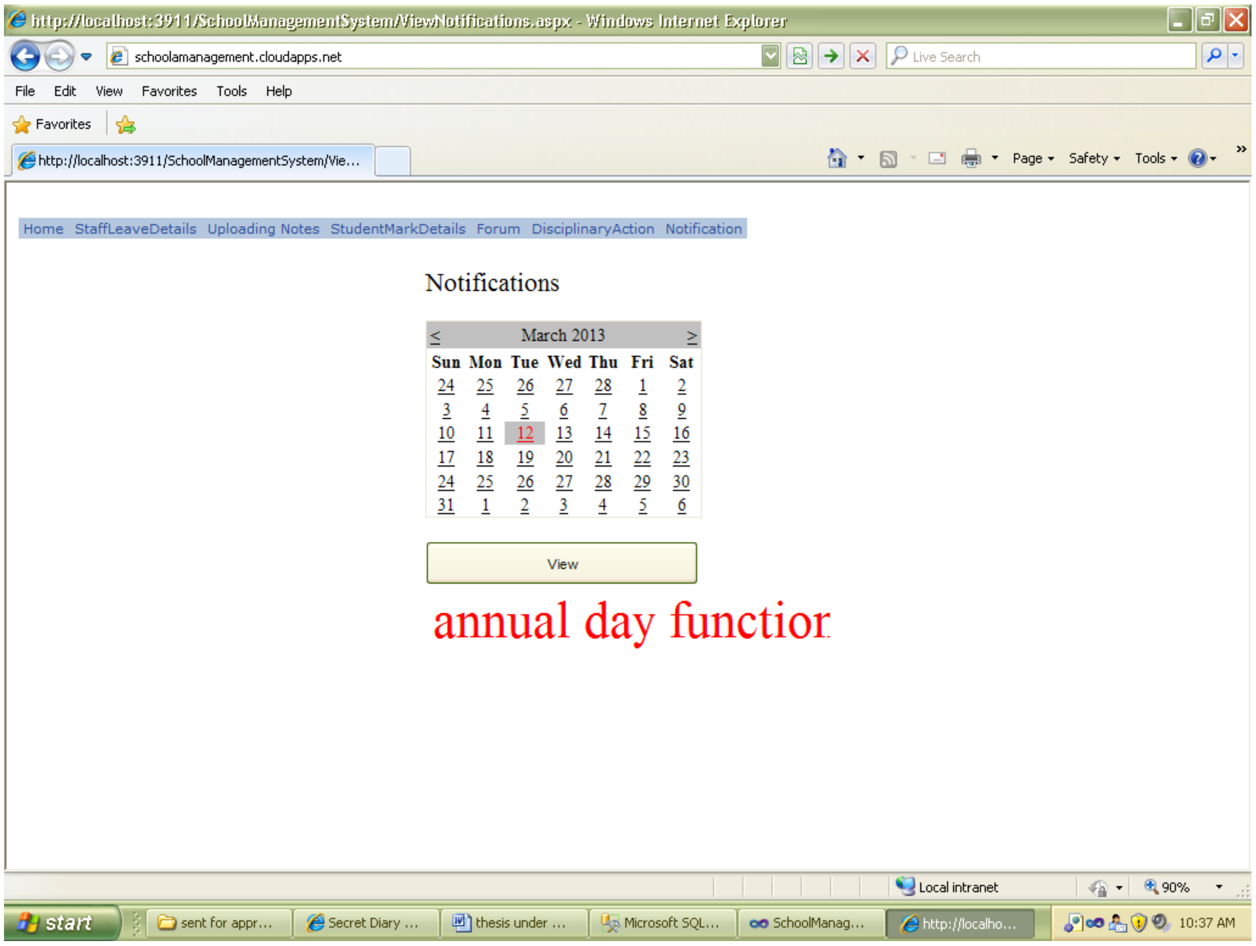
Class	-Select-
Division	
Student Name	
Parent Name	
Surname	
Residence id no	
Occupation	
Relation To Student	
If Alive	○ Alive ○ Not Alive
Box N.O	
Province	-Select-
District	
Town	
PinNo	
Home N.O	
Cell N.O	
E-Mail Id	

Creating a notification


http://localhost:3911/SchoolManagementSystem/Notification.aspx - Windows Internet Explorer
schoolmanagement.cloudapps.net
File Edit View Favorites Tools Help
HOME
STAFF DETAILS
STUDENT DETAILS
PARENTDETAILS
SETTINGS
NOTIFICATIONS
Notifications
March 2013
annual day
Create
View

Sun	Mon	Tue	Wed	Thu	Fri	Sat
24	25	26	27	28	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

Viewing a notification

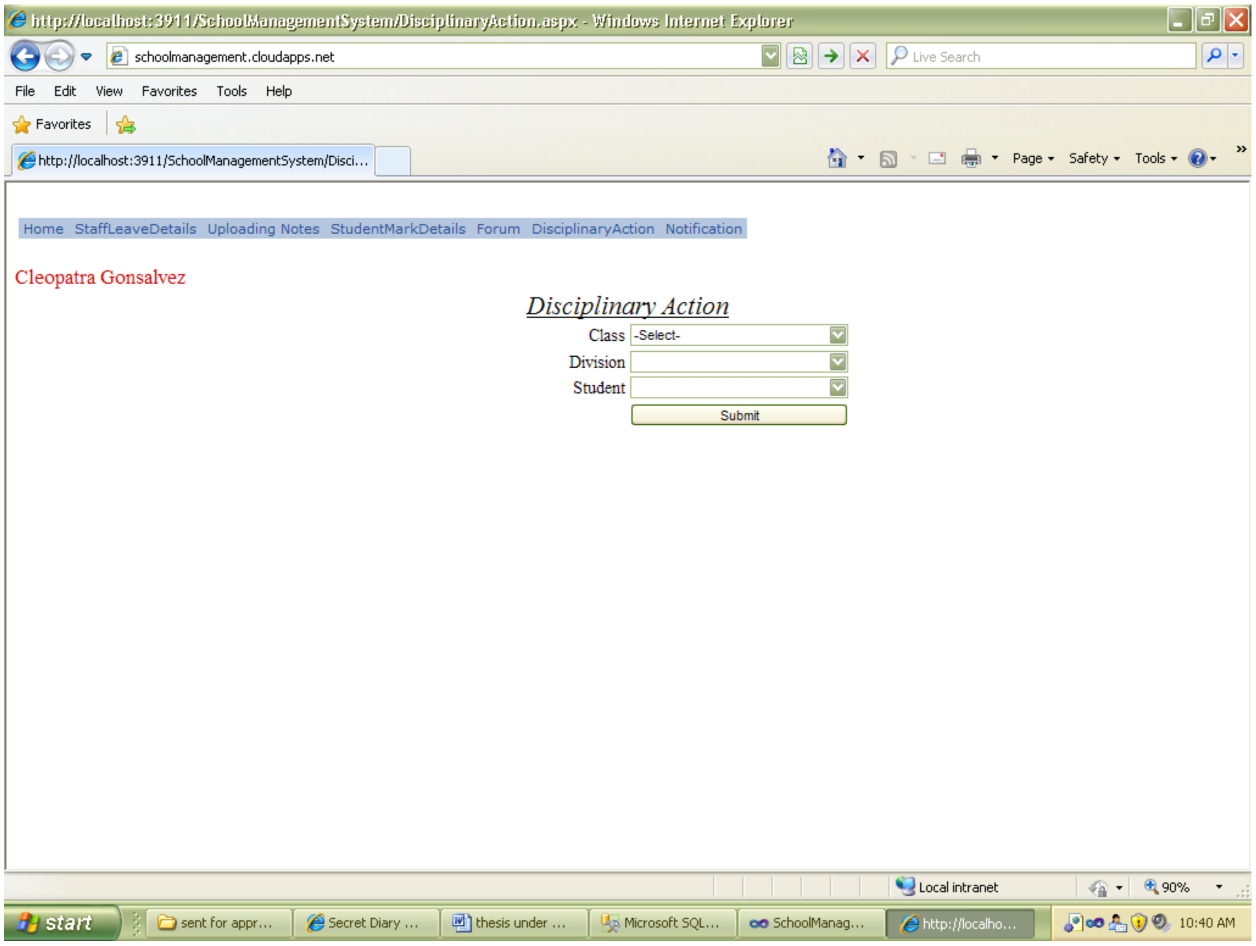

http://localhost:3911/SchoolManagementSystem/ViewNotifications.aspx - Windows Internet Explorer
schoolamanagement.cloudapps.net
File Edit View Favorites Tools Help
Home StaffLeaveDetails Uploading Notes StudentMarkDetails Forum DisciplinaryAction Notification

Notifications

<			March 2013			>
Sun	Mon	Tue	Wed	Thu	Fri	Sat
24	25	26	27	28	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

annual day functior

Registration of disciplinary action

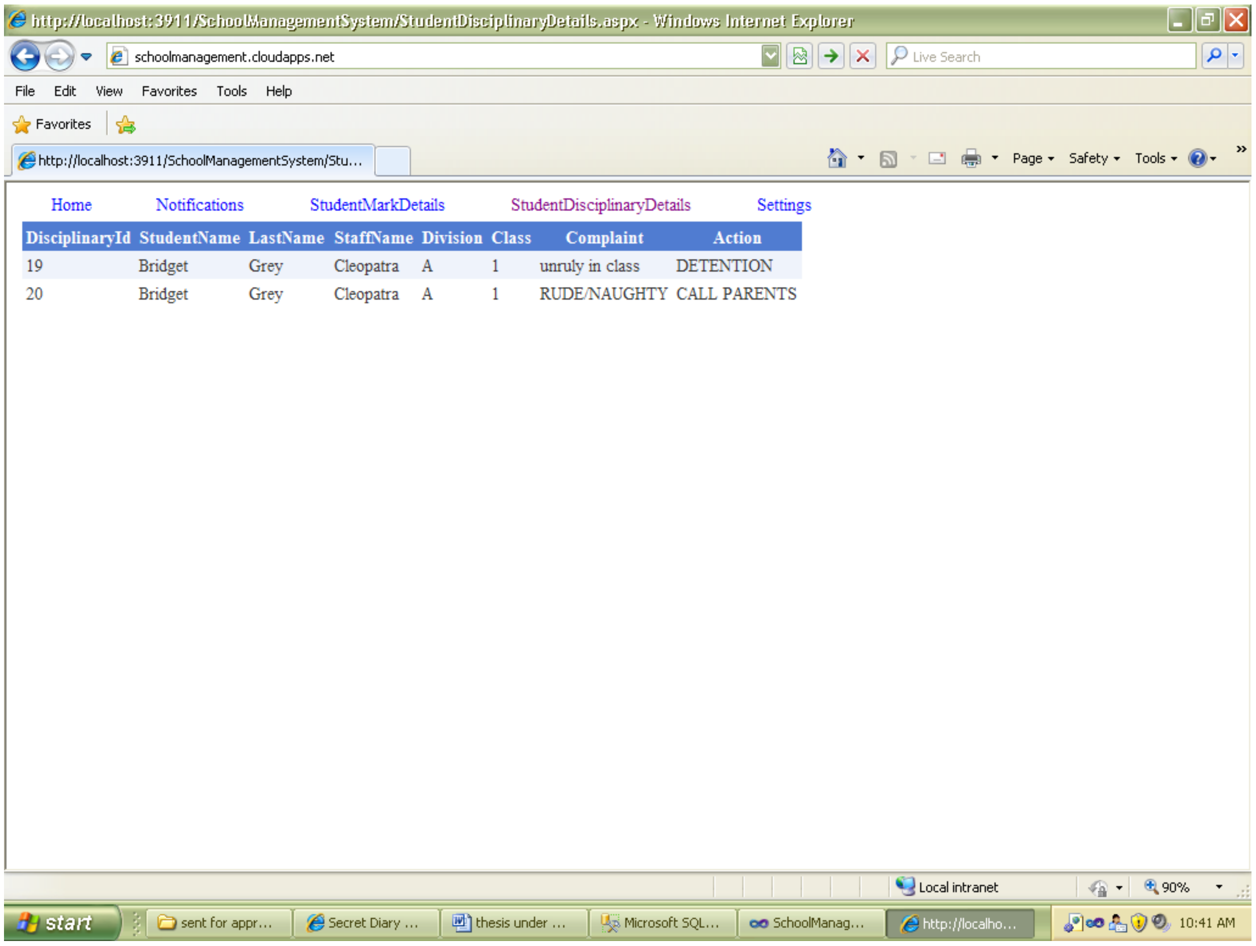

http://localhost:3911/SchoolManagementSystem/DisciplinaryAction.aspx - Windows Internet Explorer
schoolmanagement.cloudapps.net
Live Search
File Edit View Favorites Tools Help
Favorites
http://localhost:3911/SchoolManagementSystem/Disci...
Page Safety Tools
Home StaffLeaveDetails Uploading Notes StudentMarkDetails Forum DisciplinaryAction Notification
Local intranet
90%
start
sent for appr...
Secret Diary ...
thesis under ...
Microsoft SQL...
SchoolManag...
http://localho...
10:40 AM

Cleopatra Gonsalvez

Disciplinary Action

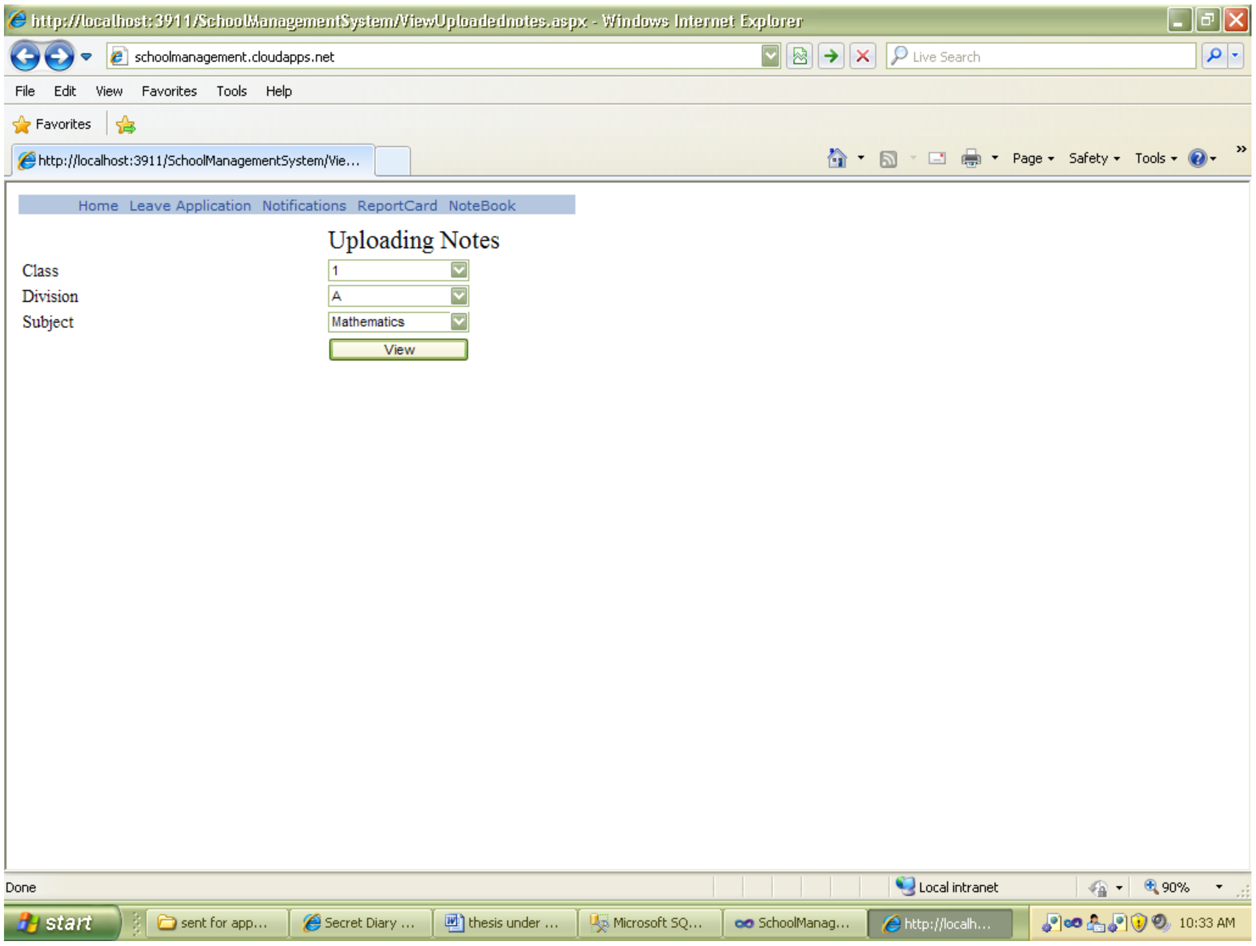
Class	-Select-
Division	
Student	
	Submit

Viewing disciplinary details


http://localhost:3911/SchoolManagementSystem/StudentDisciplinaryDetails.aspx - Windows Internet Explorer
schoolmanagement.cloudapps.net
Live Search
File Edit View Favorites Tools Help
Favorites
Home Notifications StudentMarkDetails StudentDisciplinaryDetails Settings
Local intranet
90%
start
10:41 AM

DisciplinaryId	StudentName	LastName	StaffName	Division	Class	Complaint	Action
19	Bridget	Grey	Cleopatra	A	1	unruly in class	DETENTION
20	Bridget	Grey	Cleopatra	A	1	RUDE/NAUGHTY	CALL PARENTS

Uploading student marks

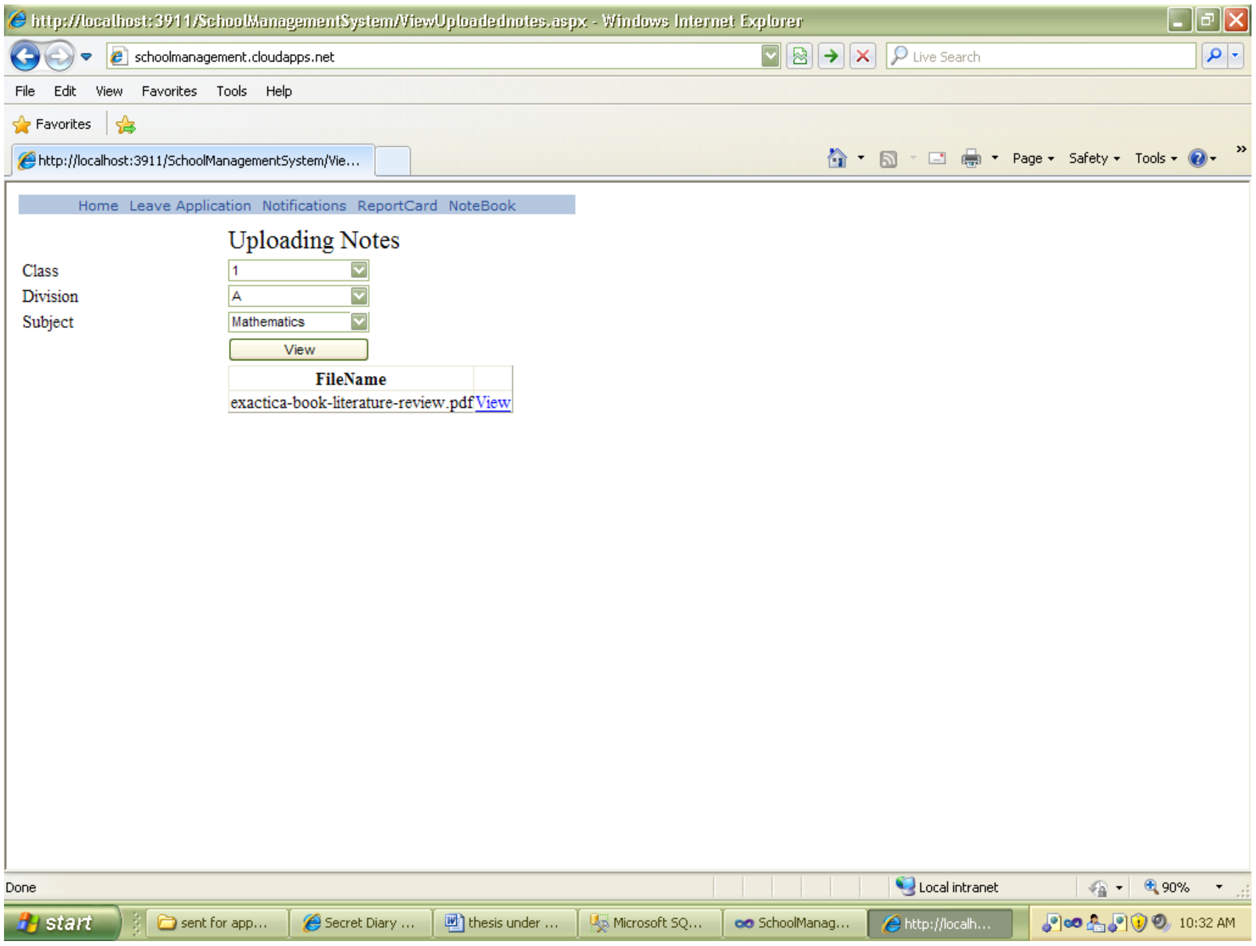

http://localhost:3911/SchoolManagementSystem/ViewUploadednotes.aspx - Windows Internet Explorer
schoolmanagement.cloudapps.net
Home Leave Application Notifications ReportCard NoteBook

Uploading Notes

Class	1
Division	A
Subject	Mathematics

View
Done
Local intranet
90%
start
10:33 AM

Viewing uploaded notes


http://localhost:3911/SchoolManagementSystem/ViewUploadednotes.aspx - Windows Internet Explorer
schoolmanagement.cloudapps.net
File Edit View Favorites Tools Help
Home Leave Application Notifications ReportCard NoteBook

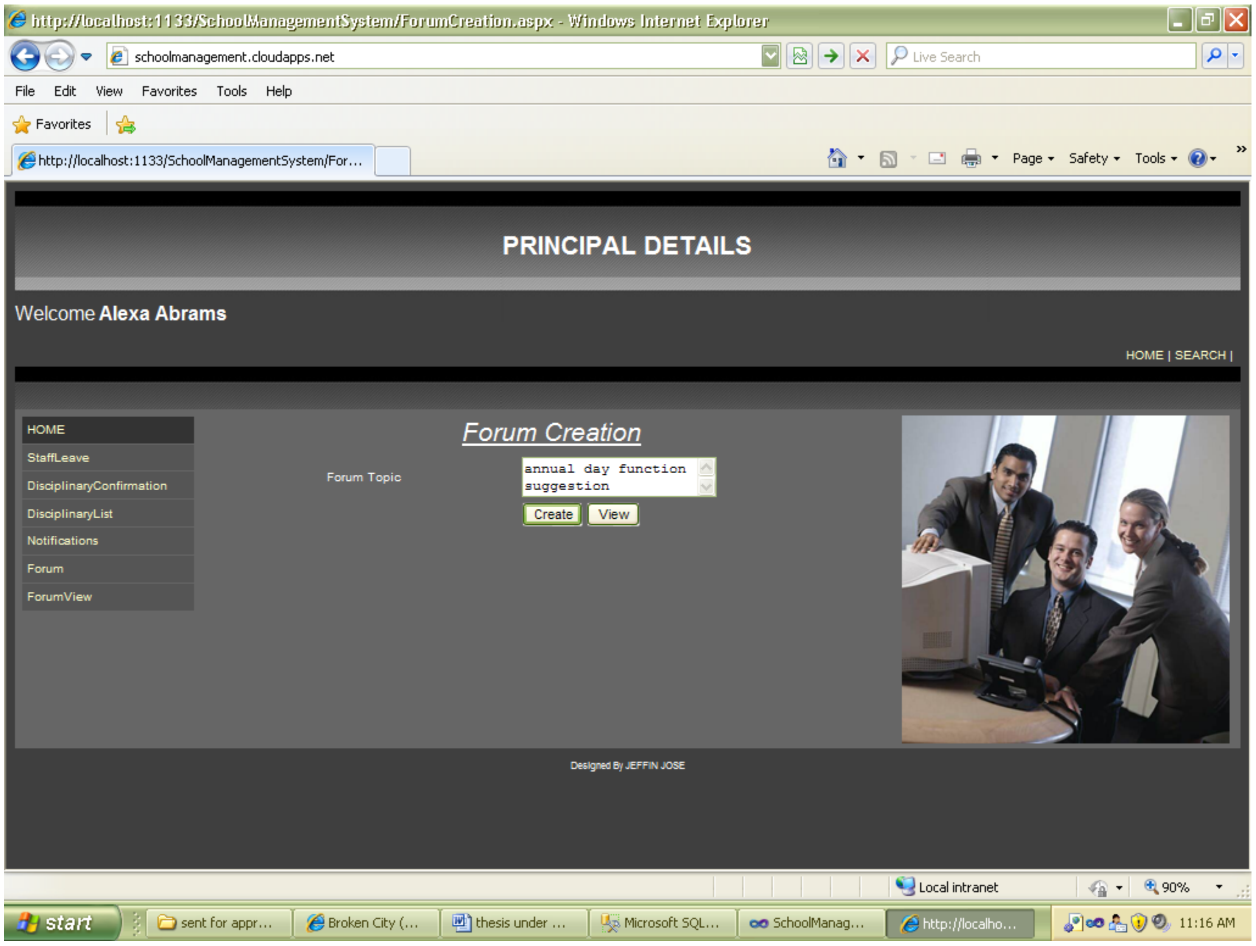
Uploading Notes

Class	1
Division	A
Subject	Mathematics

View

FileName	
exactica-book-literature-review.pdf	View

Creating a forum


http://localhost:1133/SchoolManagementSystem/ForumCreation.aspx - Windows Internet Explorer
schoolmanagement.cloudapps.net
File Edit View Favorites Tools Help
PRINCIPAL DETAILS
HOME | SEARCH |
HOME
StaffLeave
DisciplinaryConfirmation
DisciplinaryList
Notifications
Forum
ForumView
Create
View

Welcome **Alexa Abrams**

Forum Creation

Forum Topic: annual day function suggestion

Designed By JEFFIN JOSE

APPENDIX 2– CODING

User login form

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;

public partial class userlogin : System.Web.UI.Page
{
    Datalayer d_obj = new Datalayer();
    Businesslayer b_obj = new Businesslayer();
    protected void Page_Load(object sender, EventArgs e)
    {

    }
    protected void btnlogin_Click(object sender, EventArgs e)
    {

    }
    protected void ImageButton1_Click(object sender, ImageClickEventArgs e)

{
        if (txtuname.Text == "ADMIN" && txtpwd.Text == "ADMIN")
        {
            Response.Redirect("NotApprovedschools.aspx");
        }

        b_obj.uname = txtuname.Text;
        b_obj.pwd = txtpwd.Text;
        Session["b_obj.uname"] = b_obj.uname;
        Session["b_obj.pwd"] = b_obj.pwd;
        DataTable dB = new DataTable();
        dB = d_obj.checkuserlogin(b_obj);
        if (dB.Rows.Count > 0)
        {
            Session["Usertypeid"] = dB.Rows[0][0].ToString();
            Session["usertype"] = dB.Rows[0][1].ToString();
            Session["userid"] = dB.Rows[0][2].ToString();
            if (Session["usertype"].ToString() == "SCHOOLADMIN")
            {
                Session["s_id"] = dB.Rows[0][2].ToString();
                b_obj.userid = Convert.ToInt32(Session["s_id"]);
                DataTable dts = new DataTable();
                dts = d_obj.checkschoologin(b_obj);
                if (dts.Rows.Count >= 0)
                {
                    Session["s_id"] = dts.Rows[0][2].ToString();

                    Session["regcode"] = dts.Rows[0][3].ToString();
                    Response.Redirect("Schoolhome.aspx");
                }
            }
        }


        DataTable da = d_obj.UserLogin_VALIDATE();
        //int flag = 0;
        for (int i = 0; i < da.Rows.Count; i++)
        {
            string A = da.Rows[i]["Username"].ToString();
            string B = da.Rows[i]["Password"].ToString();

            string a = txtuname.Text;
            string b = txtpwd.Text;
            if (txtpwd.Text == " " || txtuname.Text == " ")
            {
                ScriptManager.RegisterStartupScript(Page, this.GetType(), "write", "alert('Please Enter
user name and password');", true);
            }

            if (A == a && B == b)
            {

                if (txtuname.Text == "ADMIN" && txtpwd.Text == "ADMIN")
                {
                    Response.Redirect("NotApprovedschools.aspx");
                }
                //if (b_obj.uname != txtuname.Text && b_obj.uname != txtuname.Text)
                //{
                //    ScriptManager.RegisterStartupScript(Page, this.GetType(), "write",
"alert('UserName and Password Mismatching');", true);
                //}
                b_obj.uname = txtuname.Text;
b_obj.pwd = txtpwd.Text;
Session["b_obj.uname"] = b_obj.uname;
Session["b_obj.pwd"] = b_obj.pwd;
DataTable dt = new DataTable();
dt = d_obj.checkuserlogin(b_obj);
if (dt.Rows.Count > 0)
{
   Session["Usertypeid"] = dt.Rows[0][0].ToString();
   Session["usertype"] = dt.Rows[0][1].ToString();
   Session["userid"] = dt.Rows[0][2].ToString();
   if (Session["usertype"].ToString() == "SCHOOLADMIN")
   {
      Session["s_id"] = dt.Rows[0][2].ToString();
      b_obj.userid = Convert.ToInt32(Session["s_id"]);
      DataTable dts = new DataTable();
      dts = d_obj.checkschoologin(b_obj);
      if (dts.Rows.Count >= 0)
      {
         Session["s_id"] = dts.Rows[0][2].ToString();

         Session["regcode"] = dts.Rows[0][3].ToString();
         Response.Redirect("Schoolhome.aspx");
      }
   }
   if (Session["usertype"].ToString() == "PARENT")
   {
      b_obj.userid = Convert.ToInt32(Session["userid"]);
      DataTable dtp = new DataTable();
      dtp = d_obj.fetch_parentdetailsusinguserid(b_obj);
      if (dtp.Rows.Count >= 0)
      {
         Session["studentid"] = dtp.Rows[0][0].ToString();
         Session["parentname"] = dtp.Rows[0][1].ToString();
         Session["parentsurname"] = dtp.Rows[0][2].ToString();
         Response.Redirect("ParentHome.aspx");
      }
   }

   if (Session["usertype"].ToString() == "STUDENT")
   {
      b_obj.userid = Convert.ToInt32(Session["userid"]);
      DataTable dts = new DataTable();
      dts = d_obj.stp_fetchcorrespondingstudent(b_obj);
      if (dts.Rows.Count >= 0)
      {
         Session["s_id"] = dts.Rows[0][0].ToString();
```

```
                b_obj.School_id = Convert.ToInt32(Session["s_id"]);
                Session["studentname"] = dts.Rows[0][1].ToString();
                b_obj.student_name = (Session["studentname"]).ToString();
                Response.Redirect("StudentHome.aspx");
            }


        }
    }

    if (Session["usertype"].ToString() == "STAFF")
    {
         b_obj.userid = Convert.ToInt32(Session["userid"]);
        DataTable dt2 = new DataTable();
        if (dt2.Rows.Count >= 0)
        {
            dt2 = d_obj.fetchdesignationfromstaffdetails(b_obj);
            Session["StaffId"] = dt2.Rows[0][0].ToString();
            Session["DesignationID"] = dt2.Rows[0][1].ToString();
            Session["Designation"] = dt2.Rows[0][2].ToString();
            Session["schoolid"] = dt2.Rows[0][3].ToString();
            Session["staffname"] = dt2.Rows[0][4].ToString();
            Session["staffsurname"] = dt2.Rows[0][5].ToString();
            b_obj.staff_details_id = Convert.ToInt32(Session["StaffId"]);
            b_obj.School_id = Convert.ToInt32(Session["schoolid"]);
            b_obj.staff_name = Session["staffname"].ToString();
            b_obj.staff_surname = Session["staffsurname"].ToString();
            if (Session["Designation"].ToString() == "PRINCIPAL")
            {
                Response.Redirect("Principal.aspx");
            }
            if (Session["Designation"].ToString() == "TEACHER")
            {

                Response.Redirect("Staffhome.aspx");
            }
            if (Session["Designation"].ToString() == "HOD")
            {
                Response.Redirect("Hod.aspx");
            }
        }
    }

}
```

```
    }

 }

    protected void LinkButton1_Click(object sender, EventArgs e)
    {

    }
}
```

School registration form

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;

public partial class SchoolRegistration : System.Web.UI.Page
{
   Datalayer d_obj = new Datalayer();
   Businesslayer b_obj = new Businesslayer();
   protected void Page_Load(object sender, EventArgs e)
   {
      if (!IsPostBack)
      {
         fetchdynamic();
         //fetchusertype();
      }
   }
   public void fetchdynamic()
   {
      DataTable dt = new DataTable();
      dt.Columns.Add("SchoolName");
      dt.Columns.Add("BoxNo");
      dt.Columns.Add("Town");
      dt.Columns.Add("District");
      dt.Columns.Add("Province");
      dt.Columns.Add("PinNumber");
      dt.Columns.Add("PhoneNumber");
      dt.Columns.Add("E-mail");
      dt.Columns.Add("ContactPerson");
      dt.Columns.Add("RegistrationCode");
      dt.Columns.Add("UserType");
      dt.Columns.Add("UserTypeId");

      Session["schooldetails"] = dt;

   }
   public void binddynamic()
   {
    lblphoneerror.Visible = false;
    DataTable dt = (DataTable)Session["schooldetails"];
    DataRow dr = dt.NewRow();
    dr["SchoolName"] = txtschoolname.Text;
    dr["BoxNo"] = txtboxno.Text;
    dr["Town"] = txttown.Text;
    dr["District"] = txtdistrict.Text;
    dr["Province"] = txtprovince.Text;
    dr["PinNumber"] = txtpin_no.Text;
    dr["PhoneNumber"] = txtphone.Text;
    dr["E-mail"] = txtemail.Text;
    dr["ContactPerson"] = txtcontactperson.Text;
    dr["RegistrationCode"] = txtregistrationcode.Text;
    dr["UserType"] = "SCHOOLADMIN";
    dr["UserTypeId"] = "2";
    Session["UsertypeId"] = dr["UserTypeId"];
    dt.Rows.Add(dr);
    Session["schooldetails"] = dt;
    if (dt.Rows.Count > 0)
    {
        GridView1.DataSource = dt;
        GridView1.DataBind();

    }

}
protected void btnsubmit_Click(object sender, EventArgs e)
{
    int phonelength = txtphone.Text.Length;
    if (phonelength < 10)
    {
        lblphoneerror.Visible = true;

    }
    else
    {
        byte[] img = FileUpload1.FileBytes;
        Session["imag"] = img;
        binddynamic();
        clearfields();

    }

}
public void clearfields()
{
    txtschoolname.Text = "";
    txtboxno.Text = "";
    txttown.Text = "";
    txtdistrict.Text = "";
    txtprovince.Text = "";
    txtpin_no.Text = "";
    txtphone.Text = "";
    txtemail.Text = "";
    txtcontactperson.Text = "";

}
protected void txtcontactperson_TextChanged(object sender, EventArgs e)
{

}
protected void GridView1_RowEditing(object sender, GridViewEditEventArgs e)
{
    GridView1.EditIndex = e.NewEditIndex;
    DataTable dt = (DataTable)Session["schooldetails"];
    GridView1.DataSource = dt;
    GridView1.DataBind();

}
protected void GridView1_RowUpdating(object sender, GridViewUpdateEventArgs e)
{
    DataTable dt = (DataTable)Session["schooldetails"];
    string schoolname = ((TextBox)GridView1.Rows[e.RowIndex].Cells[0].Controls[0]).Text;
    dt.Rows[e.RowIndex]["SchoolName"] = schoolname;
    string boxno = ((TextBox)GridView1.Rows[e.RowIndex].Cells[1].Controls[0]).Text;
    dt.Rows[e.RowIndex]["BoxNo"] = boxno;
    string pinno = ((TextBox)GridView1.Rows[e.RowIndex].Cells[5].Controls[0]).Text;
    dt.Rows[e.RowIndex]["PinNumber"] = pinno;
    string phone = ((TextBox)GridView1.Rows[e.RowIndex].Cells[6].Controls[0]).Text;
    dt.Rows[e.RowIndex]["PhoneNumber"] = phone;
    string Contact = ((TextBox)GridView1.Rows[e.RowIndex].Cells[8].Controls[0]).Text;
    dt.Rows[e.RowIndex]["ContactPerson"] = Contact;
    string RegCode = ((TextBox)GridView1.Rows[e.RowIndex].Cells[9].Controls[0]).Text;
    dt.Rows[e.RowIndex]["RegistrationCode"] = RegCode;
    Session["schooldetails"] = dt;
    GridView1.EditIndex = -1;
    GridView1.DataSource = dt;
    GridView1.DataBind();

}

protected void btnsubmitt_Click(object sender, EventArgs e)
```

```
    {

        Button b = (Button)sender;
        GridViewRow rw = (GridViewRow)b.NamingContainer;
        DataTable dt = new DataTable();
        dt = (DataTable)Session["schooldetails"];
        b_obj.stat = "not approved";
        b_obj.school_name = GridView1.Rows[rw.RowIndex].Cells[0].Text;
        b_obj.box_no = GridView1.Rows[rw.RowIndex].Cells[1].Text;
        b_obj.town = GridView1.Rows[rw.RowIndex].Cells[2].Text;
        b_obj.district = GridView1.Rows[rw.RowIndex].Cells[3].Text;
        b_obj.province = GridView1.Rows[rw.RowIndex].Cells[4].Text;
        b_obj.pin_number = GridView1.Rows[rw.RowIndex].Cells[5].Text;
        b_obj.phone = GridView1.Rows[rw.RowIndex].Cells[6].Text;
        b_obj.emailid = GridView1.Rows[rw.RowIndex].Cells[7].Text;
        b_obj.contact_person = GridView1.Rows[rw.RowIndex].Cells[8].Text;
        b_obj.RegCode = GridView1.Rows[rw.RowIndex].Cells[9].Text;
        b_obj.user_typeid = Convert.ToInt32(dt.Rows[0][11].ToString());
        b_obj.image = (byte[])Session["imag"];
        object id = new object();
        id = d_obj.insert_school(b_obj);
        Session["school_id"] = id;

        //}
        ScriptManager.RegisterStartupScript(Page, this.GetType(), "write", "alert('School details
submitted successfully');", true);
    }
    protected void GridView1_RowCancelingEdit(object sender, GridViewCancelEditEventArgs
e)
    {
        DataTable dt = new DataTable();
        dt = (DataTable)Session["schooldetails"];
        GridView1.EditIndex = -1;
        GridView1.DataSource = dt;
        GridView1.DataBind();
    }
    protected void GridView1_RowDeleting(object sender, GridViewDeleteEventArgs e)
    {
        DataTable dt = new DataTable();
        dt = (DataTable)Session["schooldetails"];
        dt.Rows[e.RowIndex].Delete();
        GridView1.DataSource = dt;
        GridView1.DataBind();
    }
    protected void GridView1_SelectedIndexChanged(object sender, EventArgs e)
    {
    }
}
```

Disciplinary action form

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;

public partial class Faculty : System.Web.UI.Page
{
    Datalayer d_obj = new Datalayer();
    Businesslayer b_obj = new Businesslayer();
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            fetchclass();
            fetch_dynamic();
            b_obj.staff_details_id = Convert.ToInt32(Session["StaffId"]);
            b_obj.School_id = Convert.ToInt32(Session["schoolid"]);
            b_obj.staff_name = Session["staffname"].ToString();
            b_obj.staff_surname = Session["staffsurname"].ToString();
            Label2.Visible = true;
            Label2.Text = b_obj.staff_name.ToString() + " " + b_obj.staff_surname.ToString();

        }
    }
    public void fetchclass()
    {
        DataTable dt = new DataTable();
        b_obj.School_id = Convert.ToInt32(Session["schoolid"]);
        dt = d_obj.fetch_class(b_obj);
        ddlclass.DataSource = dt;
        ddlclass.DataValueField = "class_creation_id";
        ddlclass.DataTextField = "class";
        ddlclass.DataBind();
        ddlclass.Items.Insert(0, "-Select-");

    }
    public void fetch_dynamic()
    {
```

```
    DataTable dt = new DataTable();
    dt.Columns.Add("StudentId");
    dt.Columns.Add("Class");
    dt.Columns.Add("Division");
    dt.Columns.Add("StudentName");
    dt.Columns.Add("Complaint");
    Session["Disciplinary"] = dt;

}
protected void ddlclass_SelectedIndexChanged(object sender, EventArgs e)
{
    DataTable dt = new DataTable();
    b_obj.class_creation_id = Convert.ToInt16(ddlclass.SelectedValue);
    dt = d_obj.fetch_Divisionfromclass(b_obj);
    ddldivision.DataSource = dt;
    ddldivision.DataValueField = "student_div_id";
    ddldivision.DataTextField = "div";
    ddldivision.DataBind();
    ddldivision.Items.Insert(0, "-Select-");
}
protected void ddldivision_SelectedIndexChanged(object sender, EventArgs e)
{
    DataTable dt = new DataTable();
    b_obj.studentdivision_id = Convert.ToInt16(ddldivision.SelectedValue);
    dt = d_obj.fetch_StudentnameFromclass(b_obj);
    ddlstudent.DataSource = dt;
    ddlstudent.DataValueField = "student_id";
    ddlstudent.DataTextField = "stud_name";
    ddlstudent.DataBind();
    ddlstudent.Items.Insert(0, "-Select-");
}
protected void Button1_Click(object sender, EventArgs e)
{
    DataTable dt = (DataTable)Session["Disciplinary"];
    DataRow dr = dt.NewRow();
    dr[0] = Convert.ToInt16(ddlstudent.SelectedValue);
    dr[1] = ddlclass.SelectedItem.Text;
    dr[2] = ddldivision.SelectedItem.Text;
    dr[3] = ddlstudent.SelectedItem.Text;
    dt.Rows.Add(dr);
    Session["Disciplinary"] = dt;
    GridView1.DataSource = dt;
    GridView1.DataBind();
    btnsave.Visible = true;
}
protected void GridView1_RowDeleting(object sender, GridViewDeleteEventArgs e)
```

```
    {
        DataTable dt = new DataTable();
        dt = (DataTable)Session["Disciplinary"];
        dt.Rows[e.RowIndex].Delete();
        GridView1.DataSource = dt;
        GridView1.DataBind();
    }
    protected void btnsave_Click(object sender, EventArgs e)
    {
        DataTable dt = new DataTable();
        dt = (DataTable)Session["Disciplinary"];
        b_obj.staff_details_id = Convert.ToInt32(Session["StaffId"]);
        b_obj.student_id =Convert.ToInt32 ( dt.Rows[0][0].ToString());
        string comp = ((TextBox)GridView1.Rows[0].Cells[4].FindControl("TextBox1")).Text;
        b_obj.Disciplinary_complaint = comp;
        b_obj.Disciplinary_action = "Not Set";
        b_obj.Disciplinary_approval = "Not Approved";
        d_obj.insert_disciplinarydetails(b_obj);
    }
}
```

Forum creation form

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;

public partial class ForumCreation : System.Web.UI.Page
{
    Datalayer d_obj = new Datalayer();
    Businesslayer b_obj = new Businesslayer();
    DateTime dt1 = System.DateTime.Now;

    protected void Page_Load(object sender, EventArgs e)
    {
        fetch_dynamic();

    }
     public void fetch_dynamic()
    {
        DataTable dt = new DataTable();
        dt.Columns.Add("ForumName");
        dt.Columns.Add("Date Of Creation");
        Session["ForumDetails"] = dt;
    }

     protected void btn_create_Click(object sender, EventArgs e)
     {
         if (txt_topic.Text == "")
        {
            ScriptManager.RegisterStartupScript(Page, this.GetType(), "write", "alert('Please Enter a
Forum Title')", true);
        }
        else
        {


            b_obj.forum_name = txt_topic.Text;
            b_obj.forum_date_of_creation= dt1.ToShortDateString();
            b_obj.staff_details_id = Convert.ToInt32(Session["StaffId"]);
            d_obj.insert_forum_creation(b_obj);
            txt_topic.Text = "";


        }
    }



      protected void btn_view_Click1(object sender, EventArgs e)
    {
        DataTable dt = d_obj.stp_fetch_forum_details();

        if (dt.Rows.Count > 0)
        {
            GridView1.DataSource = dt;
            GridView1.DataBind();

        }
    }
    protected void GridView1_RowDeleting(object sender, GridViewDeleteEventArgs e)
    {
        string id = GridView1.DataKeys[e.RowIndex].Values[0].ToString();
        b_obj.forum_id = Convert.ToInt32(id);

        d_obj.delete_forum(b_obj);
        DataTable dt = d_obj.stp_fetch_forum_details();
        if (dt.Rows.Count >= 0)
        {
            GridView1.DataSource = dt;
            GridView1.DataBind();

        }
    }
}
```

Notification form

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;

public partial class Notification : System.Web.UI.Page
{
    Datalayer d_obj = new Datalayer();
    Businesslayer b_obj = new Businesslayer();
    protected void Page_Load(object sender, EventArgs e)
    {

    }
    protected void Calendar1_DayRender(object sender, DayRenderEventArgs e)
    {
        string date = Convert.ToString(e.Day.Date.ToShortDateString());
        DataTable dt = d_obj.fetch_notificationdays();
        for (int i = 0; i < dt.Rows.Count; i++)
        {
            if (dt.Rows[i]["day"].ToString() == date)
            {

                e.Cell.ForeColor = System.Drawing.Color.Red;
            }
        }
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        if (Calendar1.SelectedDate.ToShortDateString() != "1/1/0001" && TextBox1.Text != "")
        {
            b_obj.notification_day = Calendar1.SelectedDate.ToShortDateString ();
            b_obj.notification_event = TextBox1.Text;
            d_obj.insertNotification(b_obj);
        }
        else
        {
            ScriptManager.RegisterStartupScript(Page, this.GetType(), "write", "alert('Cannot create 
notification, make sure date is selected and event is entered')", true);
        }
    }
    protected void GridView1_SelectedIndexChanged(object sender, EventArgs e)
    {

    }
    protected void GridView1_RowDeleting(object sender, GridViewDeleteEventArgs e)
    {

    }
    protected void GridView1_RowEditing(object sender, GridViewEditEventArgs e)
    {
        GridView1.EditIndex = e.NewEditIndex;
        b_obj.notification_day = Calendar1.SelectedDate.ToShortDateString();
        DataTable dt = d_obj.fetchnotificationdetails(b_obj);
        GridView1.DataSource = dt;
        GridView1.DataBind();

    }
    protected void GridView1_RowUpdating(object sender, GridViewUpdateEventArgs e)
    {
        string dk = GridView1.DataKeys[e.RowIndex].Values[0].ToString();
        string eve=((TextBox )GridView1 .Rows [e.RowIndex ].Cells [2].Controls [0]).Text ;
        b_obj.notification_id =Convert .ToInt32 ( dk);
        b_obj.notification_event = eve;
        d_obj.update_notificationevent(b_obj);
        GridView1.EditIndex = -1;
        b_obj.notification_day = Calendar1.SelectedDate.ToShortDateString();
        DataTable dt = d_obj.fetchnotificationdetails(b_obj);
        GridView1.DataSource = dt;
        GridView1.DataBind();
    }
    protected void TextBox1_TextChanged(object sender, EventArgs e)
    {

    }
    protected void Button2_Click(object sender, EventArgs e)
    {
        b_obj.notification_day  = Calendar1.SelectedDate.ToShortDateString();
        DataTable dt = d_obj.fetchnotificationdetails(b_obj );
        if (dt.Rows.Count == 0)
        {
            ScriptManager.RegisterStartupScript(Page, this.GetType(), "write", "alert('No events
Entered')", true);
            GridView1.Visible = false;
        }

        else
        {
            GridView1.Visible = true;
            for (int i = 0; i < dt.Rows.Count; i++)
            {
                GridView1.DataSource = dt;
                GridView1.DataBind();
            }
        }
    }
    protected void GridView1_RowCancelingEdit(object sender, GridViewCancelEditEventArgs
e)
    {
        GridView1.EditIndex = -1;
        b_obj.notification_day = Calendar1.SelectedDate.ToShortDateString();
        DataTable dt = d_obj.fetchnotificationdetails(b_obj);
        GridView1.DataSource = dt;
        GridView1.DataBind();
    }
}
```

View notification

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;

public partial class ViewNotifications : System.Web.UI.Page
{
    Datalayer d_obj = new Datalayer();
    Businesslayer b_obj = new Businesslayer();
    protected void Page_PreInit(object sender, EventArgs e)
    {


    }
    protected void Page_Load(object sender, EventArgs e)
    {

    }
    protected void Calendar1_DayRender(object sender, DayRenderEventArgs e)
    {
        string date = Convert.ToString(e.Day.Date.ToShortDateString());
        DataTable dt = d_obj.fetch_notificationdays();
        for (int i = 0; i < dt.Rows.Count; i++)
        {
            if (dt.Rows[i]["day"].ToString() == date)
            {

                e.Cell.ForeColor = System.Drawing.Color.Red;
            }
        }
    }
    protected void Button2_Click(object sender, EventArgs e)
     {
        string eve="";
        b_obj.notification_day = Calendar1.SelectedDate.ToShortDateString();
        DataTable dt = d_obj.fetchnotificationdetails(b_obj);
        if (dt.Rows.Count == 0)
        {
            ScriptManager.RegisterStartupScript(Page, this.GetType(), "write", "alert('No events
Entered')", true);

        }

        else
        {

            for (int i = 0; i < dt.Rows.Count; i++)
            {
                eve += dt.Rows[i][2].ToString();
                int length=dt.Rows.Count;
                if (length-1 !=i)
                {
                    eve += "<br>";
                }
                Literal1.Text="<marquee scroll amount=100 ><font size=25 color=red>" + eve+
"</font></marquee>";
        }

    }
}
```

Parent/guardian details

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;

public partial class Parent_guardianDetails : System.Web.UI.Page
{
    Datalayer d_obj = new Datalayer();
    Businesslayer b_obj = new Businesslayer();
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            fetchclass();
            fetchprovince();
            fetchdynamic();
            btnsubmit.Visible = false;
        }

    }
    public void fetchdynamic()
    {
        DataTable dt = new DataTable();
        dt.Columns.Add("STUDENTID");
        dt.Columns.Add("STUDENTNAME");
        dt.Columns.Add("CLASS");
        dt.Columns.Add("DIVISION");
        //dt.Columns.Add("RELATION");
        dt.Columns.Add("NAME");
        dt.Columns.Add("SURNAME");
        dt.Columns.Add("RESIDENCE");
        dt.Columns.Add("OCCUPATION");
        dt.Columns.Add("RELATIONTO");
        dt.Columns.Add("BOXNO");
        dt.Columns.Add("PROVINCEID");
        dt.Columns.Add("PROVINCE");
        dt.Columns.Add("DISTRICTID");
        dt.Columns.Add("DISTRICT");
        dt.Columns.Add("TOWNID");
        dt.Columns.Add("TOWN");
        dt.Columns.Add("PINNO");
        Session["PARE_GUAD"] = dt;


    }
    public void fetchclass()
    {
        DataTable dt = new DataTable();
        b_obj.School_id = Convert.ToInt16(Session["s_id"]);
        dt = d_obj.fetch_class(b_obj);
        ddlclass.DataSource = dt;
        ddlclass.DataValueField = "class_creation_id";
        ddlclass.DataTextField = "class";
        ddlclass.DataBind();
        ddlclass.Items.Insert(0, "-Select-");


    }
    public void fetchprovince()
    {
        DataTable dt = new DataTable();

        dt = d_obj.fetch_province(b_obj);

        ddlprovince.DataSource = dt;
        ddlprovince.DataValueField = "province_id";
        ddlprovince.DataTextField = "province_name";
        ddlprovince.DataBind();
        ddlprovince.Items.Insert(0, "-Select-");


    }
    protected void ddlclass_SelectedIndexChanged(object sender, EventArgs e)
    {
        if (ddlclass.SelectedIndex > 0)
        {
            DataTable dt = new DataTable();
            b_obj.class_creation_id = Convert.ToInt16(ddlclass.SelectedValue);
            dt = d_obj.fetch_Divisionfromclass(b_obj);
            ddldivision.DataSource = dt;
            ddldivision.DataValueField = "student_div_id";
            ddldivision.DataTextField = "div";
            ddldivision.DataBind();
            ddldivision.Items.Insert(0, "-Select-");
        }
    }
    protected void ddldiv_SelectedIndexChanged(object sender, EventArgs e)
    {
        if (ddldivision.SelectedIndex > 0)
        {
            DataTable dt = new DataTable();
            b_obj.studentdivision_id = Convert.ToInt16(ddldivision.SelectedValue);
            dt = d_obj.fetch_StudentnameFromclass(b_obj);
            ddlstudent.DataSource = dt;
            ddlstudent.DataValueField = "student_id";
            ddlstudent.DataTextField = "stud_name";
            ddlstudent.DataBind();
            ddlstudent.Items.Insert(0, "-Select-");
        }
    }
    protected void save_Click(object sender, EventArgs e)
    {
        btnsubmit.Visible = true;
        int flag = 0;
        {
            if (ddldistrict.SelectedIndex == 0 || ddlstudent.SelectedIndex == 0 ||
ddldivision.SelectedIndex == 0 || ddlprovince.SelectedIndex == 0 || ddltown.SelectedIndex == 0)
            {

                ScriptManager.RegisterStartupScript(Page, this.GetType(), "write", "alert('Please Enter
All Important Information');", true);
                flag = 1;
            }
            if(flag==0)
            {

                b_obj.student_id = Convert.ToInt32(ddlstudent.SelectedValue);
                DataTable da = d_obj.parent_guardian_details_validate(b_obj);
                for (int j = 0; j < da.Rows.Count; j++)
                {
                    string A = da.Rows[j]["name"].ToString();
                    string B = da.Rows[j]["surname"].ToString();
                    int C = Convert.ToInt32(da.Rows[j]["residence_id_no"]);
                    int D = Convert.ToInt32(da.Rows[j]["student_id"]);

                    string a = txtname.Text;
                    string b = txtsurname.Text;
                    int c = Convert.ToInt32(txtresidence.Text);
                    int d = Convert.ToInt32(ddlstudent.SelectedValue);
                    if (A == a && B == b && C == c && D == d)
                    {
```

```
                    ScriptManager.RegisterStartupScript(Page, this.GetType(), "write",
"alert('Existing Details');", true);
                    flag = flag + 1;
                }
            }
            if (flag == 0)
            {
                b_obj.student_id = Convert.ToInt16(ddlstudent.SelectedValue);
                if (rbtparent.Checked == true)
                {
                    b_obj.pare_guad_status = rbtparent.Text;
                }
                else
                {
                    b_obj.pare_guad_status = rbtguardian.Text;
                }
                b_obj.pare_guad_name = txtname.Text;
                b_obj.pare_guad_surname = txtsurname.Text;
                b_obj.pare_gaud_residence_id = Convert.ToInt16(txtresidence.Text);
                b_obj.pare_guad_occupation = txtoccupatn.Text;
                b_obj.relatn_to_student = txtrelationtostudent.Text;
                b_obj.pare_gaud_boxno = Convert.ToInt16(txtbxno.Text);
                b_obj.pare_guad_province = Convert.ToInt32(ddlprovince.SelectedValue);
                b_obj.pare_guad_district = Convert.ToInt32(ddldistrict.SelectedValue);
                b_obj.pare_guad_town = Convert.ToInt32(ddltown.SelectedValue);
                b_obj.pare_gaud_pinno = Convert.ToInt16(txtpino.Text);
                ScriptManager.RegisterStartupScript(Page, this.GetType(), "write", "alert('PLEASE
eNSURE DETAILS ENTERED BELOW ARE CORRECT');", true);

                DataTable dt = (DataTable)Session["PARE_GUAD"];
                DataRow dr = dt.NewRow();
                //int a = Convert.ToInt32(Session["count"]);
                //dr["ID"] = a;
                dr["STUDENTID"] = ddlstudent.SelectedValue;
                dr["STUDENTNAME"] = ddlstudent.SelectedItem.Text;
                dr["CLASS"] = ddlclass.SelectedItem.Text;
                dr["DIVISION"] = ddldivision.SelectedItem.Text;
                dr["NAME"] = txtname.Text;
                dr["SURNAME"] = txtsurname.Text;
                dr["RESIDENCE"] = txtresidence.Text;
                dr["OCCUPATION"] = txtoccupatn.Text;
                dr["RELATIONTO"] = txtrelationtostudent.Text;
                dr["BOXNO"] = txtbxno.Text;
                dr["PROVINCEID"] = ddlprovince.SelectedValue;
                dr["PROVINCE"] = ddlprovince.SelectedItem.Text;
                dr["DISTRICTID"] = ddldistrict.SelectedValue;
```

```
                    dr["DISTRICT"] = ddldistrict.SelectedItem.Text;
                    dr["TOWNID"] = ddltown.SelectedValue;
                    dr["TOWN"] = ddltown.SelectedItem.Text;
                    dr["PINNO"] = txtpino.Text;
                    dt.Rows.Add(dr);
                    Session["PARE_GUAD"] = dt;
                    if (dt.Rows.Count > 0)
                    {
                        GridView1.DataSource = dt;
                        GridView1.DataBind();
                    }


                    ScriptManager.RegisterStartupScript(Page, this.GetType(), "write",
"alert('SAVED!');", true);
                }
            }
        }
    }
    protected void ddlprovince_SelectedIndexChanged(object sender, EventArgs e)
    {
        if (ddlprovince.SelectedIndex > 0)
        {
            DataTable dt = new DataTable();
            b_obj.province_id = Convert.ToInt16(ddlprovince.SelectedValue);
            dt = d_obj.fetch_district(b_obj);
            ddldistrict.DataSource = dt;
            ddldistrict.DataValueField = "district_id";
            ddldistrict.DataTextField = "district_name";
            ddldistrict.DataBind();
            ddldistrict.Items.Insert(0, "-Select-");
        }
    }
    protected void ddldistrict_SelectedIndexChanged(object sender, EventArgs e)
    {
        if (ddldistrict.SelectedIndex > 0)
        {
            DataTable dt = new DataTable();
            b_obj.district_id = Convert.ToInt32(ddldistrict.SelectedValue);
            dt = d_obj.fetch_town(b_obj);
            ddltown.DataSource = dt;
            ddltown.DataValueField = "town_id";
            ddltown.DataTextField = "town_name";
            ddltown.DataBind();
            ddltown.Items.Insert(0, "-Select-");
        }
```

```
}
protected void GridView1_RowDeleting(object sender, GridViewDeleteEventArgs e)
{
   DataTable dt = (DataTable)Session["PARE_GUAD"];
   dt.Rows[e.RowIndex].Delete();
   Session["PARE_GUAD"] = dt;
   GridView1.DataSource = dt;
   GridView1.DataBind();
}
protected void btnsubmit_Click(object sender, EventArgs e)
{
   DataTable dt = new DataTable();
   dt = (DataTable)Session["PARE_GUAD"];
   for (int i = 0; i < dt.Rows.Count; i++)
   {

      b_obj.student_id = Convert.ToInt16( dt.Rows[i][0].ToString());
      if (rbtparent.Checked == true)
      {
         b_obj.pare_guad_status = rbtparent.Text;
      }
      else
      {
         b_obj.pare_guad_status = rbtguardian.Text;
      }
      b_obj.pare_guad_name = dt.Rows[i][4].ToString();
      b_obj.pare_guad_surname=dt.Rows[i][5].ToString();
      b_obj.pare_gaud_residence_id= Convert.ToInt16( dt.Rows[i][6].ToString());
      b_obj.pare_guad_occupation=dt.Rows[i][7].ToString();
      b_obj.relatn_to_student=dt.Rows[i][8].ToString();
      b_obj.pare_gaud_boxno= Convert.ToInt16( dt.Rows[i][9].ToString());
      b_obj.pare_guad_province =  Convert.ToInt16( dt.Rows[i][10].ToString());
      b_obj.pare_guad_district = Convert.ToInt16( dt.Rows[i][12].ToString());
      b_obj.pare_guad_town = Convert.ToInt16( dt.Rows[i][14].ToString());
      b_obj.pare_gaud_pinno= Convert.ToInt16(dt.Rows[i][16].ToString());
      d_obj.insertPareGuadDeatils(b_obj);

   }
}
protected void GridView1_RowEditing(object sender, GridViewEditEventArgs e)
{
   GridView1.EditIndex = e.NewEditIndex;
   DataTable dt = (DataTable)Session["PARE_GUAD"];
   GridView1.DataSource = dt;
   GridView1.DataBind();
}
```

```
    protected void GridView1_RowCancelingEdit(object sender, GridViewCancelEditEventArgs
e)
    {
        DataTable dt = new DataTable();
        dt = (DataTable)Session["PARE_GUAD"];
        GridView1.EditIndex = -1;
        GridView1.DataSource = dt;
        GridView1.DataBind();
    }
    protected void GridView1_RowUpdating(object sender, GridViewUpdateEventArgs e)
    {
        DataTable dt = (DataTable)Session["TransferDetail"];
        string aa = ((TextBox)GridView1.Rows[e.RowIndex].Cells[4].Controls[0]).Text;
        dt.Rows[e.RowIndex]["NAME"] = aa;
        string ff = ((TextBox)GridView1.Rows[e.RowIndex].Cells[5].Controls[0]).Text;
        dt.Rows[e.RowIndex]["SURNAME"] = ff;
        string gg = ((TextBox)GridView1.Rows[e.RowIndex].Cells[6].Controls[0]).Text;
        dt.Rows[e.RowIndex]["RESIDENCE"] = gg;
        string bb = ((TextBox)GridView1.Rows[e.RowIndex].Cells[7].Controls[0]).Text;
        dt.Rows[e.RowIndex]["OCCUPATION"] = bb;
        string cc = ((TextBox)GridView1.Rows[e.RowIndex].Cells[8].Controls[0]).Text;
        dt.Rows[e.RowIndex]["RELATIONTO"] = cc;
        string dd = ((TextBox)GridView1.Rows[e.RowIndex].Cells[9].Controls[0]).Text;
        dt.Rows[e.RowIndex]["BOXNO"] = dd;
        string zz = ((TextBox)GridView1.Rows[e.RowIndex].Cells[16].Controls[0]).Text;
        dt.Rows[e.RowIndex]["PINNO"] = zz;
        Session["TransferDetail"] = dt;
        GridView1.EditIndex = -1;
        GridView1.DataSource = dt;
        GridView1.DataBind();
    }
}
```

Upload notes form

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
using System.IO ;


public partial class Uploadnotes : System.Web.UI.Page
{
    Datalayer d_obj = new Datalayer();
    Businesslayer b_obj = new Businesslayer();
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            fetchclass();
        }

    }
    public void fetchclass()
    {
        DataTable dt = new DataTable();
        //b_obj.School_id = Convert.ToInt16(Session["s_id"]);
        b_obj.School_id = 15;
        dt = d_obj.fetch_class(b_obj);
        ddlclass.DataSource = dt;
        ddlclass.DataValueField = "class_creation_id";
        ddlclass.DataTextField = "class";
        ddlclass.DataBind();
        ddlclass.Items.Insert(0, "-Select-");

    }
    protected void ddlclass_SelectedIndexChanged(object sender, EventArgs e)
    {
        DataTable dt = new DataTable();
        b_obj.class_creation_id = Convert.ToInt16(ddlclass.SelectedValue);
        dt = d_obj.fetch_Divisionfromclass(b_obj);
        ddldivision.DataSource = dt;
        ddldivision.DataValueField = "student_div_id";
        ddldivision.DataTextField = "div";
        ddldivision.DataBind();
        ddldivision.Items.Insert(0, "-Select-");
    }
    protected void ddldivision_SelectedIndexChanged(object sender, EventArgs e)
    {
        DataTable dt = new DataTable();
        b_obj.studentdivision_id = Convert.ToInt32(ddldivision.SelectedValue);
        dt = d_obj.fetch_division_correspond_subject(b_obj);
        ddlsubject.DataSource = dt;
        ddlsubject.DataValueField = "subject_creation_id";
        ddlsubject.DataTextField = "subject";
        ddlsubject.DataBind();
        ddlsubject.Items.Insert(0, "-Select-");
    }
    protected void btnupload_Click(object sender, EventArgs e)
    {
         string s = FileUpload1.FileName;
      if (s == "")
        {
        ScriptManager.RegisterStartupScript(Page, this.GetType(), "write", "alert('please select file
for upload ')", true);
        }
      else
      {
        FileInfo fi = new FileInfo(s);
        string ex = fi.Extension;

            if (ex == ".pdf" || ex == ".doc" || ex== ".jpg")
            {

                byte[] image = FileUpload1.FileBytes;
                b_obj.topic  = s;
                b_obj.notes  = image;


                byte[] arr = null;
                if (image != null)
                {
                    arr = FileUpload1.FileBytes;
                }
                b_obj.subject_creation_id = Convert.ToInt32( ddlsubject.SelectedValue);
                d_obj.insert_upload_notes (b_obj );
                ScriptManager.RegisterStartupScript(Page, this.GetType(), "write",
"alert('Uploaded')", true);
            }
        else
        {
            ScriptManager.RegisterStartupScript(Page, this.GetType(), "write", "alert('Please
upload either a pdf file or word document')", true);
        }



    }
    }
    protected void btnview_Click(object sender, EventArgs e)
    {
        DataTable dt = new DataTable();
        dt = d_obj.view_upload_notes();
        if (dt.Rows.Count > 0)
        {
            GridView1.DataSource = dt;
            GridView1.DataBind();
        }
    }
    protected void LinkButton1_Click(object sender, EventArgs e)
    {

        LinkButton lb = (LinkButton)sender;
        GridViewRow gvr = (GridViewRow)lb.NamingContainer as GridViewRow;
        string id = GridView1.DataKeys[gvr.RowIndex].Values[0].ToString();
        int id1 = Convert.ToInt32(id);
        b_obj.uploading_notes_id = id1;
        DataTable dm = new DataTable();
        dm = d_obj .view_uploadnote_details (b_obj );
        if (dm.Rows.Count > 0)
        {
            byte[] arr = (byte[])dm.Rows[0][1];
            //string notes = dm.Rows[0][1].ToString();
            string topic = dm.Rows[0][0].ToString();
            Response.ContentType = "application/octet-stream";
            Response.AddHeader("content-disposition", "attachment;filename=" + topic + "");
            Response.BinaryWrite(arr);
            Response.End();
        }
    }
    protected void GridView1_RowDeleting(object sender, GridViewDeleteEventArgs e)
    {
    }
}
```

View uploaded notes

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
using System.IO;

public partial class ViewUploadednotes : System.Web.UI.Page
{
    Datalayer d_obj = new Datalayer();
    Businesslayer b_obj = new Businesslayer();
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            fetchclass();

        }
    }
    public void fetchclass()
    {

        DataTable dt = new DataTable();
        b_obj.School_id = Convert.ToInt32(Session["s_id"]);
        dt = d_obj.fetch_class(b_obj);
        if (dt.Rows.Count > 0)
        {
            ddlclass.DataSource = dt;
            ddlclass.DataValueField = "class_creation_id";
            ddlclass.DataTextField = "class";
            ddlclass.DataBind();
            ddlclass.Items.Insert(0, "-Select-");
        }
    }
    protected void ddlclass_SelectedIndexChanged(object sender, EventArgs e)
    {
        if (ddlclass.SelectedIndex > 0)
        {
            DataTable dt = new DataTable();
            b_obj.class_creation_id = Convert.ToInt16(ddlclass.SelectedValue);
        dt = d_obj.fetch_Divisionfromclass(b_obj);
        ddldivision.DataSource = dt;
        ddldivision.DataValueField = "student_div_id";
        ddldivision.DataTextField = "div";
        ddldivision.DataBind();
        ddldivision.Items.Insert(0, "-Select-");
    }
}
protected void ddldivision_SelectedIndexChanged(object sender, EventArgs e)
{
    if (ddldivision.SelectedIndex > 0)
    {
        DataTable dt = new DataTable();
        b_obj.studentdivision_id = Convert.ToInt32(ddldivision.SelectedValue);
        dt = d_obj.fetch_division_correspond_subject(b_obj);
        ddlsubject.DataSource = dt;
        ddlsubject.DataValueField = "subject_creation_id";
        ddlsubject.DataTextField = "subject";
        ddlsubject.DataBind();
        ddlsubject.Items.Insert(0, "-Select-");
    }
}
protected void ddlsubject_SelectedIndexChanged(object sender, EventArgs e)
{

}
protected void btnupload_Click(object sender, EventArgs e)
{

}
protected void btnview_Click(object sender, EventArgs e)
{
    if (ddlclass.SelectedIndex > 0)
    {
        if (ddldivision.SelectedIndex > 0)
        {
            if (ddlsubject.SelectedIndex > 0)
            {
                DataTable dt = new DataTable();
                b_obj.subject_creation_id = Convert.ToInt32(ddlsubject.SelectedValue);
                dt = d_obj.view_uploadnote_detailsfromclass(b_obj);
                if (dt.Rows.Count > 0)
                {
                    GridView1.DataSource = dt;
                    GridView1.DataBind();
                }
                }
            }
        }
    }
    protected void LinkButton1_Click(object sender, EventArgs e)
    {

    }
    protected void GridView1_SelectedIndexChanged(object sender, EventArgs e)
    {

    }
    protected void LinkButton1_Click1(object sender, EventArgs e)
    {

        LinkButton lb = (LinkButton)sender;
        GridViewRow gvr = (GridViewRow)lb.NamingContainer as GridViewRow;
        string id = GridView1.DataKeys[gvr.RowIndex].Values[0].ToString();
        int id1 = Convert.ToInt32(id);
        b_obj.uploading_notes_id = id1;
        DataTable dm = new DataTable();
        dm = d_obj.view_uploadnote_details(b_obj);
        if (dm.Rows.Count > 0)
        {
            byte[] arr = (byte[])dm.Rows[0][1];
            //string notes = dm.Rows[0][1].ToString();
            string topic = dm.Rows[0][0].ToString();
            Response.ContentType = "application/octet-stream";
            Response.AddHeader("content-disposition", "attachment;filename=" + topic + "");
            Response.BinaryWrite(arr);
            Response.End();
        }
    }
    protected void GridView1_SelectedIndexChanged1(object sender, EventArgs e)
    {

    }
}
```

Add staff details

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;

public partial class StaffDetails : System.Web.UI.Page
{
    Datalayer d_obj = new Datalayer();
    Businesslayer b_obj = new Businesslayer();
    int admission;
    string admissionno;
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            fetchprovince();
            fetchdesignation();
        }

    }


    public void fetchprovince()
    {
        DataTable dt = new DataTable();
        b_obj.School_id = Convert.ToInt32(Session["s_id"]);

        dt = d_obj.fetch_province(b_obj);

        ddlprovince.DataSource = dt;
        ddlprovince.DataValueField = "province_id";
        ddlprovince.DataTextField = "province_name";
        ddlprovince.DataBind();
        ddlprovince.Items.Insert(0, "-Select-");


    }
     public void fetchdesignation()
      {
            DataTable dt = new DataTable();
            b_obj.School_id = Convert.ToInt32(Session["s_id"]);
            dt = d_obj.fetch_designation(b_obj);


            ddldesignation.DataSource = dt;
            ddldesignation.DataValueField = "designation_id";
            ddldesignation.DataTextField = "designation_name";
            ddldesignation.DataBind();
            ddldesignation.Items.Insert(0, "-Select-");

        }

    protected void submit_Click(object sender, EventArgs e)
    {
        if (ddldesignation.SelectedIndex > 0)
        {
            if (ddlprovince.SelectedIndex > 0)
            {
                if (ddldistrict.SelectedIndex > 0)
                {
                    if (ddltown.SelectedIndex > 0)
                    {
                        b_obj.RegCode = Convert.ToString(Session["regcode"]);
                        string regcod = b_obj.RegCode + "S";
                        string num = "0";
                        DateTime dtnow = System.DateTime.Now;
                        string dtmonth = dtnow.ToString("MM");
                        string year = Convert.ToString(dtnow.Year);
                        DataTable dt1 = new DataTable();
                        b_obj.School_id = Convert.ToInt16(Session["s_id"]);
                        dt1 = d_obj.fetch_staffregcode(b_obj);
                        if (dt1.Rows.Count == 0)
                        {
                            admission = Convert.ToInt32(year + dtmonth + num);
                            admissionno = Convert.ToString(regcod + admission);

                        }
                        else
                        {
                            string rg = dt1.Rows[0][0].ToString();
                            int reglength = Convert.ToInt32(rg.Length) - 11;
                            string yr = rg.Substring(5, 4);
                            string rgs = rg.Substring(9, 2);
                            string Nos = rg.Substring(11, reglength);
                            if (yr == year)
                            {
                                if (rgs == dtmonth)
                                {
                                    //string adno = dt1.Rows[0][0].ToString();

                                    Nos = (Convert.ToInt32(Nos) + 1).ToString();
                                    admission = Convert.ToInt32(year + dtmonth + Nos);
                                    admissionno = Convert.ToString(regcod + admission);
                                }
                            }
                            else
                            {
                                admission = Convert.ToInt32(year + dtmonth + num);
                                admissionno = Convert.ToString(regcod + admission);
                            }
                        }

                        b_obj.School_id = Convert.ToInt16(Session["s_id"]);
                        b_obj.staff_name = txtname.Text;
                        b_obj.staff_surname = txtsurnm.Text;
                        b_obj.staff_maidenname = txtmaidnnam.Text;
                        DataTable dtusertype = new DataTable();
                        b_obj.staff_designation = Convert.ToInt32(ddldesignation.SelectedValue);
                        dtusertype = d_obj.fetch_usertypefromDesignation(b_obj);
                        b_obj.user_typeid = Convert.ToInt32(dtusertype.Rows[0][0].ToString());
                        Session["uid"] = b_obj.user_typeid;
                        b_obj.staff_date_of_birth = txtdateofbirth.Text;
                        if (rbtmale.Checked == true)
                        {
                            b_obj.staff_gender = rbtmale.Text;
                        }
                        else
                        {
                            b_obj.staff_gender = rbtfemale.Text;
                        }
                        b_obj.staff_persal_no = Convert.ToInt32(txtpersalno.Text);
                        b_obj.staff_tax_no = Convert.ToInt32(txttax.Text);
                        b_obj.staff_date_of_join = txtdateofjoining.Text;
                        if (rbtsingle.Checked == true)
                        {
                            b_obj.staff_marital_status = rbtsingle.Text;
                        }
                        if (rbtmarried.Checked == true)
                        {
                            b_obj.staff_marital_status = rbtmarried.Text;
```

```
                        }
                        if (rbtseperated.Checked == true)
                        {
                            b_obj.staff_marital_status = rbtseperated.Text;
                        }
                        if (rbtdivorced.Checked == true)
                        {
                            b_obj.staff_marital_status = rbtdivorced.Text;
                        }

                        b_obj.staff_depandant_no = Convert.ToInt32(txtdependno.Text);
                        b_obj.staff_home_no = Convert.ToInt64(txthmno.Text);
                        b_obj.staff_cell_no = Convert.ToInt64(txtcellnumb.Text);
                        b_obj.staff_mail_id = txtemail.Text;
                        b_obj.staff_box_no = Convert.ToInt32(txtboxnum.Text);
                        b_obj.staff_province = Convert.ToInt32(ddlprovince.SelectedValue);
                        b_obj.staff_district = Convert.ToInt32(ddldistrict.SelectedValue);
                        b_obj.staff_town = Convert.ToInt32(ddltown.SelectedValue);
                        b_obj.staff_pin_no = Convert.ToInt32(txtpinno.Text);
                        b_obj.staff_regid = admissionno;

                        object staffid = new object();
                        staffid = d_obj.insert_staff_basic_details(b_obj);
                        Session["staffid"] = staffid;
                        Panel1.Visible = true;
                        txtusername.Text = admissionno;
                        ScriptManager.RegisterStartupScript(Page, this.GetType(), "write",
"alert('StaffDetails Entered Successfully');", true);
                    }
                    else
                    {
                        ScriptManager.RegisterStartupScript(Page, this.GetType(), "write", "alert('Please
select town');", true);
                    }
                }
                else
                {
                    ScriptManager.RegisterStartupScript(Page, this.GetType(), "write", "alert('Please
select district');", true);
                }
            }
            else
            {
                ScriptManager.RegisterStartupScript(Page, this.GetType(), "write", "alert('Please
select province');", true);
            }
```

```
        }
        else
        {
            ScriptManager.RegisterStartupScript(Page, this.GetType(), "write", "alert('Please select
designation');", true);
        }

    }
    protected void rbtseperated_CheckedChanged(object sender, EventArgs e)
    {

    }
    protected void ddlprovince_SelectedIndexChanged(object sender, EventArgs e)
    {
        if (ddlprovince.SelectedIndex > 0)
        {
            DataTable dt = new DataTable();
            b_obj.province_id = Convert.ToInt16(ddlprovince.SelectedValue);
            dt = d_obj.fetch_district(b_obj);
            ddldistrict.DataSource = dt;
            ddldistrict.DataValueField = "district_id";
            ddldistrict.DataTextField = "district_name";
            ddldistrict.DataBind();
            ddldistrict.Items.Insert(0, "-Select-");
        }

    }
    protected void ddldistrict_SelectedIndexChanged(object sender, EventArgs e)
    {
        if (ddldistrict.SelectedIndex > 0)
        {
            DataTable dt = new DataTable();
            b_obj.district_id = Convert.ToInt32(ddldistrict.SelectedValue);
            dt = d_obj.fetch_town(b_obj);
            ddltown.DataSource = dt;
            ddltown.DataValueField = "town_id";
            ddltown.DataTextField = "town_name";
            ddltown.DataBind();
            ddltown.Items.Insert(0, "-Select-");
        }
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        b_obj.userid = Convert.ToInt32(Session["staffid"]);
        b_obj.user_typeid = Convert.ToInt32(Session["uid"]);
        b_obj.user_username = txtusername.Text;
        b_obj.user_password = txtpwd.Text;
        d_obj.insert_userlogindetails(b_obj);
        ScriptManager.RegisterStartupScript(Page, this.GetType(), "write", "alert('Username and
password has been sent');", true);
    }
    protected void ddldesignation_SelectedIndexChanged(object sender, EventArgs e)
    {

    }
}
```

Adding a student

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
using System.Collections ;

public partial class StudentBasicDetails : System.Web.UI.Page
{
    int admission;
    string admissionno;
    ArrayList reg = new ArrayList();
    Datalayer d_obj = new Datalayer();
    Businesslayer b_obj = new Businesslayer();
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {

            fetchprovince();
            //fetchusertype();
        }

    }
    public void fetchdynamic()
    {
        DataTable dt = new DataTable();
        dt.Columns.Add("StudentName");
        dt.Columns.Add("Surname");

    }
    public void fetchprovince()
    {
        DataTable dt = new DataTable();
        b_obj.School_id = Convert.ToInt16(Session["s_id"]);
        dt = d_obj.fetch_province(b_obj);
        ddlprovince.DataSource = dt;
        ddlprovince.DataValueField = "province_id";
        ddlprovince.DataTextField = "province_name";
        ddlprovince.DataBind();
        ddlprovince.Items.Insert(0, "-Select-");


    }

    protected void sumbitstuddetail_Click(object sender, EventArgs e)
    {
        b_obj.RegCode = Convert.ToString(Session["regcode"]);
        string regcod = b_obj.RegCode;
        string  num = "0";
        DateTime dtnow = System.DateTime.Now ;
        string dtmonth = dtnow.ToString("MM");
        string year = Convert.ToString(dtnow.Year);
        DataTable dt1 = new DataTable();
        b_obj.School_id = Convert.ToInt16(Session["s_id"]);
        dt1 = d_obj.fetch_schooladmissionnumber(b_obj);
        if (dt1.Rows.Count == 0)
        {
            admission  = Convert.ToInt32 ( year + dtmonth + num);
            admissionno = Convert.ToString(regcod + admission);

        }
        else
        {
            string rg = dt1.Rows[0][0].ToString();
            int reglength=Convert.ToInt32 (rg.Length )-10;
            string yr = rg.Substring(4, 4);
            string rgs = rg.Substring(8, 2);
            string Nos = rg.Substring(10, reglength);
            if (yr == year)
            {
                if (rgs == dtmonth)
                {

                    Nos = (Convert.ToInt32(Nos) + 1).ToString();
                    admission = Convert.ToInt32(year + dtmonth + Nos);
                    admissionno = Convert.ToString(regcod + admission);
                }
            }
            else
            {
                admission = Convert.ToInt32(year + dtmonth + num);
                admissionno = Convert.ToString(regcod + admission);
            }
        }
```

```
        DateTime dt = Convert.ToDateTime(txt_dateofenrollment.Text);

        byte[] img = fileupload1.FileBytes;
        Session["imag"] = img;
        b_obj.School_id = Convert.ToInt16(Session["s_id"]);
        b_obj.student_name = txtstudentname.Text;
        b_obj.student_lastname = txt_surname.Text;
        b_obj.student_dob = txtdob.Text;

        if (rbtmale.Checked == true)
        {
            b_obj.student_gender = rbtmale.Text;
        }
        else
        {
            b_obj.student_gender = rbtfemale.Text;
        }
        b_obj.student_boxno = Convert.ToInt32(txt_box.Text);
        b_obj.student_province = Convert.ToInt32(ddlprovince.SelectedValue);
        b_obj.student_district = Convert.ToInt32(ddldistrict.SelectedValue);
        b_obj.student_town = Convert.ToInt32(ddltown.SelectedValue);
        b_obj.student_pinno = Convert.ToInt32(txt_pinno.Text);
        b_obj.student_residenceno = Convert.ToInt32(txtresidence.Text);
        b_obj.student_dateofenrollment = txt_dateofenrollment.Text;
        b_obj.student_image = img;
        b_obj.school_regid = admissionno ;
        if (rbtorphanyes.Checked == true)
        {
            b_obj.student_iforphan = rbtorphanyes.Text;
        }
        else
        {
            b_obj.student_iforphan = rbtorphano.Text;
        }
        b_obj.user_typeid = Convert.ToInt32 (3);
        Session["usertypeid"] = b_obj.user_typeid;
        object studid = new object();
        studid =d_obj.insert_studentregistrationdetails(b_obj);
        Session["studentid"] = studid                                        ;
        ScriptManager.RegisterStartupScript(Page, this.GetType(), "write", "alert('Student
    Registered Successfully');", true);
        Panel1.Visible = true;
        txtusername.Text  = admissionno;
    }
    protected void DropDownList3_SelectedIndexChanged(object sender, EventArgs e)
    {

    }
    protected void ddlprovince_SelectedIndexChanged(object sender, EventArgs e)
    {
        DataTable dt = new DataTable();
        b_obj.province_id = Convert.ToInt16(ddlprovince.SelectedValue);
        dt = d_obj.fetch_district(b_obj);
        ddldistrict.DataSource = dt;
        ddldistrict.DataValueField = "district_id";
        ddldistrict.DataTextField = "district_name";
        ddldistrict.DataBind();
        ddldistrict.Items.Insert(0, "-Select-");
    }
    protected void ddldistrict_SelectedIndexChanged(object sender, EventArgs e)
    {
        DataTable dt = new DataTable();
        b_obj.district_id = Convert.ToInt32(ddldistrict.SelectedValue);
        dt = d_obj.fetch_town(b_obj);
        ddltown.DataSource = dt;
        ddltown.DataValueField = "town_id";
        ddltown.DataTextField = "town_name";
        ddltown.DataBind();
        ddltown.Items.Insert(0, "-Select-");
    }
    protected void btnsend_Click(object sender, EventArgs e)
    {
        b_obj.userid = Convert.ToInt32 ( Session["studentid"]);
        b_obj.user_typeid = Convert.ToInt32(Session["usertypeid"]);
        b_obj.user_username = txtusername.Text;
        b_obj.user_password = txtpassword.Text;
        d_obj.insert_userlogindetails(b_obj );
        ScriptManager.RegisterStartupScript(Page, this.GetType(), "write", "alert('Username and
password has been sent');", true);
    }
}
```

Viewing marks

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;


public partial class StudentMarkdetails : System.Web.UI.Page
{
    Datalayer d_obj = new Datalayer();
    Businesslayer b_obj = new Businesslayer();
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            fetchclass();
            fetch_dynamic();
        }
    }
    public void fetch_dynamic()
    {

        DataTable dt = new DataTable();
        dt.Columns.Add("StudentId");
        dt.Columns.Add("StudentName");
        dt.Columns.Add("DivisionId");
        dt.Columns.Add("Division");
        dt.Columns.Add("SubjectId");
        dt.Columns.Add("Subject");
        dt.Columns.Add("Mark");
        Session["StudentMarkDetails"] = dt;

    }
    public void fetchclass()
    {
        DataTable dt = new DataTable();
        b_obj.School_id = Convert.ToInt16(Session["schoolid"]);
        dt = d_obj.fetch_class(b_obj);
        ddlclass.DataSource = dt;
        ddlclass.DataValueField = "class_creation_id";
        ddlclass.DataTextField = "class";
        ddlclass.DataBind();
        ddlclass.Items.Insert(0, "-Select-");

    }
    protected void ddlclass_SelectedIndexChanged(object sender, EventArgs e)
    {
        DataTable dt = new DataTable();
        b_obj.class_creation_id = Convert.ToInt16(ddlclass.SelectedValue);
        dt = d_obj.fetch_Divisionfromclass(b_obj);
        ddldivision.DataSource = dt;
        ddldivision.DataValueField = "student_div_id";
        ddldivision.DataTextField = "div";
        ddldivision.DataBind();
        ddldivision.Items.Insert(0, "-Select-");
    }
    protected void ddldivision_SelectedIndexChanged(object sender, EventArgs e)
    {
        DataTable dt = new DataTable();
        b_obj.studentdivision_id = Convert.ToInt16(ddldivision.SelectedValue);
        dt = d_obj.fetch_StudentnameFromclass(b_obj);
        ddlstudent.DataSource = dt;
        ddlstudent.DataValueField = "student_id";
        ddlstudent.DataTextField = "stud_name";
        ddlstudent.DataBind();
        ddlstudent.Items.Insert(0, "-Select-");
        subject();
    }
    public void subject()
    {
        DataTable dt1 = new DataTable();
        b_obj.studentdivision_id = Convert.ToInt32(ddldivision.SelectedValue);
        dt1 = d_obj.fetch_division_correspond_subject(b_obj);
        ddlsubject.DataSource = dt1;
        ddlsubject.DataValueField = "subject_creation_id";
        ddlsubject.DataTextField = "subject";
        ddlsubject.DataBind();
        ddlsubject.Items.Insert(0, "-Select-");
    }
    protected void ddlstudent_SelectedIndexChanged(object sender, EventArgs e)
    {

    }
    protected void ddlsubject_SelectedIndexChanged(object sender, EventArgs e)
    {
    }
    protected void btnaddtolist_Click(object sender, EventArgs e)
    {
       DataTable dt = (DataTable)Session["StudentMarkDetails"];
       DataRow dr = dt.NewRow();
       dr[0] = Convert.ToInt16(ddlstudent.SelectedValue);
       dr[1] = ddlstudent.SelectedItem.Text;
       dr[2] = ddldivision.SelectedValue;
       dr[3] = ddldivision.SelectedItem.Text;
       dr[4] = ddlsubject.SelectedValue;
       dr[5] = ddlsubject.SelectedItem.Text;
       dt.Rows.Add(dr);
       Session["StudentMarkDetails"] = dt;
       GridView1.DataSource = dt;
       GridView1.DataBind();
    }
    protected void GridView1_RowDeleting(object sender, GridViewDeleteEventArgs e)
    {
       DataTable dt = new DataTable();
       dt = (DataTable)Session["StudentMarkDetails"];
       dt.Rows[e.RowIndex].Delete();
       Session["StudentMarkDetails"] = dt;
       GridView1.DataSource = dt;
       GridView1.DataBind();
    }
    protected void btnsave_Click(object sender, EventArgs e)
    {
       DataTable dt = new DataTable();
       dt = (DataTable)Session["StudentMarkDetails"];
       for (int i = 0; i < dt.Rows.Count; i++)
       {
          TextBox tb1 = new TextBox();
          tb1 = (TextBox)(GridView1.Rows[i].Cells[6].FindControl("txt_stu_mark"));

          b_obj.student_id = Convert.ToInt16(dt.Rows[i][0].ToString());
          // b_obj.student_name = dt.Rows[i][1].ToString();
          b_obj.studentdivision_id = Convert.ToInt32(dt.Rows[i][2].ToString());
          b_obj.subject_creation_id = Convert.ToInt32(dt.Rows[i][4].ToString());
          b_obj.student_mark = Convert.ToDecimal(tb1.Text);
          d_obj.insert_student_mark_details(b_obj);

       }
    }
    protected void LinkButton10_Click(object sender, EventArgs e)
    {
       for (int i = 0; i < GridView1.Rows.Count; i++)
```

```
        {
            DataTable dt = new DataTable();
            dt = (DataTable)Session["StudentMarkDetails"];
                TextBox tb1 = new TextBox();
                tb1 = (TextBox)(GridView1.Rows[i].Cells[6].FindControl("txt_stu_mark"));
                 b_obj.student_id = Convert.ToInt16(dt.Rows[i][0].ToString());
                // b_obj.student_name = dt.Rows[i][1].ToString();
                b_obj.studentdivision_id = Convert.ToInt32(dt.Rows[i][2].ToString());
                b_obj.subject_creation_id = Convert.ToInt32(dt.Rows[i][4].ToString());
                b_obj.staff_details_id = Convert.ToInt32(Session["StaffId"]);
                b_obj.student_mark = Convert.ToDecimal(tb1.Text);
                d_obj.insert_student_mark_details(b_obj);
                GridView1.Visible = false;
         }
           }
    protected void GridView1_SelectedIndexChanged(object sender, EventArgs e)
    {

    }
}
```

APPENDIX 3 – QUESTIONNAIRE FOR USABILITY TESTING

Usability testing of the staff module

a) How much time does it take the system to display a student's personal details? Please rate your answer on a scale of 1-5, where 1 represents extremely slow and 5 exceptionally fast times.

b) How much time does it take the system to display a student's ailment details? Please rate your answer on a scale of 1-5.

c) How much time does it take the system to register a disciplinary record? Please rate your answer on a scale of 1-5.

d) How much time does it take the system to process information entered under a forum? Please rate your answer on a scale of 1-5.

e) What modification would you suggest in future for this module?

Usability testing of the staff module

a) How much time does it take the system to process notification information entered? Please rate your answer on a scale of 1-5.

b) How much time does it take the system to create a topic under a forum? Please rate your answer on a scale of 1-5.

c) How much time does it take the system to display disciplinary records that are yet to be approved? Please rate your answer on a scale of 1-5.

d) How much time does it take the system to display student details? Please rate your answer on a scale of 1-5.

e) How much time does it take the system to display staff details? Please rate your answer on a scale of 1-5.
f) What modification would you suggest in future for this module?

Usability testing of the student module

a) How much time does it take the system to view a notification? Please rate your answer on a scale of 1-5.
b) How much time does it take the system to display the exam results?
c) What modification would you suggest in future for this module?

Usability testing of the site admin module

a) How much time does it take the system to register a school?
b) What modification would you suggest in future for this module?

APPENDIX 4 QUESTIONAIRE'S GIVEN TO MEMBERS OF THE SCHOOL

Principal

a) What kinds of softwares are presently used for managing the activities of a school?

b) What are the functions of the SAMS software?

c) What are some of the records SAMS software maintains?

d) Who enters this information into the database?

e) How helpful do you find SAMS to be?

f) What are some of the drawback of this SAMS Software?

g) How often are staff meetings held?

h) Do you feel that all teachers contribute to these meetings?

i) As a principal what are some of your resbonsibilities?

Educators

a) What are the challenges that educators face?

b) How do you collect information about a student?

c) Are you able to meet the principal and other educators often and able to address some of the difficulties you are faced with?

d) If a software such as SAMS were available to you, would it aid you to understand your pupils better?