

|  |
| --- |
| PAT Phase 1  Table of Contents Jerome PR 12B Information Technology |

|  |  |
| --- | --- |
|  |  |
| Scenario | Pg2 |
| User Requirements | Pg3 |
| Navigation/Flow diagram | Pg4 |
| Data Structures | Pg5 |
| Database Design | Pg5-6 |
| Database Table Relationships | Pg6 |
| Textiles & Arrays | Pg6 |
| GUI Design | Pg7 |
| IPO | Pg8 |
| Data input | Pg8-9 |
| Data Processing | Pg9-10 |
| Data output | Pg10-11 |

Page 1

They have hired us to create a new and intuitive system. To do this we have thought up the following plan. We have decided to keep the overall system simple, so we went with only having two user levels, which would consist of an admin level and user level. The data will be stored in a database and the admin will act as the main database table that would branch to form relationships with other tables such as the parent and learner tables. This system is only for registration so having that in mind only Parents (Admins as well btw) can access their children/wards profile to edit any info, however this can only be done after the registration application has been accepted. The parent will receive an email if their child(ren)/ward(s) have been accepted or not, and any other necessary information.

Wart Hog’s high school, for the past 20 years, have been receiving and sending out registration applications as well as other notices by printing out these documents and physically handing them out to the necessary parties. However, over the past two years, ever since the inception of the corona virus, otherwise known as COVID-19, the school has adopted a more digital approach to their methods of registration and communication in order to prevent as well as reduce the amount of physical contact to help lower the risk of transmission of the virus. Wart Hog’s in this two-year period has managed to fully digitalize their communication systems, that being said, their registration and application system is still rather primitive, primitive in the sense that the schools administrative staff were making use of paper based methods of storing and accepting registration application ,this was done by the use of faxing documents, when both sending and receiving documentation , thus they have made the informative decision to make the investment and have a proper system installed/implemented.

Scenario

Problem Statement/Solution

Page 2

User Requirements

|  |  |  |
| --- | --- | --- |
| User Type | Data Value | Access And Limitations |
| Admin | 1 | * Login as admin * Admin privileges:   -Accept/Decline Student applications  -Access to school Database.  - Ability to edit/remove Records from  School database.   * Ability to review new/updates by students or   Staff before uploading to news wire.   * Able to upload new or updates to newswire. * Can send messages or advice to students or   Prospective students via the messenger and view  Replies in the inbox. |
| User | 0 | * Basic access privileges * Can create learner profile * View and edit personal profile * View Classroom Layout and seat location * View academic progress and view advice or   Tips given by teachers in the inbox.   * Send messages to other students via the   Messenger. |

# Extra Info:

* Data Value is the value that is present in both a username and password to differentiate between User accounts and admin accounts. This is so that the system will Know what form to take the respective user too once they have entered their username and password.

Page 3

Navigation

Home Page

Help

Login/Registration

Student Dash

Administrator Dash

Student Application

Help

Page 4

Page 5

Data Structures

Database Design

|  |  |  |
| --- | --- | --- |
| TblParent | Field | Field Size |
| (PK)ParentID | Text | 13 |
| ParentName | Text | 30 |
| ParentSurname | Text | 30 |
| Address | Text | 40 |
| Marital Status | Text | 15 |
| ParentNum | Text | 10 |
| Occupation | Text | 30 |
| Email | Text | 50 |

|  |  |  |
| --- | --- | --- |
| TblLearner | Field | Field Size |
| (PK)LearnerID | Text | 13 |
| LearnerName | Text | 30 |
| LearnerSurname | Text | 30 |
| (FK)ParentID | Text | 13 |
| RegFee | Boolean | - |
| DateOfBirth | DateTime | ShortDate |
| DateOfRegistration | DateTime | ShortDate |
| Learner\_No | Integer | 4 |
| Learner\_Password | String | 30 |
| Fees | Currency | 5 |

|  |  |  |
| --- | --- | --- |
| tblGrade | Field | Field Size |
| (PK)Grade | Text | 3 |
| ClassSize | Integer | - |
| FormTeacher | Text | 30 |
| (FK)SubjectComboNo | Text | 4 |

|  |  |  |
| --- | --- | --- |
| TblRegistered\_Learner | Field | Field Size |
| (FK)LearnerID | Text | 13 |
| (PK)Grade | Text | 3 |
| (PK)AdmNo | Text | 5 |
| (FK)ParentID | Text | 13 |
| SubjectComboNo | Text | 4 |

|  |  |  |
| --- | --- | --- |
| TblSubject | Field | Field Size |
| (PK) SubjectComboNo | Text | 4 |
| Subject1 | Text | 50 |
| Subject2 | Text | 50 |
| Subject3 | Text | 50 |
| Subject4 | Text | 50 |
| Subject5 | Text | 50 |
| Subject6 | Text | 50 |
| Subject7 | Text | 50 |

|  |  |  |
| --- | --- | --- |
| TblProspective\_Learner | Field | Field Size |
| (PK) Learner\_ID | Text | 13 |
| LearnerName | Text | 30 |
| LearnerSurname | Text | 30 |
| Parent\_ID | Text | 13 |
| RegFee | Currency | 5 |
| DOB | DateTime | - |
| DOR | DateTime | - |
| Temp\_pass | Text | 6 |
| Grade | string | 3 |
| SubjectComboNo | string | 4 |
| Email | string | 30 |

Database Table Relationships

Graphical user interface

Description automatically generated

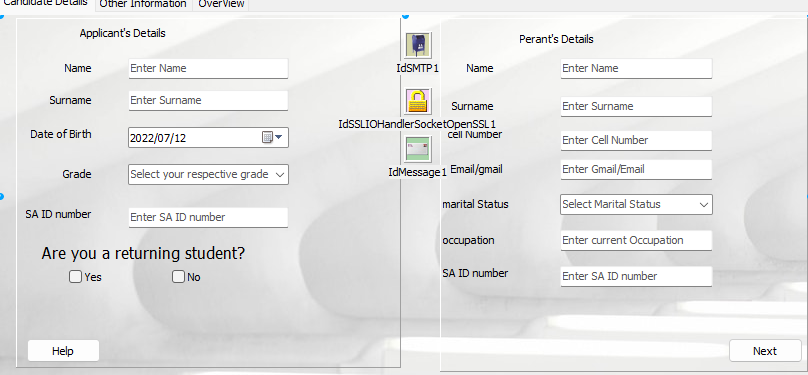
TextFiles and Array’s

* Textfiles will be used to store Help info as well as admin profiles.
* An array will be used to display results of learners and other performance states alongside with graphs.

Page 6

Graphical User

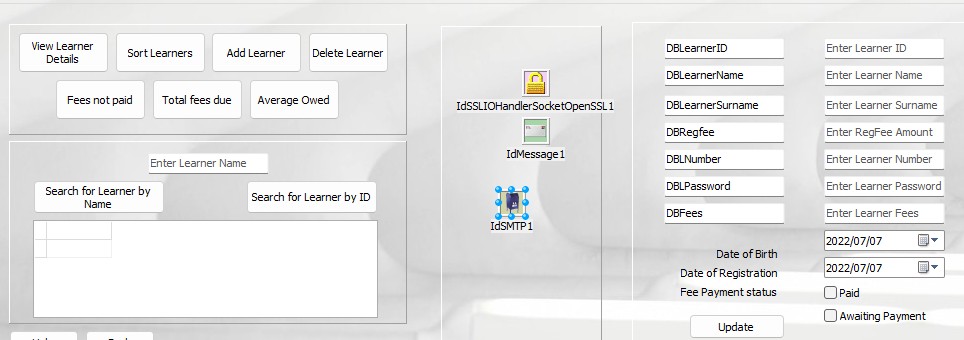
Interface



Graphical user interface

Description automatically generated

Page 7



IPO

Data Input

Dat

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Input | Source | DataType | Component | Validation | Error Message |
| Applicant Name | Keyboard | String | Tedit/DBedit | Check if value entered only consists of string character’s and is less than or equal to 30 characters. | The name you have entered is invalid , Please re-enter a valid name |
| Surname | Keyboard | String | Tedit/DBedit | Check if value entered only consists of string character’s and is less than or equal to 30 characters. | The surname you have entered is invalid , Please re-enter a valid surname. |
| Age | Keyboard | integer | Spinedit/DBedit | Check if value entered is only an integer value and between the set age limits. | The age you have entered is invalid or the age you have entered is not eligible for the respective grade. |
| Previous School | mouse | Boolean and string | Check box and edit | Check if applicant select whether or not they are a returning student. | Please indicate if you are a returning student. |
| DOB | keyboard | DateTime | TDateTimePicker | Check if the date chose will equal and age older than 16. | Please choose an appropriate date. |
| ID number | Keyboard | string | Tedit/DBedit | Check if field is filled and if entered value contains 13 charaters. | The id number you have entered is invalid. |

Data Processing

What Processing will be Done:

* Searching Data Bases for specific user data
* Editing the data bases
* Password creation
* Data collection and manipulation
* Formatting and displaying data
* Writing/reading data with textfiles

How Processes will be done:

1. Reading Admin Data to a textfile:

* Username > edit
* Password > edit
* If file (Admin.txt) exists
* Assign file to tfile
* Counter > 0
* Reset tfile
* While not eof tfile

Beginning of loop

Increment counter

Readln (sline,tfile)

arrUsersname>sline

arrPassword>sline

end of loop

close tfile

1. Searching Learner by Learner Name:

* Input Learner Number
* Input Learner Password
* Open tblLearners
* Go to first record
* While not eof tblLearners

Beginning of loop

If tblLearners [‘Learner\_No’] = Learner Number AND tblLearners[‘Learner\_Password’] = Learner Password

Student Dash > show

Login Page> hide

Else Go to next Record

Else Show Error Message

End Of loop

1. Adding New Learner:

* Input Learner ID
* Input Learner Name
* Input Learner Surname
* Input Learner Number
* Input Parent ID
* Input Regfee
* Input Fees
* Input FeesPaid
* Input Password
* Input DOB
* Input DOR
* Open tblLearners
* Go to Last Record
* Set tblLearners into Insert mode
* tblLearners[‘Leaner\_ID’]>Learner ID
* tblLearners[‘LearnerName’]>Learner Name
* tblLearners[‘LearnerSurname’]>Learner Surname
* tblLearners[‘Parent\_ID’]>Parent ID
* tblLearners[‘Regfee’]> Regfee
* tblLearners[‘Fees’]>Fees
* tblLearners[‘FeesPaid’]>FeesPaid
* tblLearners[‘Password’]>Password
* tblLearners[‘DOB’]>DOB
* tblLearners[‘DOR’]>DOR

Post record to tblLearners

Close tblLearners

1. Calculating Fees Owing:

* With WortHogs data module
* Close SQL
* Clear SQL
* Add SQL (‘Select sum(Fees) +sum(RegFee) as [Total fees due] from tblLearner’)
* Open SQL

Page 9

Data Output

|  |  |  |
| --- | --- | --- |
| Output | Data Type | Component |
| Applicant Name | String | DBgrid |
| Surname | String | DBgrid |
| Age | integer | DBgrid |
| Grade | String | DBgrid |
| DOB | DateTime | DBgrid |
| ID number | string | DBgrid |
| Date of registration | DateTime | DBgrid |
| Reg Fees | True/False | DBgrid |
| Parent ID number | string | DBgrid |
| Previous School | String | DBgrid |