

Tool Support for the Prevention of Family Violence System

A Manuscript

Submitted to

the Department of Computer Science

and the Faculty of the

University of Wisconsin-La Crosse

La Crosse, Wisconsin

by

Jami Hundt

in Partial Fulfillment of the

Requirements for the Degree of

Master of Software Engineering

November, 2007

Tool Support for the Prevention of Family Violence System

By Jami Hundt

We recommend acceptance of this manuscript in partial fulfillment of this candidate's requirements for the degree of Master of Software Engineering in Computer Science. The candidate has completed the oral examination requirement of the capstone project for the degree.

Dr. Kasi Periyasamy
Examination Committee Chairperson

Date

Dr. Thomas Gendreau
Examination Committee Member

Date

Dr. David Riley
Examination Committee Member

Date

ABSTRACT

HUNDT, JAMI, B., “Tool Support for the Prevention of Family Violence System”, Master of Software Engineering, November 2007, (Dr. Kasi Periyasamy, Dr. Thomas Gendreau).

Family violence is very prevalent in the State of Wisconsin. Family violence is classified into two types of violence: domestic violence or sexual assault. The Prevention of Family Violence System (PFV) was first developed in 2000 by the PKSoftware Company. The system is currently used by ten different agencies throughout the State of Wisconsin to record family violence cases. After long-term use of this application, it was determined that the application needed a renovation. The purpose of this project is to reengineer the PFV system. The new system contains all current functionality as well as new enhancements as specified by the sponsor. In addition, the data for the system was converted from Microsoft Access databases to Microsoft SQL databases and normalized.

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to my project advisors Dr. Kasi Periyasamy and Dr. Thomas Gendreau for their valuable guidance. I thank them for their time and expertise. I also would like to thank the project sponsor, PKSoftware and John Parshall, who initiated this project and provided the support for this project. I need to express my special thanks to my co-workers and fellow IT personnel for their valuable input and advice. I would also like to thank to the Computer Science Department, Murphy Library and the University of Wisconsin-La Crosse for providing me with the computing environment for my project. In addition, I wish to thank my husband, our parents and my daughters for their patience and encouragement during the tenure of this project. I truly appreciate the sacrifices they made to enable me to complete this degree. Finally, I want to thank all my friends and family for their support and encouragement during the duration of this project and the MSE program. I couldn't have done it without you.

TABLE OF CONTENTS

ABSTRACT.....	iii
ACKNOWLEDGEMENTS.....	iv
TABLE OF CONTENTS.....	v
LIST OF TABLES.....	vi
LIST OF FIGURES.....	vii
GLOSSARY.....	1
1. Background Information.....	1
2. A Brief Introduction to Software Life Cycle Models.....	3
3. Development of the Prevention of Family Violence System.....	5
3.1. Reverse Engineering of the Existing Product.....	5
3.2. Rapid Prototyping for PFV 16.0.....	7
3.3. Modifying the Abstraction – Requirements, Analysis and Specification.....	8
3.4. Design.....	15
4. The PFV 16.0 Application.....	16
4.1 High Level Architectural Design.....	16
4.2 Detailed Architecture of PFV 16.0.....	17
4.3 Database Design.....	20
4.4 User Interface Design.....	25
4.5 Security Design.....	26
4.6 Deploying PFV 16.0.....	28
5. Limitations.....	29
6. Continuing Work.....	30
7. Conclusion.....	31
BIBLIOGRAPHY.....	32
APPENDIX A: Selected PFV 16.0 Screen Shots.....	33

LIST OF TABLES

Table 1. Core Functionalities	6
Table 2. Maintenance Functionalities	6
Table 3. Functionalities Provided By Third Party Software.....	6
Table 4. Additional Enhancements/Requirements for PFV 16.0.....	9
Table 5. Enterprise Considerations of Microsoft Access vs. Microsoft SQL Server .	13
Table 6. Physical Characteristics of Microsoft Access vs. Microsoft SQL Server	14
Table 7. Data Type Differences Between Microsoft Access vs. Microsoft SQL Server	22

LIST OF FIGURES

Figure 1. High-Level Architecture of PFV 16.0.....	16
Figure 2. Detailed Architecture of PFV 16.0 (UML Class Diagram).....	18
Figure 3. User Classifications (Security Use Case Model) of PFV 16.0.....	19
Figure 4. Database Diagram of PFV 16.0 Tables	24
Figure 5. Database Diagram of PFV 16.0 Maintenance Tables	25
Figure 6. Login Screen.....	33
Figure 7. Main Menu Screen	34
Figure 8. File Search Screen	35
Figure 9. File Screen – Master Client	36
Figure 10. File Screen – Secondary Client	37
Figure 11. File Screen – Common Information	38
Figure 12. File Screen – Contact Information	39
Figure 13. File Screen – Sexual Assault Information	40
Figure 14. File Screen – DV Intake Information	41
Figure 15. File Screen – Milestone Information.....	42
Figure 16. File Screen – TANF Information	43
Figure 17. File Screen – Images/Docs Information	44
Figure 18. Information and Referral Screen	45
Figure 19. Education Presentation Screen	46
Figure 20. Reporting Screen	47
Figure 21. Agency Information Screen.....	48
Figure 22. User Account Information Screen.....	49
Figure 23. Database Maintenance Screen – Description Maintenance.....	50
Figure 24. Database Maintenance Screen – Service Maintenance	51
Figure 25. Delete Client Screen	52
Figure 26. Site Maintenance Screen	53

GLOSSARY

Active Directory

Active Directory is an implementation of LDAP (Lightweight Directory Access Protocol) directory services by Microsoft for use primarily in Windows environments. [2]

ADO.NET

A set of computer software components that can be used by programmers to access data and data services. It is a part of the base class library that is included with the Microsoft .NET Framework. It is commonly used by programmers to access and modify data stored in relational database systems, though it can also be used to access data in non-relational sources. [2]

Data Integrity

A process that ensures that the particular data is "whole" or complete. It refers to the condition in which data is identically maintained during any operation, such as transfer, storage, and retrieval. It also includes the preservation of data for their intended use. [2]

DV

Domestic Violence is one type of family violence. It refers to the inflicting of physical injury by one family or household member on another; it also includes a repeated or habitual pattern of such behavior [1].

Executable

An executable or executable file, in computer science, is a file whose contents are meant to be interpreted as a program by a computer. [2]

Fat Client

A desktop application. A computer (client) in client-server architecture networks which typically provides rich functionality independent of the central server [2].

I & R

Information and Referral.

Microsoft Access 2000

A relational database management system from Microsoft which combines the relational Microsoft Jet Database Engine with a graphical user interface. [2]

Microsoft Jet Database Engine

A database engine on which several Microsoft products were built. A database engine is the underlying component of a database, a collection of information stored on a computer in a systematic way. JET stands for Joint Engine Technology, sometimes being referred to as Microsoft JET Engine or simply Jet. [2]

Microsoft SQL Server 2000

A relational database management system (RDBMS) produced by Microsoft. Its primary query language is Transact-SQL, an implementation of the ANSI/ISO standard Structured Query Language (SQL) used by both Microsoft and Sybase. [2]

Normalization

Database normalization is a design technique by which relational database tables are structured in such a way as to make them less vulnerable to certain types of logical inconsistencies and anomalies. [2] The goal of normalization is to reduce redundancy.

PFV

Prevention of Family Violence

SA

Sexual Assault is one type of family violence. Illegal sexual contact that usually involves force upon a person without consent or is inflicted upon a person who is incapable of giving consent (as because of age or physical or mental incapacity) or who places the assailant (such as a doctor) in a position of trust or authority [1].

Stored Procedure

A program that is physically stored within a database that can be run to perform various queries or other database operations.

T-SQL

Transact - Structured Query Language is a primary language used to create, modify and query relational databases. Sometimes abbreviated T-SQL, Transact-SQL is Microsoft's and Sybase's proprietary extension to the SQL language. [2]

SRS

Software Requirements Specification, a document format supplied by the IEEE for specifying the requirements of a software system.

TANF

Temporary Assistance for Needy Families.

Thick Client

See 'Fat Client'.

Thin Client

A thin client (sometimes also called a lean client) is a client computer or client software in client-server architecture networks which depends primarily on the central server for processing activities, and mainly focuses on conveying input and output between the user and a remote server. [2]

UML Class Diagram

In the Unified Modeling Language (UML), a class diagram describes the static structure of a system by showing the system's classes, their attributes, and the relationships between the classes. [2]

Unit Testing

Functional and reliability testing in an Engineering environment. It produces tests for the behavior of components of a product to ensure their correct behavior prior to system integration. [8]

Use Case

A technique for capturing functional requirements of systems and systems-of-systems. Each use case provides one or more scenarios that convey how the system should interact with the users called actors to achieve a specific business goal or function. [2]

Visual Basic 6

An event driven programming language and associated development environment from Microsoft for its COM programming model. [2]

Visual Basic .NET

The latest technology from Microsoft for creating desktop applications based on the .NET framework. It is a framework for building desktop applications built on top of the Microsoft Common Language Runtime (CLR).

Visual Studio .NET

A Microsoft GUI development environment for building many kinds of software applications.

VLAN

Virtual LAN. A method of creating independent logical networks within a physical network. [2]

1. Background Information

Family violence is very prevalent in the state of Wisconsin. According to the state of Wisconsin, family violence can be of two types: domestic violence or sexual assault. The state is required to track all family violence cases and collect data on the specific incidents that occur. The data itself serves multiple purposes. The data is (1) used in court cases as evidence, (2) analyzed and used at the state level to determine future funding for each agency based on the quantity and severity of the cases, and (3) examined and referenced when molding existing programs and building future programs based on the clientele needs. The State of Wisconsin needed software to better organize the advocate's case management activities by reducing manual steps, improving productivity and creating a complete accounting of all the clients and services performed.

The State of Wisconsin decided to use the Prevention of Family Violence (PFV) system as the tool to electronically capture the data. The PFV system contains modules to record all the data necessary. The user is able to log a case. A case is either considered to be domestic violence or sexual assault. A case consists of a master client and zero or many secondary clients. Each client includes their own information in the system. In addition, information about each case may also include scanned images and documents. The system will also report on the information captured in both tabular and chart formats. The system also captures information that is not related to a case but is needed for state and agency documentation. There is also a system maintenance area of the application that allows each agency to customize and maintain its own application data.

The Prevention of Family Violence (PFV) system was first developed in 2000 by the PKSoftware Company. The current system is a desktop application utilized by ten different agencies throughout the State of Wisconsin. Each agency houses its own version of the system. Although all versions start out the same at installation, all agencies have the ability to customize the application through maintenance screens.

The current PFV system was written in Microsoft Visual Basic 6.0 with backend Microsoft Access 2000 databases. Through the system, the user can create up to 10 different sites. Each site contains its own database and set of tables. So, for each installation of the system, there are 10 databases, all containing the same database schema. The current system also utilizes Crystal Reports for reporting purposes.

After long-term use of this application, increased user expectations and changing technology, it was determined that the application needed an overhaul. This report describes the development process used to re-engineer the current PFV system. The sponsor of this project received and compiled a list of enhancements from the agencies currently using the application. Due to agency funding and the computer knowledge of the users, it was decided to keep the new version of PFV as a thick client application. The new PFV system will also contain some security enhancements to protect the data.

2. A Brief Introduction to Software Life Cycle Models

The software life cycle is a structure imposed on the development of a software product. There are several models for such processes, each describing approaches to a variety of tasks or activities that take place during the process [11].

A decades-long goal has been to find repeatable, predictable processes or methodologies that improve productivity and quality. Some try to systematize or formalize the seemingly unruly task of writing software. Others apply project management techniques to writing software. Without project management, software projects can easily be delivered late or over budget. With large numbers of software projects not meeting their expectations in terms of functionality, cost, or delivery schedule, effective project management is proving difficult [12]. There is no one software life cycle that is best suited for one project. More than one model may be used to compliment the other. Choosing the appropriate life cycle model for the project is often a very difficult task. There are many factors that may influence the choice of the life cycle model such as team size, project size, available resources, deadlines or milestones, team skills and experience, business policies, and application domain.

The best known and oldest life cycle model is the waterfall model. The waterfall model is a sequential software development model in which development is seen as flowing steadily downwards (like a waterfall) through the phases of requirements analysis, design, implementation, testing (validation), integration, and maintenance [11]. There are many variations of the waterfall model but they all contain the above core phases. An in-depth discussion of the waterfall model is beyond the scope of this report, but additional information can be found in many texts on Software Engineering practices.

A more recent and evolutionary life cycle model is the rapid prototyping model. In this model, once the requirement analysis is done and the design for a prototype is made, the development process gets started. Once the prototype is created, it is given

to the customer for evaluation. The customer tests the package and gives his/her feedback to the developer who refines the product according to the customer's exact expectation. After a finite number of iterations, the final software package is given to the customer. In this methodology, the software is evolved as a result of periodic shuttling of information between the customer and developer. This is the most popular development model in the contemporary IT industry [12].

In this project, the developer applied both the rapid prototyping model and the waterfall model. Initially, one rapid prototype model helped the developer identify the existing requirements of the project, provide screens shots to demonstrate the new look of the application and also discover additional requirements. After discovering all possible requirements, the developer used the waterfall model to develop the final product.

3. Development of the Prevention of Family Violence System

The PFV 16.0 project was developed as a reengineering project. The first phase of development involved reverse engineering of the original product PFV 15.0. This was done to retrieve the functionalities of the existing product that capture a majority of the requirements for PFV 16.0. In the second phase, requirements were analyzed and a detailed specification was written. A rapid prototype was developed to communicate with the sponsor and more quickly discover additional requirements as well as changes to existing requirements. The prototype also demonstrated the new look and navigation of the application. In the third phase, the product was implemented using the waterfall model.

3.1. *Reverse Engineering of the Existing Product*

In order to reverse engineer the PFV system, several activities were conducted to (1) understand the existing product from an end-user perspective, including the application domain and (2) understand it from an architectural and development perspective.

For this project, the sponsor served as the application domain resource and the source code expert. The first step was to observe a complete demonstration of the existing product and have a brief discussion of the application domain and the system itself from a new user's perspective. The product was then installed and tested from a new user's perspective to gain knowledge of existing functionalities. In addition, the source code and data sources of the product were received from the sponsor and studied. After the discussion with the sponsor and a detailed study of the existing system, the functionalities were identified and grouped into three major categories: (1) core functionalities, (2) maintenance functionalities, (3) functionalities provided by third party software. These functionalities are described in Table 1 through Table 3.

	Module Name	Functionalities
1	File Module	View, add, update, print and search files/cases and all corresponding information related to a file/case. There are 9 sub-modules that are included in the main file/case module.
2	Information and Referral Module	Add and/or print references given and all calls taken by the agency staff.
3	Education Module	Add, edit and print the training and courses taken by the agency staff.
4	Reports Module	View and/or print a report based on a date range specified by the user.
5	Delete Client Module	Delete a file/case. This delete will remove all corresponding information related to the file/case.
6	Age Update	Update the age of all master and secondary clients in the database.
7	Log In Module	Allows the user to successfully log into the system when supplying the correct credentials.
8	Log out module	Safely logs out the user.

Table 1. Core Functionalities

	Name of module	Purpose
1	Agency Module	Update the existing agency information.
2	User Maintenance Module	Add, update and delete the users of the system. This includes maintenance of all passwords and security access levels.
3	Database Maintenance Module	Add, update and delete the data that will appear in pre-specified fields within the system.
4	Site Database Module	Add, update and delete sites with the system.

Table 2. Maintenance Functionalities

	Name of module	Purpose
1	Spell Checker Module	Provides alternative spellings to misspelled or unrecognized words. Allows the user to update or delete a word based on the dictionary and grammar within the spell checker.

Table 3. Functionalities Provided By Third Party Software

During the reverse engineering process, a detailed review and analysis was done on the Microsoft Access database schema and structure. It was quickly realized that the tables in the Microsoft Access databases were not normalized or organized appropriately. An examination of the database was performed based on the following criteria:

- For every table/field, is the table/field used in the PFV 15.0 system?
 - If so,
 - Where is it used?
 - Does it have a relationship to any other table/field?
- Are the keys identified properly in each table?
- Are there foreign keys defined in the tables?
- Is there duplication among the table schemas?

The analysis revealed that there are tables and fields not being used in the PFV 15.0 system. There are no relationships defined in the database itself, but the application domain clearly defines relationships among the data. The keys, defined and used, seemed to be appropriate based on the application domain. There was duplication throughout many of the tables used for maintenance. Many of the issues found within the database can potentially cause performance and latency issues in the system. It should also be noted that for each install of the PFV 15.0 system, there are 10 databases – each with the same database schema. The duplication within the PFV 15.0 was one of the main concerns for the sponsor.

3.2. *Rapid Prototyping for PFV 16.0*

During the reverse engineering process, it became clear in working with the sponsor of the current project that prototyping a subset of functionality would be of great value to both the sponsor and the developer. It helped in refining requirements, and evaluating the feasibility of the product. Upon the sponsor's request, Visual Basic .NET was chosen as the programming language and tool for the project. The sponsor also specified Microsoft SQL Server 2000 as the database tool. VB.NET was

used to quickly develop prototype user interface screens. ADO.NET was used to connect to the SQL database and prototype a small portion of non-GUI functionalities. This prototype included user logins and simple record queries to establish communication between the VB.NET project files and the SQL Server.

The prototype implemented most of the functionalities in PFV 15.0 including some screen shots. It also included some additional enhancements that the sponsor had asked for. The sponsor's feedback was positive and the re-engineering process started.

3.3. *Modifying the Abstraction – Requirements, Analysis and Specification*

In this phase, the requirements for PFV 16.0 were identified and analyzed. A subset of the requirements was extracted from the functionalities of PFV 15.0. An additional subset of the requirements was provided by the sponsor based on user feedback. The additional enhancements/requirements requested by the sponsor are shown below in Table 4.

	Screen/Module	Enhancement
1	File Module, Service Sub-Module	Improve the interface for the deleting / changing Services on the Service Screen.
2	File Module, File Search Sub-Module	Color the main search list that identifies the ages of cases older then 2 years.
3	Report Module	Implement the export feature for Crystal.
4	File Module, Sexual Assault Sub-Module	On the first Sexual Assault screen, break apart relationship name to be first and last name.
5	File Module, Sexual Assault Sub-Module	On the first Sexual Assault screen, allow more space for the relationship name.
6	File Module, Education Sub-Module	Add Search Tool and interface for education section.
7	File Module	Add an Alternate ID to the database for identification purposes.
8	File Module, Intake Sub-Module	Allow the user to enter multiple intakes.
9	Database Maintenance Module	Improve the interface for the database maintenance screen.

10	Report Module	Change Reports to reflect the new Alt ID database changes.
11	File Module, Common Sub-Module	In the common area, made the 'Other Special Needs' drop down box table driven instead of hard-coded.
12	File Module, Contact Sub-Module	Allow 00 for minutes under the contact screens.
13	Security	Encrypt the password stored in the database.

Table 4. Additional Enhancements/Requirements for PFV 16.0

Through many discussions with the sponsor, it was very important to note that the users of the PFV system are agency workers with little or no computer experience. For this reason, it was requested by the sponsor to keep this system as a fat client (desktop) application. The sponsor wanted to enhance the application and use current technology but did not want to change the system so drastically that there could potentially be training issues with the users. Since the PFV 16.0 system will replace the PFV 15.0 system, the sponsor wanted a seamless transition to the users. A more detailed examination of the definition of a fat client application was done to help justify this decision. The following are some of the important advantages and disadvantages of a fat client application:

Advantages of fat client applications

- Rich screen functionality, high level of interactivity. Since the client is a program installed on the PC, it can be made to perform virtually any functionality desired. This allows screen choices that affect other parts of the screen or application, etc. The application appears more interactive and “live” to the user. It also appears to be faster, since the client’s machine is performing some of the processing and fewer trips are usually made to the server. [4]
- There is added security due to the fact that each user must have the client application installed. Meaning that people without the client installed cannot

access the system. Having a client application increases security to some level, as outsiders without the application installed on their PC will not be able to easily access server side information. [4]

- Client is more easily able to access printer and disk. Application based clients have more access to the client's PC utilities, allowing more elaborate print functionality, storage access, multimedia display, etc. [4]
- Fewer server requirements. A thick client server does not require as high a level of performance as a thin client server (since the thick clients themselves do much of the application processing). This allows the system administrators to use cheaper servers. [2]
- Offline working. Thick clients have advantages in that a constant connection to the central server is often not required. [2]
- Better multimedia performance. Thick clients have advantages in multimedia-rich applications that would be bandwidth intensive if fully served. For example, thick clients are well suited for video gaming. [2]
- More flexibility. On some operating systems software products are designed for personal computers that have their own local resources. Trying to run this software in a thin client environment can be difficult (such as Microsoft Windows). [2]

Disadvantages of fat client applications

- Updates, new features, and bug fixes require users to download and install client side software. Every user interface change requires client software redistribution to the PC. This is the biggest disadvantage of all thick clients. Even with the extreme care used in constructing installation scripts, the variation in users' machines is great enough to cause problems in some cases. These cases require help desk intervention and sometimes even a site visit to correct. Under the thick client model, new features, bug fixes, and other updates that require user interface changes cannot be implemented in

production until the new client user interface is built, distributed and installed among the user community. [4]

- Server must account for multiple versions of clients, increasing complexity. Versioning is an often overlooked thick client issue. Since users rarely change over to new client software applications all at once, and some stay with the old client forever, the server side must recognize and forever accept requests from old or obsolete client software. This dramatically increases the complexity in both design and test of future server architectures, since they must account for these past client formats. [4]
- Users cannot access the system from any location unless the client software is installed in that location. This can be a problem for users working from home or multiple work locations. It also means they cannot, on demand, get an answer to a spontaneous request, as they could if they could access server information from any browser enabled machine. [4]

It was also determined by the sponsor to migrate the Microsoft Access 2000 database to Microsoft SQL Server 2000. During the analysis phase of the project, a detailed study of the differences between Access and SQL Server was done. Table 5 provides the enterprise considerations for Access vs. SQL Server. Table 6 displays the physical differences and limitations of the two database options as found in the literature [6].

	Difference	Microsoft Access	Microsoft SQL Server
1	Reliability	With Access each client reads and writes directly to the raw data tables. If a client machine crashes while writing data this will usually cause the back-end database to also crash and become corrupt. The same thing will occur if the network fails, has a glitch or temporarily	With SQL Server the clients do not talk directly with the tables but with an intelligent data manager on the server. This in turn reads and writes data from and to the tables. If a client machine crashes, or the network hiccups, this will not affect the underlying

		<p>becomes overloaded. This problem becomes more apparent as the amount of data or the number of users increases. [5]</p>	<p>tables; instead the data manager realizes that the transaction has not been completed and does not commit the partially transmitted data to the database. The database therefore continues to run without problem.</p> <p>The client/server system also maintains an automatic 'transaction log'. If a backup has to be restored the transaction log can be run and should restore all completed transactions up to the time of the crash. [5]</p>
2	<p>Performance: For most databases the main performance bottleneck is data transmission over the network hence reducing this can give a really dramatic improvement in performance. [5]</p>	<p>With Access all tables involved in a form, report or a query are copied across the network from the server to the client's machine. The tables are then processed and filtered to generate the required recordset. For example if looking up details for one particular order from an orders table containing, say, 50,000 records then the whole table (all 50,000 records) is dragged over the network and then 49,999 of these records are thrown away (this is an oversimplification since indexing can be used to mitigate this to some extent). Contrast this with SQL Server where the filtering takes place on the server (if designed properly) and only 1 record is transmitted over the network. [5]</p>	<p>With SQL Server, the filtering takes place on the server (if designed properly) and only 1 record is transmitted over the network. This can affect performance in two ways. First, SQL Server is highly optimized and can usually perform the required filtering much more quickly than the client machine. Second, the amount of data sent across the network link is vastly reduced. [5]</p>

3	Scalability	A file server system such as Access is designed for small workgroups and is scalable to perhaps 10 concurrent clients. Above this level performance starts to degrade rapidly as more users are added. [5]	With the SQL Server client/server architecture many hundreds, or even thousands (with the appropriate infrastructure), of concurrent users can be supported without significant performance degradation. [5]
4	Security	Access databases (using the Microsoft Jet Database Engine) can be password protected and encrypted.	SQL Server provides user security which is used to manage and restrict data access to prevent accidental or malicious damage. [5] SQL Server offers 128-bit encryption and storage in a remote location from the user and application.
5	Support Environment	Access is very user friendly and easy to use. There is help through the application and virtually anyone could pick it up and start using it.	SQL Server is a (much) bigger and more complex beast than is Access. Although it is now easier to manage than in the past it is less suitable for a company with no IT support staff (in-house or outsourced) than is the simpler Access. [5]
6	Deployment Cost	Access run-time version is royalty free. [5] There is virtually no implementation cost when using Access.	The company needs to buy SQL Server licenses and the implementation cost will take longer and, therefore, be more costly.

Table 5. Enterprise Considerations of Microsoft Access vs. Microsoft SQL Server

	Feature	Microsoft Access	Microsoft SQL Server
1	Database size	2 GB plus linked tables size	1,048,516 TB
2	Objects in a database	32,768	2,147,483,647
3	User name length	20	128
4	Password length	14	128
5	Table name length	64	128
6	Column name length	64	128

7	Index name length	64	128
8	Number of concurrent users	255	limited by available memory
9	Columns per table	255	1024
10	Table size	1 GB	limited by available storage
11	Number of indexes in a table	32	250
12	Number of columns in an index	10	16
13	Bytes per row	2000	8060
14	Number of tables in a query	32	256
15	Columns per SELECT statement	255	4096
16	Nested subqueries	50	32
17	Number of enforced relationships	32	253
18	SQL statement size	approximately 64,000	65,536 * Network packet size (4 KB, by default)

Table 6. Physical Characteristics of Microsoft Access vs. Microsoft SQL Server

Although the cost and implementation factors of the SQL Server are a concern for the sponsor, the additional security and robustness that SQL Server offers outweighs that concern.

The decision to migrate the Microsoft Access database to SQL Server database caused a great deal of analysis to achieve the proper schema for the data and the application. There are many aspects of migration that need to be considered prior to starting the process. These considerations will identify any major problems or conflicts.

- Syntactical differences of query statements. This includes the supported clauses and operators, aggregate functions, conversion functions and date functions.
- Data types. Access contains different data types than SQL Server.

- Keys and indexes. Access and SQL Server both have their own way of key creation and indexing.

3.4. Design

Since the technologies of the application were decided by the sponsor, there was no analysis done on additional technologies. However, a brief study was done to verify that the specified technologies could accommodate the requirements for the project. The technologies chosen for this project are: VB.Net, SQL Server 2000, Crystal Reports 8.5 and Polar SpellChecker for .NET. It was found that all the technologies were able to accommodate the requirements as specified by the sponsor.

An architectural design was created using VB.NET. The developer then created a more detailed design after refining the architectural design through analysis. The detailed design was then used as a basis for further implementation. A variation of the waterfall model was used during the implementation and testing by adding a few functionalities at a time. Once a subset of functionalities was added, it was unit tested and integrated. A full system test of 337 test cases was completed at the end of the design.

4. The PFV 16.0 Application

This section explains the high-level architectural design of PFV 16.0 application, detailed design architecture, database design, security design, and user interface along with some detailed usability issues that were factors in its construction.

4.1 High Level Architectural Design

PFV 16.0 was developed using the Microsoft .NET framework and it is written in Visual Basic. An installation of the PFV executable resides on a server machine. The user, from a client machine, will run an instance of the PFV application. The PFV application is a data entry application that retrieves and stores information to an SQL database based on the view, add, update and delete requests or commands requested by the user. The high-level architecture of PFV 16.0 is presented in Figure 1.

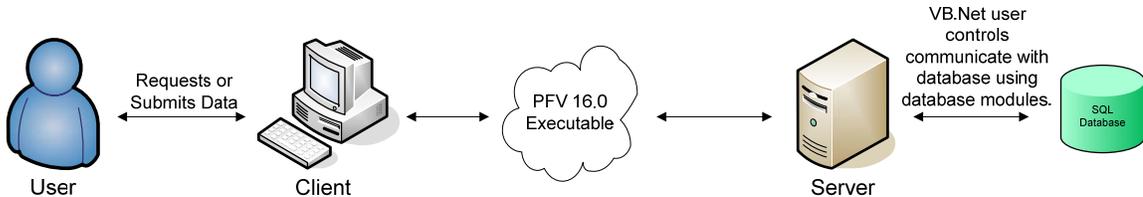


Figure 1. High-Level Architecture of PFV 16.0

The PFV 16.0 application is designed such that the system can be deployed as a single-tiered, two-tiered, or three-tiered application. As a single-tiered system, the application executable and database would be placed on the same machine. In the two-tiered scenario, the clients would be on separate machines and the application executable and database would be on the same server. Finally, in a three-tiered system, the client, the application executable and the database would all be on separate machines. The developer designed the application this way not knowing what the environment resources would be at each state agency location.

4.2 Detailed Architecture of PFV 16.0

The PFV 16.0 application is structured as a collection of classes. The classes form multiple layers: a presentation layer, an application logic layer, and a database layer. The presentation layer is the interface that the users interact with. It presents the data as it is returned from the application layer. The application layer contains all the vital application domain logic. This layer performs all the main processing of the system. It is considered to be the bridge between the user interface and the database. The database layer processes data requests from the application layer and returns the data requested.

Figure 2 illustrates the detailed architecture of PFV 16.0 using a UML Class diagram. At the highest level, the application contains the user login functionality and the main menu which connects to the various functional areas of the application.

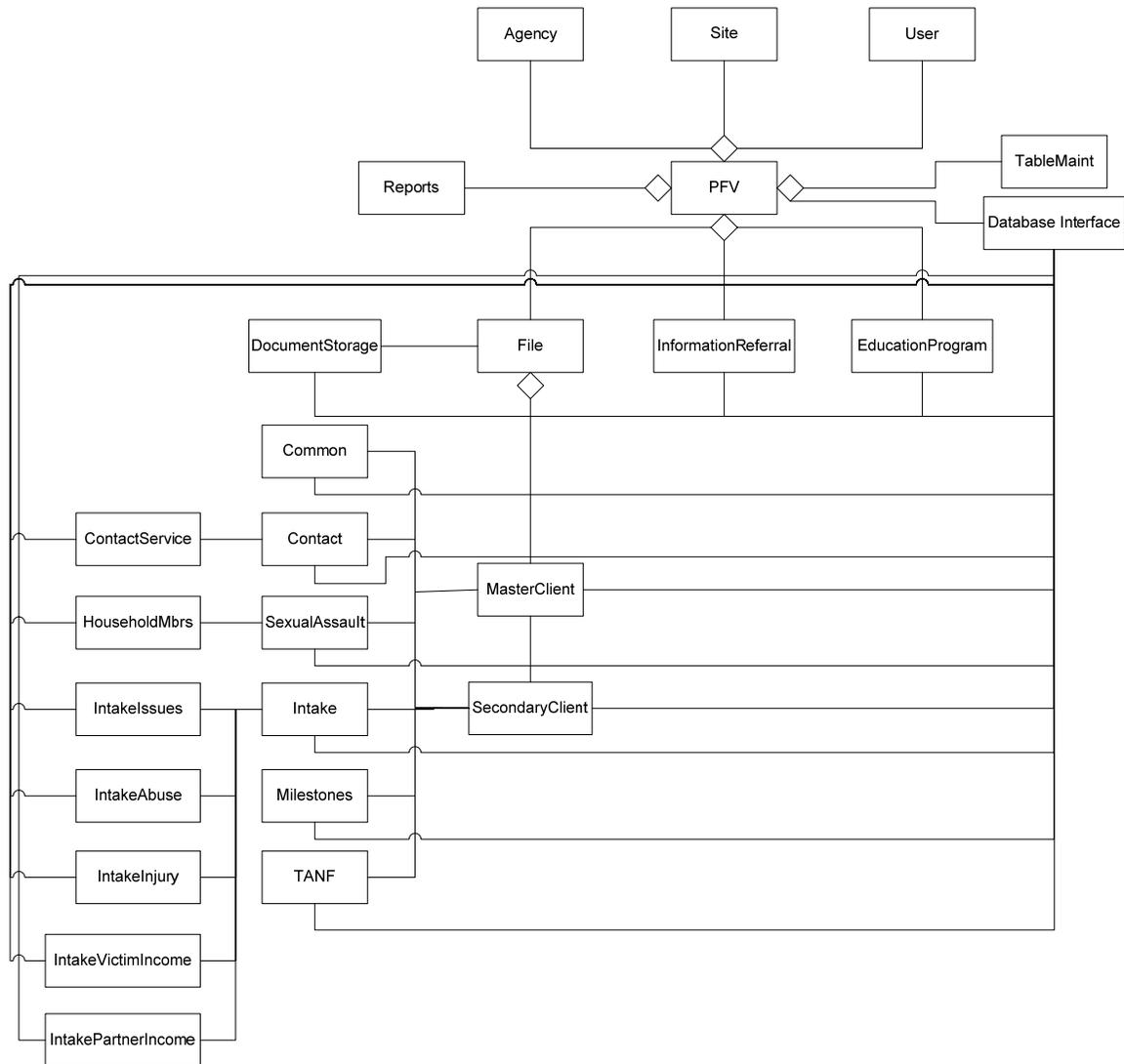


Figure 2. Detailed Architecture of PFV 16.0 (UML Class Diagram)

The user's type and classification will determine their capability with the application. There are four types of users: (1) admin user, (2) viewing user, (3) maintenance user and (4) imaging user. These classifications are not mutually exclusive and a user can have any combination of the four types. Figure 3 illustrates the user classifications in a security use case model.



Figure 3. User Classifications (Security Use Case Model) of PFV 16.0

With the architecture give above, the application is easy to maintain. By designing each layer independently, it is possible to change one layer without affecting (or with minimal changes made to) the other layer. For example, the UI (or presentation layer) can be modified without affecting the application logic. Likewise the database technology can change with little modifications to the database layer and no changes will need to be made to the application layer.

4.3 Database Design

The database design was the most complex part of the project. As discussed during the analysis phase, it was a requirement given by the sponsor to migrate the Access 2000 databases to SQL Server 2000. Since there are 10 databases per install of the application (all with the same schema), it only made sense to normalize the data in the process. The main difficulty in database normalization, when there is existing data already in the tables, is losing the integrity of the data in the process. During the analysis of the two databases, some key differences that affect the project were found; these are:

- Syntactical differences of query statements. This includes the supported clauses and operators, aggregate functions, conversion functions and date functions.
- Data types. Access contains different data types than SQL Server.
- Keys and indexes. Access and SQL Server both have their own way of key creation and indexing.

Syntactical Differences of Query Statements

The syntactical differences among the two databases were small and manageable. This affected the project during in the coding phase, when writing the Transact-SQL query statements. Fortunately, all functions that were coded in the queries of the PFV 15.0 application also had a corresponding function in SQL.

There is a function requirement in PFV 15.0 that will update the age of all master and secondary clients in the system. When the developer ran this update through a direct Transact-SQL statement in the code, the system was timing out. However, since SQL is so robust, the developer was able to create a stored procedure to process the data and return the counts. SQL Server greatly increased the processing time of this function through the use of the stored procedure.

Data Types

The data types, however, became a bigger issue. Table 7 illustrates the differences in the data types and the guidelines used when migrating the databases.

[7]

	Access Data Type	SQL Server Data Type
1	Autonumber (Long Integer)	int (Identity)
2	Binary	varbinary
3	Byte	Smallint
4	Currency	Money
5	Date/time	Datetime
6	Double	Float
7	Hyperlink	ntext (link functionality lost)*
8	Integer	Smallint
9	Long Integer	Int
10	Memo	Ntext
11	Memo	Text
12	Number	Decimal
13	Number	Float
14	Number	Int
15	Number	Decimal
16	Number	Real
17	Number	Smallint

18	Number	Tinyint
19	Number	uniqueidentifier
20	OLE Object	Image
21	Replication ID	varbinary
22	Single	Real
23	Text	Nvarchar
24	Text	Varchar
25	Yes/No	Bit

Table 7. Data Type Differences Between Microsoft Access vs. Microsoft SQL Server

By default, when an Access database is imported into SQL Server, the Access data type Text maps to nText in SQL Server. The nText data type causes problems when using a Crystal Report. The error message received was “Unicode data in a Unicode-only collation or ntext data cannot be sent to clients using DB-Library (such as ISQL) or ODBC version 3.7 or earlier.” This error message is from the SQL Server complaining that it doesn't support pure Unicode via TDS or older versions of ODBC. The error was quickly resolved by changing the SQL data type from nText to nVarchar.

The other main problem with the data types was caused with the design of the original database schema. The original database contained many date fields that were identified as a data type of text. It was originally designed this way because many of the date fields are not validated or required. The developer wanted to make the data types as specific as possible so the decision was made to make this data type a SmallDateTime in SQL Server. The data conversion process, which will be run during the installation of the application, will enter the date 12/31/2050 (a future date chosen at random) for all existing records with a date of empty or null. In addition, two functions were added to the presentation layer: (1) to display an empty field on the screen if the value of the date returned from the database is 12/31/2050 and (2) to

send the application layer class the value 12/31/2050 if the user did not enter a date into the field.

Keys and Indexes

During the analysis phase of the project, it was discovered that the Access databases contained keys and indexes but no true relationships were defined within the database itself. Because of this, the developer had to look at the code and the application domain to identify the relationships among the data. In addition to recreating the existing primary keys and indexes in SQL Server, foreign keys were also created between tables to define the relationships.

Due to the decision to normalize the original 10 databases into one, an additional key had to be created on all of the tables. A key called SiteID was implemented on all tables and added to the primary key. This additional key then had to be implemented throughout the system. The creation of the SiteID key affected the code, most of the Transact-SQL statements and the reporting.

The migration from Access to SQL Server was beneficial. The database went from 10 databases, each with 42 tables to 1 database with 27 tables. The developer wrote a data conversion document explaining the steps of how to migrate the existing data to PFV 16.0. This data conversion document will be provided to the sponsor as part of the implementation tasks. Figure 4 and Figure 5 illustrate the database diagram for the PFV 16.0 system.

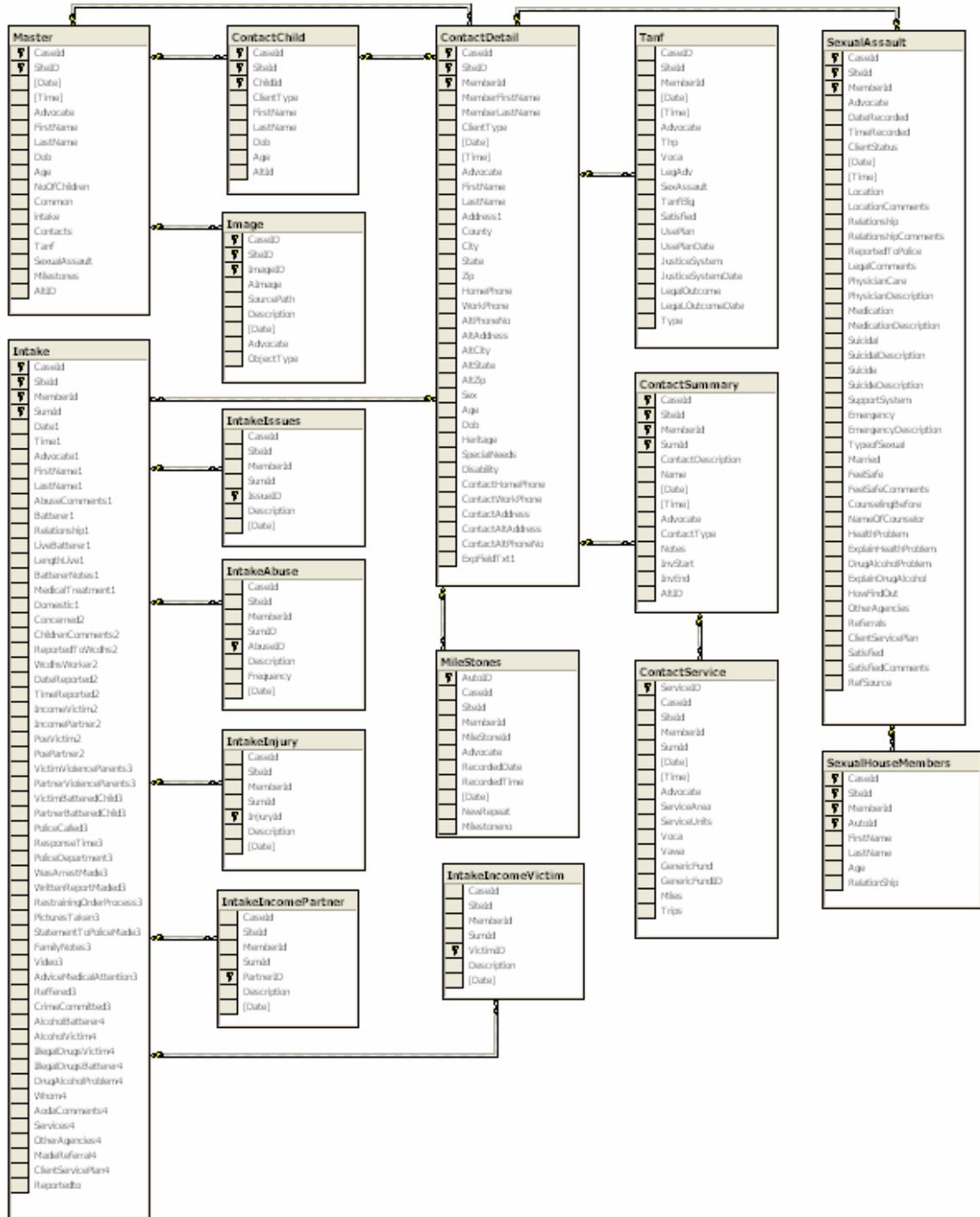


Figure 4. Database Diagram of PFV 16.0 Tables

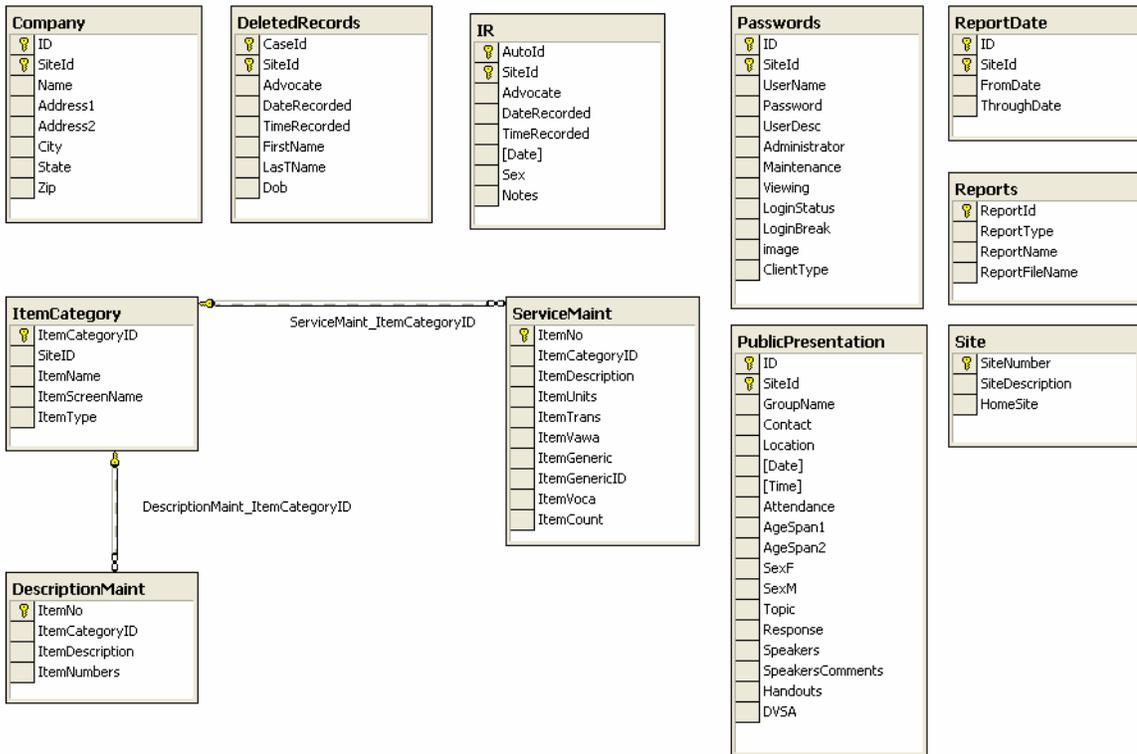


Figure 5. Database Diagram of PFV 16.0 Maintenance Tables

4.4 User Interface Design

The PFV 16.0 application contains a thick client or desktop user interface. The user interface consists of many forms through which the user interacts. The form takes advantage of the Visual Studio .NET controls as well as additionally added controls. The code behind the form is written in VB.NET. VB.NET is written the same way as any other object-oriented programming language using classes, methods, attributes, and the like. Visual Studio .NET takes care of associating the code with the associated form used for the actual presentation layer. The code-behind methodology allows for better encapsulation, easier maintenance and better readability.

Since the users of the PFV system are not computer savvy, the sponsor wanted the interface of the PFV 16.0 application to have a similar look and feel as the PFV 15.0

application. The main menu of the application did not change. However, the developer did combine the current two login screens into one. The developer also designed the main File Module into a tabular format. The tabular format is much easier for navigational purposes. In addition because the application was designed with the layered architecture, the user interface can easily be changed with little or no coding changes to the application layer.

4.5 Security Design

The PFV application captures highly sensitive and confidential information for specific individuals. Security of the system and data is a must.

As a desktop application, the installation of the application will be located within each agency. So only users granted access into the agency network will be able to access the application. Active Directory logins and directory security will be utilized to grant rights to access the application on the network from the desktop. The main purpose of Active Directory is to provide central authentication and authorization services for Windows based computers. Active Directory also allows administrators to assign policies, deploy software, and apply critical updates to an entire organization. Active Directory stores information and settings relating to an organization in a central, organized, accessible database. [2]

By default, the data being sent between the application and SQL Server database is not encrypted (with the exception of the user's password, as discussed below). However, the sponsor indicated that the agency network administrator will setup a VLAN with encryption within the network.

The application contains different user roles. The security officer of the application will assign each user a username and password, if they don't already have one. The system will authenticate the user logging into the application against the credentials stored in the database. The password will be stored in the database as an encrypted string using a 128 bit encryption key. In addition, each agency currently has security practices already in place. The security officer at each agency is

responsible for the security setup of the application and enforcing the policies and procedures when using the application.

Additional security was implemented when migrating the database from Access to SQL Server. SQL Server offers many additional security features that Access did not. By default, in order to access data from a database, a user must pass through two stages of authentication: one at the SQL Server level and the other at the database level. These two stages are implemented using login names and user accounts, respectively. A valid login is required to connect to a SQL Server instance and a valid user account is required to access a database. During a new connection request, SQL Server verifies the login name supplied, to make sure that login is authorized to access SQL Server. This verification process is called authentication. [9] The connection strings for SQL Server authentication are stored and sent as clear text. However, Microsoft does offer additional steps and tips in how to secure that information.

Apart from managing permissions at the individual database user level, SQL Server implements permissions using roles. A role is nothing but a group to which individual logins and users can be added. The main benefit of this is that permissions can be applied to a group instead of applying the permissions to all the individual logins and users. [9]

Once a role is established, the security officer can grant and revoke permissions to and from database users and roles. Permissions can be granted, denied, or revoked to users and roles on all database objects. You can manage permissions at as low as the column level. [9] With these added security features provided by SQL Server, each agency can manage and make the data as secure as they would like. From an application perspective, one application username and password will need to be created to grant the PFV 16.0 system access to the data. It is recommended that another username and password be created for a reporting user since the report user only needs read access.

4.6 *Deploying PFV 16.0*

The PFV 16.0 application is deployed by installing the application on a Windows server that has the .NET Framework installed. The installation will install a configuration file as part of needed files for the application. The configuration file contains server and configuration settings used to connect to the new SQL Server database.

In addition to the install of the application, the existing PFV Access database needs to be migrated to the given SQL Server provided by the agency. The implementation team will use the Data Conversion document to migrate the Access databases at the agency to the SQL Server.

5. Limitations

One limitation of the PFV 16.0 application is the distribution of the application. Since the decision was made to create this application as a thick client application, a new install of the application is required for bug/fixes, enhancements and updates. This means that a list of modifications will likely be compiled prior to deploying a new version of the application.

Another limitation is concurrent users accessing the same file/case. Although the sponsor has said that only one advocate is assigned a case and/or client, there may be times where the users exchange cases and/or clients for backup reasons (if the main advocate is on vacation, etc.). For this reason, there is nothing preventing other users from accessing a case/file that he/she is not responsible for. There is the possibility that more than one user can modify the same case/file at the same time. If this occurs, the application will update the data based on what is on the user's screen at the time of the transaction. That being said, after discussion with the sponsor, concurrency issues do not warrant additional enhancements to the system at this time.

The PFV 16.0 application requires the agency system administrators and/or security officers to be aware and cautious in providing adequate security by appropriately configuring the application and database during installation. The application, by default, does not provide the network security measures.

6. Continuing Work

The PFV 16.0 application was created without the approval from the State of Wisconsin. The developer will package the code, database, data conversion plan and the documentation for the sponsor. The sponsor will then complete additional testing and perform additional enhancements that were not listed in the requirements document. Further modifications to the user interface may also be performed as well to achieve the desired look and feel of the application. The sponsor will then present the application to the State of Wisconsin as an upgrade product available for purchase. The sponsor will be responsible for the creation of the deployment package and implementation of the product at each agency, if the state approves the product.

7. Conclusion

The PFV 16.0 application is a re-engineered desktop application. The new application contains many new enhancements using all new technology. The application is more secure than ever and each agency can set the environment security to the desired level.

The system was designed to capture all vital information when documenting and recording a family violence case. The application will help each agency organize each advocate's case management activities by reducing manual steps, improving productivity and creating a complete accounting of all the clients and services performed. In addition, the application will provides reporting functions and export features to further the efficiency of the advocate.

The application is scalable and has the ability for each agency to customize portions of the application. The PFV 16.0 system contains all required functionalities of the PFV 15.0 application along with additional enhancements provided by the sponsor.

The State of Wisconsin will have the option to buy and use this product or continue to use the existing product. If they choose to continue the use of the PFV 15.0 system, it will be communicated that no additional enhancements or modifications will be made to the application.

BIBLIOGRAPHY

1. <http://www.webster.com/>, Merriam-Webster's Online Dictionary, Merriam-Webster, Incorporated, 2006-2007.
2. <http://en.wikipedia.org/>, Wikipedia, Wikimedia Foundation, Inc., April 2007.
3. A.M. Davis, *Software Requirements Analysis and Specification*, Prentice Hall, 1990.
4. <http://www.donjacobsvw.com/images/weekendspecials/Thick-vs-Thin.pdf>, Thick vs. Thin Client Comparison, App Design Group, Inc., January 2007.
5. <http://www.cypressinland.com/>, YardView, Aldex Software Ltd., 2002.
6. http://www.mssqlcity.com/Articles/Compare/sql_server_vs_access.htm, MS SQL City, Bits on the Wire, Inc, 1997, 2005.
7. <http://www.microsoft.com/technet/prodtechnol/sql/2000/reskit/part2/c0561.mspx?mfr=true>, Microsoft TechNet, Microsoft Corporation, 2007.
8. <http://www.aptest.com/testtypes.html>, APTest Manager, Applied Testing and Technology, Inc., January 14, 2007
9. http://www.sql-server-performance.com/vk_sql_security.asp, Overview of the SQL Server Security Model and Security Best Practices, SQL-Server-Performance.Com, Vyas Kondreddi, 2007.
10. I. Sommerville, *Software Engineering 6th Ed.*, Addison Wesley, 2001
11. http://en.wikipedia.org/wiki/Software_development_process, Wikipedia – Software Development Process, Wikimedia Foundation, Inc., April 2007.
12. <http://www.stylusinc.com/Common/Concerns/SoftwareDevPhilosophy.php>, Software Development Life Cycle (SDLC), StylusInc.com, 2006.

APPENDIX A: Selected PFV 16.0 Screen Shots



The screenshot shows a window titled "PFV System" with a blue title bar. The main content area has a light beige background. At the top, there is a dark brown banner with the text "Welcome to the Prevention of Family Violence System" in white. Below this is an orange banner with the text "Please Log In" in white. The login form consists of the following elements:

- Site:** A dropdown menu with "1 - Production" selected.
- Username:** A text input field.
- Password:** A text input field.
- Exit:** A button located at the bottom left.
- Login:** A button located at the bottom right.

Figure 6. Login Screen

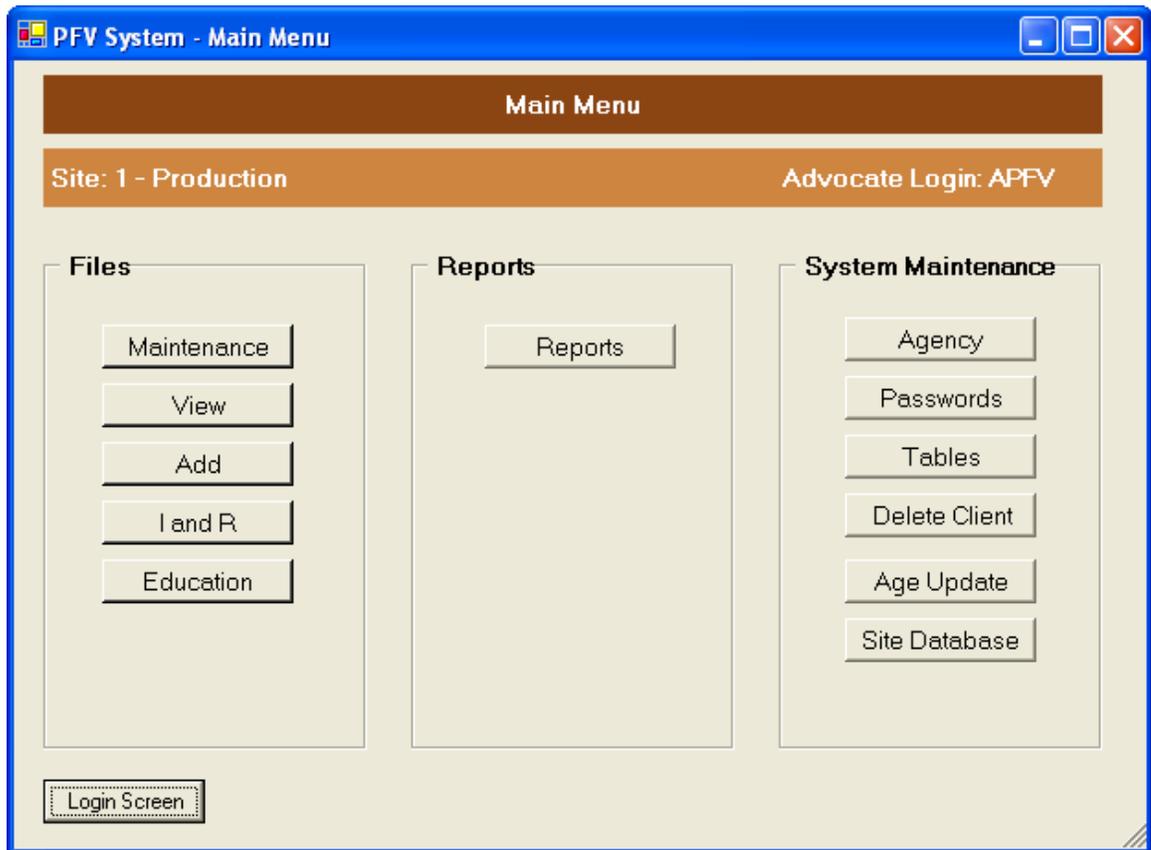


Figure 7. Main Menu Screen

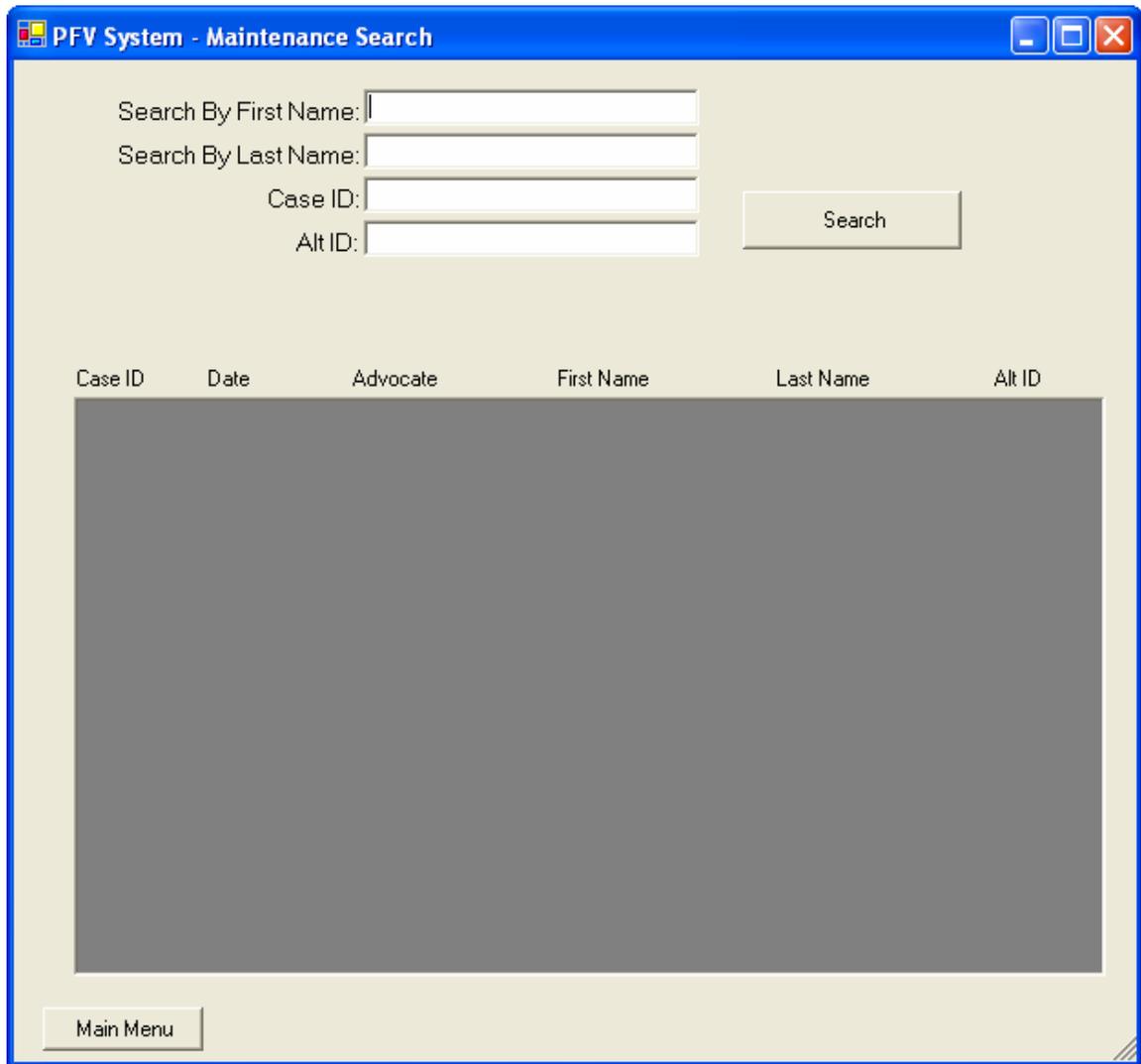


Figure 8. File Search Screen

PFV System - Master

Site Number: **1 - Production** Master Client: Recorded Advocate: **APFV**
Case ID: Current Client: Date: **5/8/2007 3:09:00 PM**

Master Client | Secondary Clients | Common | Contacts | Sexual Assault | DV Intake | Milestones | TANF | Images/Docs

Master Client Information

Case ID: Date: 5/8/2007 3:09:00 PM DT

Alt ID:

First Name:

Last Name:

D.O.B: DT Age:

DV SA

Main Clients File Contents

Common Intake Contact
 Tanf SA Milestones

Figure 9. File Screen – Master Client

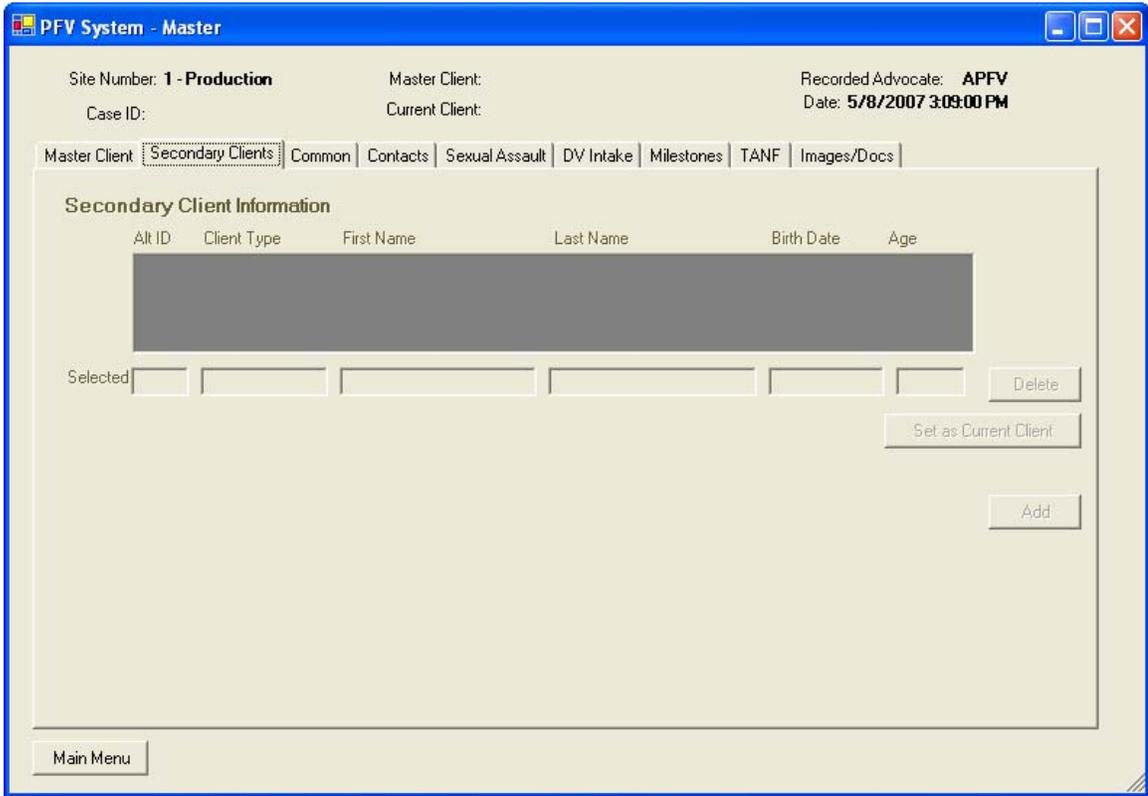


Figure 10. File Screen – Secondary Client

PFV System - Master

Site Number: **1 - Production** Master Client: Recorded Advocate: **APFV**
Case ID: Current Client: Date: **5/8/2007 3:09:00 PM**

Master Client | Secondary Clients | **Common** | Contacts | Sexual Assault | DV Intake | Milestones | TANF | Images/Docs

Common File Information

First Name: _____
Last Name: _____
D.O.B.: _____ Age: _____
Special Needs: _____
Other Special Needs: _____
Disability List: _____
Disability: _____
County: _____
Sex: _____
Heritage: _____
Client Type: _____

Address: _____
City: _____
State/Zip: _____
Home #: _____
Work #: _____
Alt Address: _____
Alt City: _____
Alt State/Zip: _____
Alt #: _____

Add Detail

Main Menu

Figure 11. File Screen – Common Information

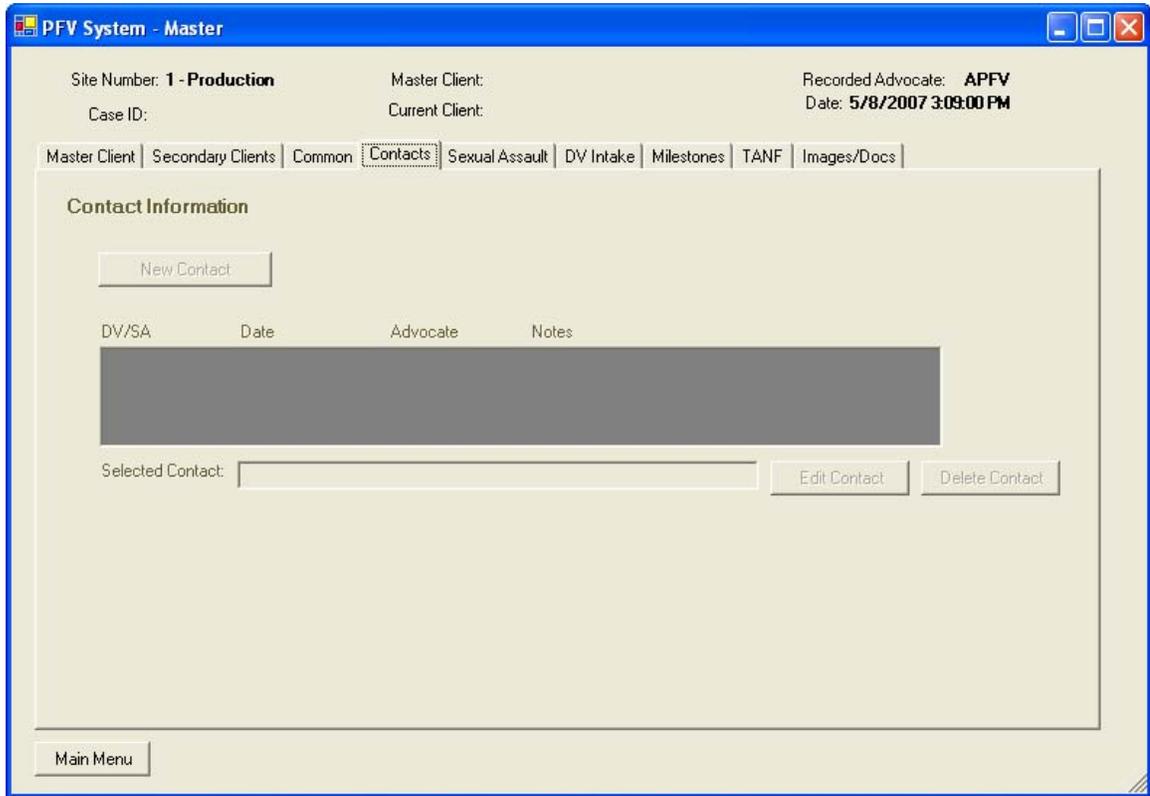


Figure 12. File Screen – Contact Information

PFV System - Master

Site Number: **1 - Production** Master Client: Recorded Advocate: **APFV**
Case ID: Current Client: Date: **5/8/2007 3:09:00 PM**

Master Client | Secondary Clients | Common | Contacts | **Sexual Assault** | DV Intake | Milestones | TANF | Images/Docs

Sexual Assault First Name Last Name Age Relationship Page 1 of 4

Household Members

Record Selected	First Name	Last Name	Age	Relationship	
					Delete
					Add

Are you currently married or in a relationship?

Do you feel safe and comfortable in this relationship?

Have you ever received counseling before?

Name of Counselor:

Have you ever been hospitalized for mental health problem?

If yes, when, where, and for what duration:

Do you currently, or have you in the past, had a drug or alcohol problem?

If yes, please explain:

Comments

Print Continue to Sexual Assault Page 2

Main Menu

Figure 13. File Screen – Sexual Assault Information

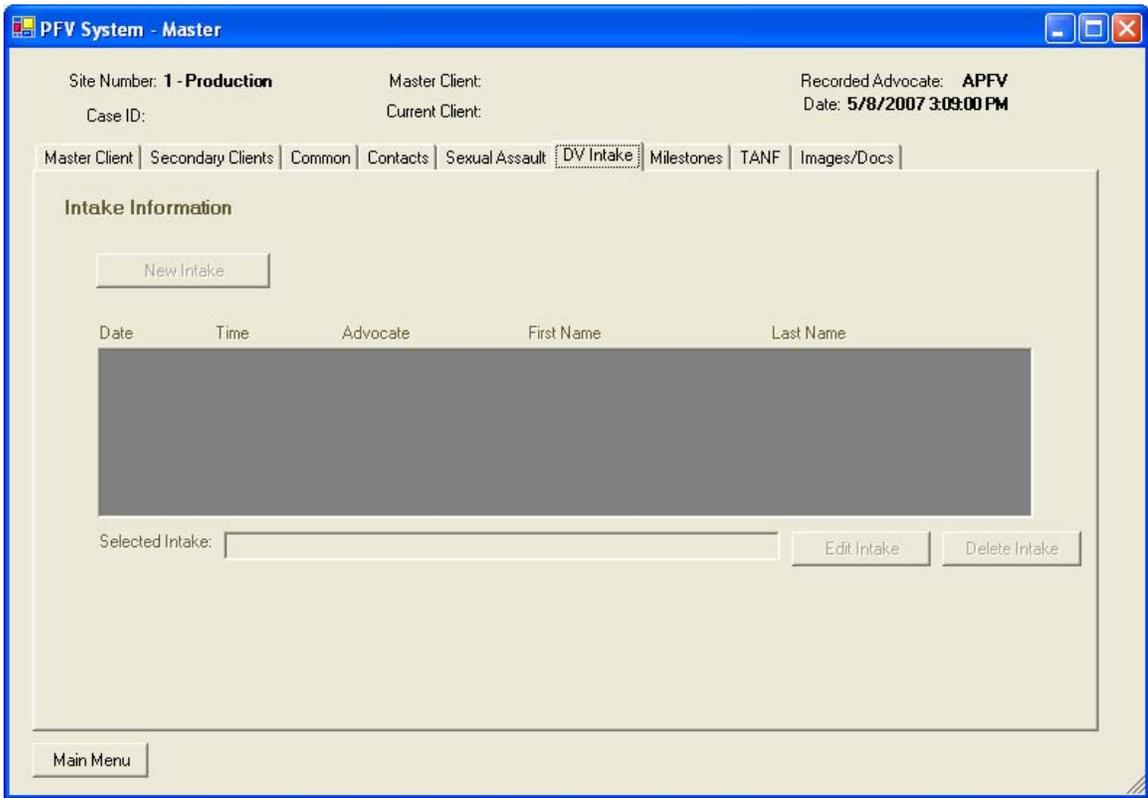


Figure 14. File Screen – DV Intake Information

PFV System - Master

Site Number: **1 - Production** Master Client: Recorded Advocate: **APFV**
Case ID: Current Client: Date: **5/8/2007 3:09:00 PM**

Master Client | Secondary Clients | Common | Contacts | Sexual Assault | DV Intake | Milestones | **TANF** | Images/Docs

TANF

TANF Eligible? DOB: Age:

T.H.P. Client? Legal Advocate Client? VOCA Client? Sexual Assault Client?

Were you satisfied with APFV services?

Received increased information on safety and have a usable plan? Date:

Received increased information regarding criminal justice system? Date:

Received improved legal outcome? Date:

Type:

Figure 16. File Screen – TANF Information

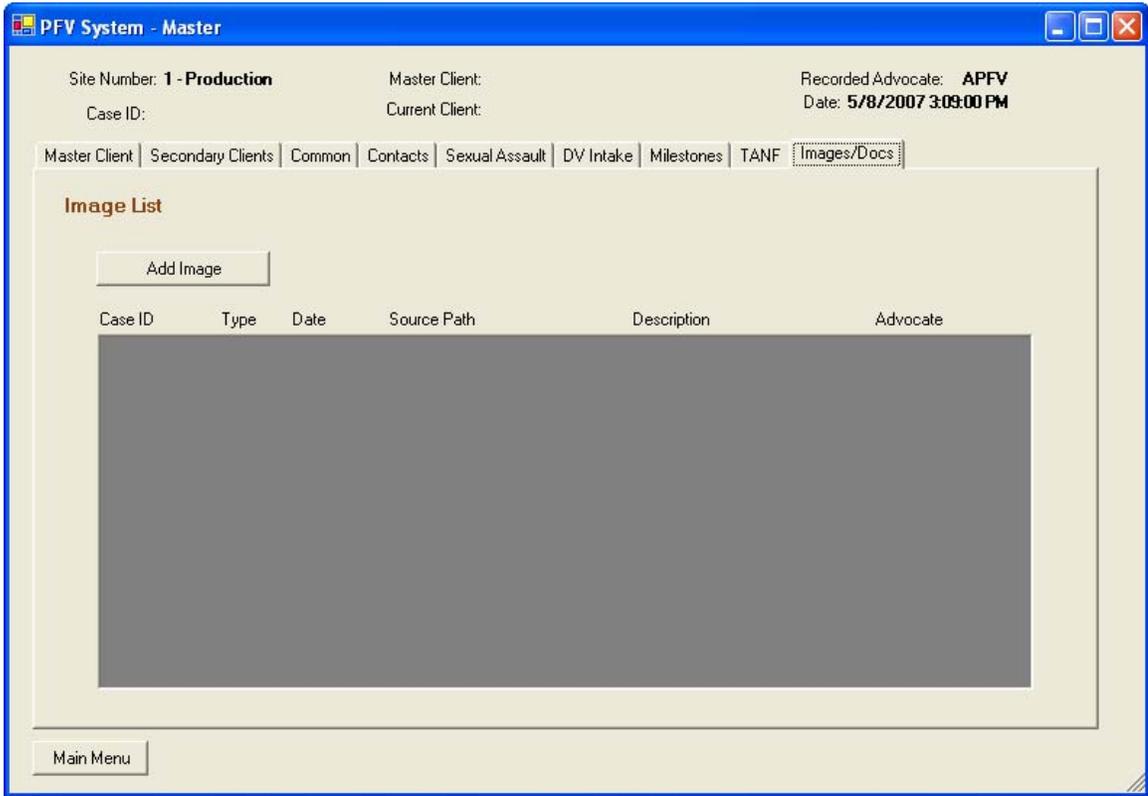


Figure 17. File Screen – Images/Docs Information

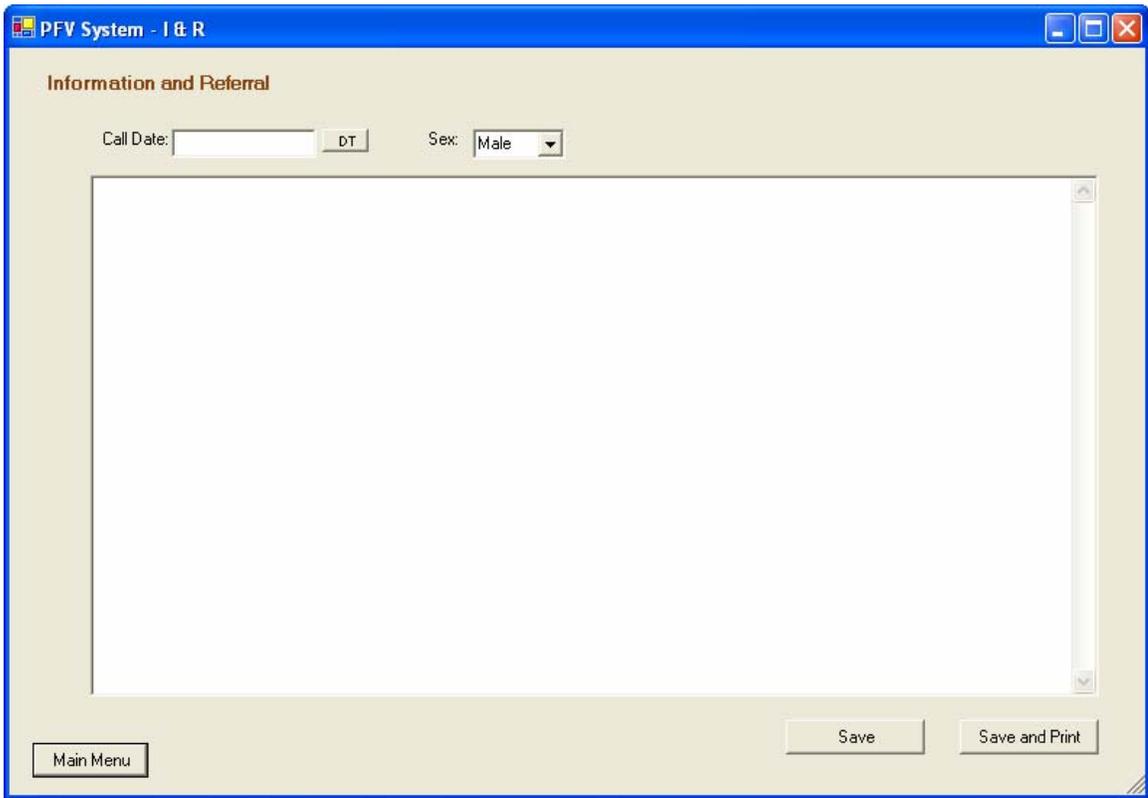


Figure 18. Information and Referral Screen

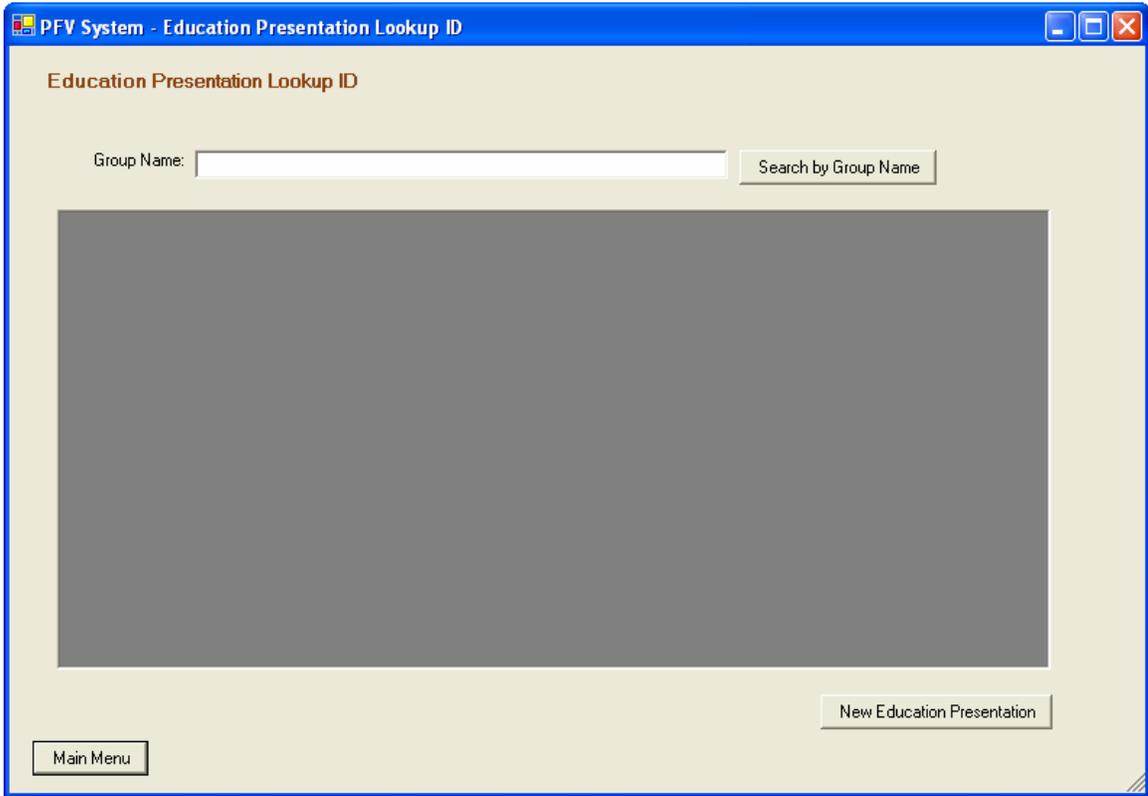


Figure 19. Education Presentation Screen

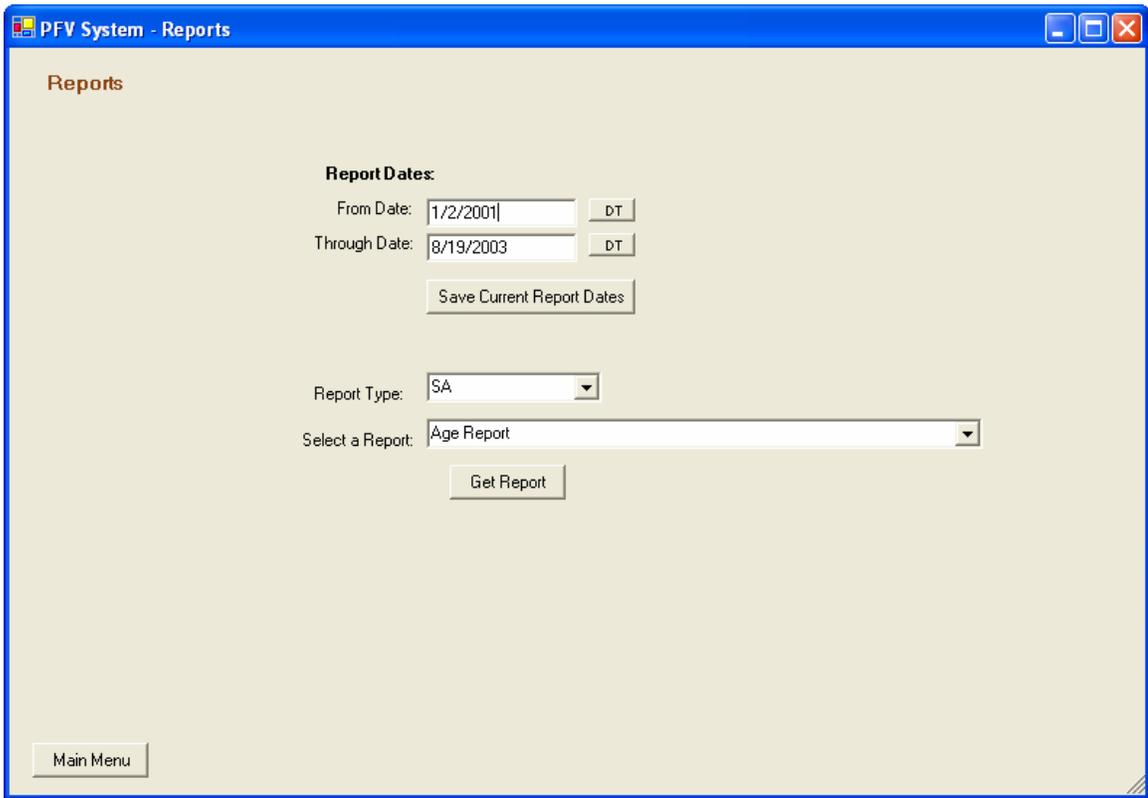


Figure 20. Reporting Screen

The screenshot shows a web application window titled "PFV System - Agency Information". The window has a blue title bar with standard minimize, maximize, and close buttons. The main content area has a light beige background and is titled "Agency Information" in bold. Below the title, there are several input fields for agency data:

- Name:
- Address:
- City:
- State:
- Zip Code:

Below the input fields, there is a button labeled "Save Agency Info". In the bottom-left corner of the window, there is a button labeled "Main Menu".

Figure 21. Agency Information Screen

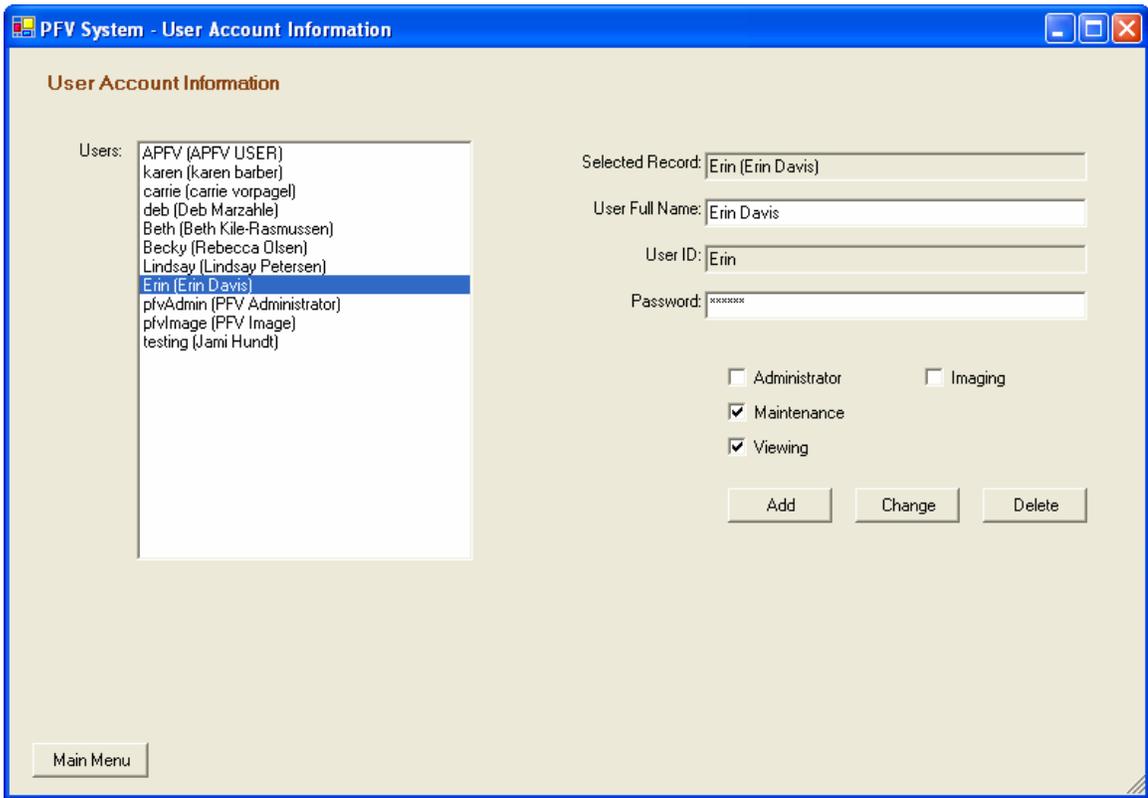


Figure 22. User Account Information Screen

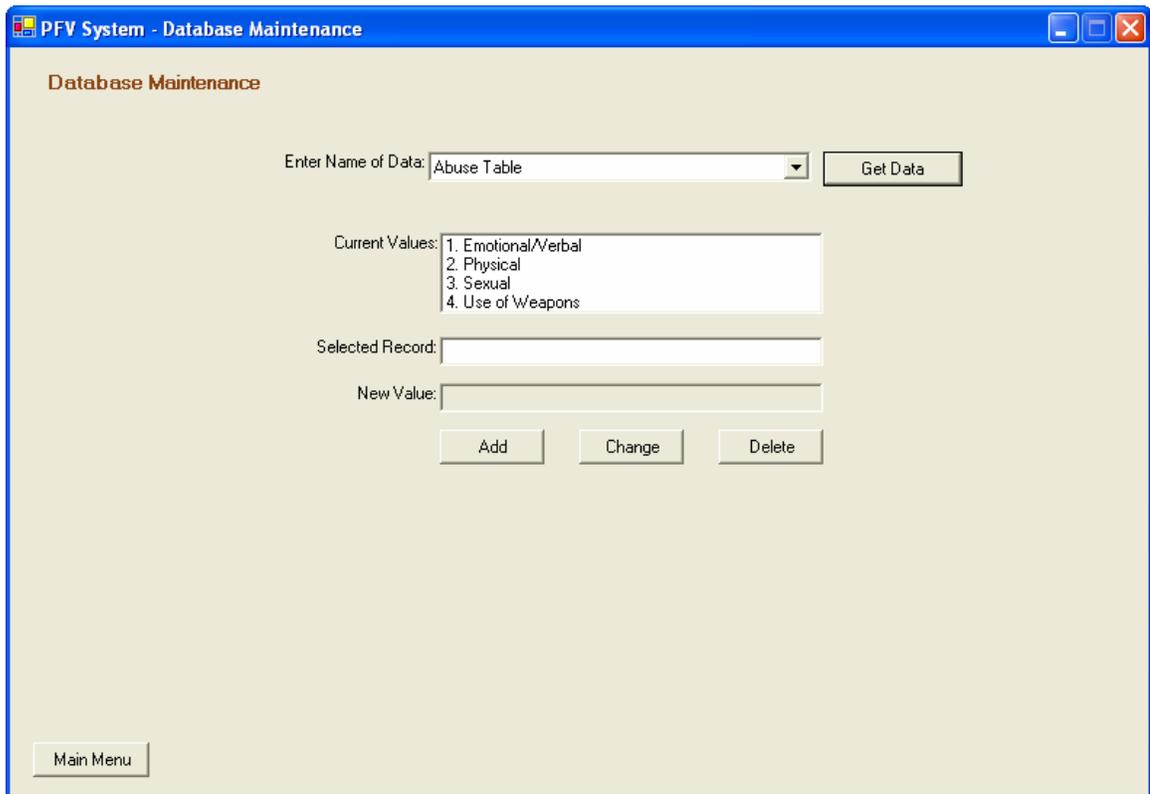


Figure 23. Database Maintenance Screen – Description Maintenance

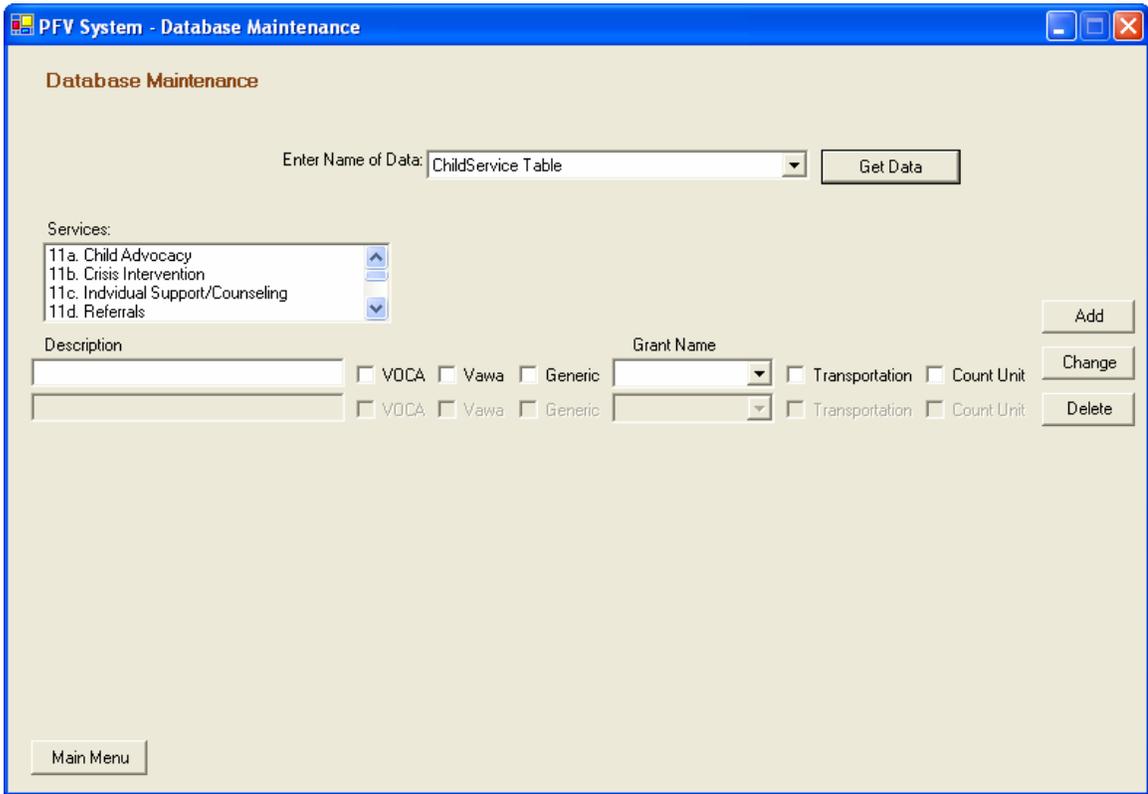


Figure 24. Database Maintenance Screen – Service Maintenance

PFV System - Delete Client

Delete Client

Enter Case ID:

First Name:

Last Name:

D.O.B.:

Age:

Number of Children:

Main Client's File Contents

<input type="checkbox"/> Common	<input type="checkbox"/> Intake	<input type="checkbox"/> Contact
<input type="checkbox"/> Tanf	<input type="checkbox"/> SA	<input type="checkbox"/> Milestones

Figure 25. Delete Client Screen

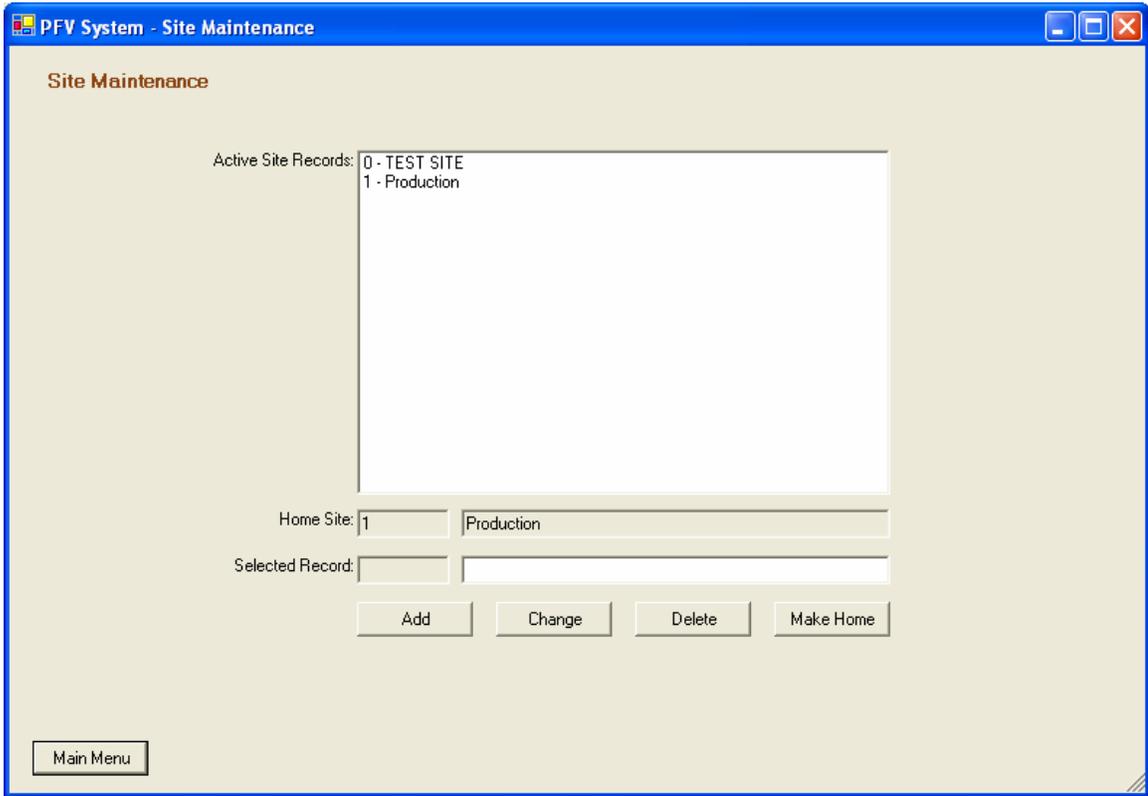


Figure 26. Site Maintenance Screen