Spring 2006

Design an Object-Oriented Home Inspection Application for a Portable Device

Dion M. Collier
Regis University

Regis University
School for Professional Studies Graduate Programs
Final Project/Thesis

Disclaimer

Running head: DESIGN AN OO HOME INSPECTION PDA APPLICATION

Design an Object-Oriented Home Inspection Application for a Portable Device

Dion M. Collier

Regis University

School for Professional Studies

Master of Science in Computer Information Technology

Abstract

Recent advancements in the personal digital assistant (PDA) Windows application programming methodology made it easier to develop PDA applications. The release of the Microsoft® Visual Studio 2005 .NET incorporated handheld programming support while the Microsoft® Mobile™ 5.0 operating system dramatically improved the PDA's operation and hardware configuration. This paper researches and analyzes object-oriented languages, relational database and dynamic report generation technologies for the PDA as they apply to the development of a professional home inspection application. The focus of this paper is on the implementation of the most advanced PDA technologies for a high-end database PDA application design.

Dedication

This thesis is dedicated to my beloved wife Tracy for supporting my pursuit of my MSCIT degree and to our two beautiful daughters, Makena and Sierra.

Table of Contents

Design an Object-Oriented Home Inspection Application for a Portable Device

Feel Good Home Inspection LLC has been providing commercial and home inspection services to property owners and potential buyers. The average home inspection could take the inspector(s) up to four hours to conduct. All property component conditions, problem and safety issues are documented within the final inspection report. The customer could then use the report as a financial decision tool in order to renegotiate or complete the property transaction.

Problem Statement

The American Home Inspection Training Institute (AHIT®) Home Inspection Report has been used by Feel Good Home Inspection LLC for legally documenting property inspections. Included in the report are the inspection agreement, property documentation forms, and the inspection receipt. Completed reports must be kept for seven years according to the American Society of Home Inspectors (ASHI) standards of practice, Georgia and South Carolina state laws, and liability protection and taxation reasons. Storing these large paper reports over several years has been a spatial and financial burden to the company. Another major burden been the amount of time it took the inspector to handwrite the report and assemble report copies for all the parties involved. The inefficient paper system would be changed to a more efficient electronic system to create a reduction in the operation costs while increasing the customer's satisfaction. The professional project was to research, analyze, and design an object-oriented (OO) home inspection application targeted for a personal digital assistant in order to resolve these problems.

Business Case

Feel Good Home Inspection LLC owner, Terry Ryan, wanted a custom home inspection application that would run on a PDA. The new application would help the company reduce the

business operating costs while at the same time increase profits. The first targeted business goal was to replace the current paper system with an electronic system to save both time and money by not purchasing and handwriting the inspection reports. The second goal was to increase the customer's satisfaction by reducing the time to inspect a property by twenty-five percent and immediately producing the final report after the inspection. The final optional goal was to market the home inspection PDA application as another source of income.

Project Goals

The main project goal was to design an object-oriented home inspection application for a PDA. The final design was successfully based upon the technological and business requirements discovered during the project's requirement and preliminary design phases. Three deliverable documents were successfully delivered at the end of the project and could be used to continue the application development outside of the professional project. The deliverables contained information on the home inspection and technology requirements needed in order to start the Systems Development Life Cycles (SDLC) Coding Phase.

*Home Inspection System Architecture***.** The proposed home inspection system architecture was based on using the PDA at the job site. Collected inspection data would be stored long term on the PDA with a powerful relationship database management system (RDMS). Digital photographs would be imported into the application by plugging the camera's media storage card into the PDA. The application would then prompt the inspector to assign each photograph to a particular inspection section and prompt for comments. The final inspection report would be generated after the application verified that all of the information had been collected. The report would then be printed through the wireless 802.11b network to a mobile printer.

Incorporating a home based desktop server to collect and store the home inspection data was found to be a desirable option for future development. However, this option was found to be outside the scope of the project and would be investigated after the implementation of the PDA home inspection application.

*PDA Hardware***.** Thorough research was conducted on PDA hardware technologies and available options. The PDA hardware requirements for the home inspection application were generated in order to select the optimum PDA and accessories for the home inspection system.

*PDA Operating Systems***.** The owner's project requirement was to use a Windows® based PDA operating system (OS). He felt the Windows® Mobile™ OS looked and operated similar to Windows® XP which he was already familiar with. The recently released Microsoft® Mobile™ 5.0 OS and the older Microsoft® Mobile™ 2003 Second Edition OS were researched as possible application platforms.

*PDA Object-Oriented Languages***.** Both C++ and Java™ object-oriented languages were heavily scrutinized for developing the PDA home application system for the Microsoft® Windows® Mobile™ 5.0 OS. Implementing an object-oriented based language would allow the object-oriented analysis and design (OOAD) process to be used on the project. This process would allow the author to break down the application problems into more manageable pieces by using the Unified Modeling Language (UML™). Charles Richer stated that the UML™ diagrams provided several different views of a system (Richter, 1999, p. 10). These various views would provide a better understanding of the problems and how to address the problems. The end result of the OOAD would produce several great benefits for the PDA home inspection application: reusability, extensibility, reliability, maintainability and robustness.

PDA Data Storage Systems. The electronic storage of the final inspection report would provide a ninety percent paper storage reduction for Feel Good Inspections LLC. The relatively large final inspection report, ASHI® Standards of Practice, and Code of Ethics would now be saved electronically. The financial savings would come from the reduced office space needed for very expensive fireproof filing cabinets. The software application would also allow the generation of past inspection reports from the database upon demand. The PDA data storage investigation involved searching for PDA support for Extensible Markup Language (XML) and stand-alone relational database systems (RDBS). The chosen data storage system should provide quick data access and reliable data retention.

PDA Dynamic Report Generation. The overall goal of the home inspection application was to provide a thoroughly completed final inspection report to the customer. This report would be generated from all of the data collected during the property inspection. The idea was to only provide information about the property and not have any uncompleted sections. For example, the paper AHIT® inspection report contains four bathroom sections. If a home with only one bath is inspected, then there would be three blank bathroom sections shown on the report. The customer and the owner of Feel Good Inspections LLC do not want to have these blank sections on the final report. The final inspection report should be generated dynamically to only display the information collected.

Project Deliverables

The professional project deliverables were outcomes from the first three phases of the Systems Development Life Cycle Phases. The completion of these documents signified the completion of the SDLC Design phase and that the required project deliverables were

successfully completed. All three project deliverables would be needed in the remaining SDLC phases to complete the home inspection application.

"*Requirements Definition" Document***.** The "Requirements Definition" document was created within the SDLC Requirement Analysis Phase and maintained in the Preliminary Design and Design phases. This document incorporated the home inspection system requirements and constraints, system architecture layout and the project task plan.

"*Preliminary Design" Report***.** The SDLC Preliminary Design Phase resulted with the completion of the "Preliminary Design" report. This report identified the specific PDA and configuration options requirements, system hardware requirements, programming Integrated Development Environment (IDE), database IDE, utility Commercial-off-the-Shelf (COTS) applications, and the projected project expenses.

*"Final Feasibility Design" Document***.** The completed "Final Feasibility Design" document was the last project deliverable. This document was the result of completing the SDLC Design Phase. The contents of this document provided the object-oriented design (OOD) aids to direct the creation of the object-oriented C++ code: Use Case System Diagram, Use Case Scenarios, Sequence Diagrams, and the Static Class Diagrams. An Entity-Relationship Diagrams (ERD) was provided to aid in the development of the PDA relational database.

Project Significance

Feel Good Home Inspections LLC foresaw many advantages to developing a PDA professional home inspection application. The biggest advantage came from electronically storing final home inspection reports for long term referencing. This single goal eliminated the need from physically storing a paper report copy for seven years and from purchasing the expensive preprinted blank inspection reports. It was expected that the customer's satisfaction

would be increased due to the twenty-five percent reduction in the amount of time to generate the typed final inspection report. The owner wanted the capability to add or modify inspection questions, safety issues and problems in order to provide an updated and better product to the customer. Inspection methods and points of interests changed frequently and this option would allow the inspector to quickly implement the inspection updates into the application. Feel Good Home Inspections LLC projected a thirty percent increase in the company's profit by developing the home inspection application. Additional profits could also be created by marketing the application to other home inspection professionals.

The project stemmed from the author learning to analyze and design an object-oriented home inspection PDA application without any prior PDA experience. The PDA device may have provided many good unique features due to its small physical size, but it also had many operational and software disadvantages. Recent PDA hardware, PDA operating system and development IDE technological have changed the programming methodology for portable devices. This paper discusses the improved PDA programming methods as well as its short comings. This paper also addresses how the object-oriented C++ PDA application with database support was analyzed and designed.

Barriers and/or Issues

The largest barrier to overcome was the author's lack of knowledge on the home inspection profession. The owner shared his knowledge with the author and provided great research references on the home inspection profession. In return, a brief SDLC overview was given to the owner so that he would know more about software development management.

The next issue was that neither the author nor the owner had any previous experience with operating a PDA. The author had no prior knowledge about designing and programming PDA

applications. A lot of additional project research time was needed in order to learn about programming PDAs.

Questions to be Answered or Discovered

There were several PDA technical questions that needed to be researched and answered. These questions revolved around the special considerations for designing and programming a PDA application.

PDA Operation & Limitations

Most of the technical PDA operation and limitation questions were discovered throughout the research on PDA devices. These questions included: *What are the PDA storage limitations? What touch screen options are available? How long can a PDA operate without recharging the battery?* A few more questions were discovered during the PDA research and answered in chapter 4: *Will the slower PDA processing speeds affect the design? How can data loss due to PDA system lockup or shutdown be prevented?*

Programming the PDA

Since the author was new to the PDA programming environment, several questions had to be answered to become knowledgeable of the latest PDA software development technologies. This led to the following questions: *Is either C++ or Java™ programming languages supported on PDAs? What IDEs are available for the PDAs? Does the PDA operating system limit programming language support?*

PDA Database Options

The home inspection PDA application needed the capability to manage a huge amount of data. The decision to implement XML or RDMS technologies was heavily influenced by the PDA programming limitations. Implementing XML or a relational database solution for the

PDA was conducted to answer the following technical questions: *Is there support for XML or RDMS programming on the PDA? What COTS database systems are available for the PDA?*

Dynamic Report Generation

The main outcome of using the PDA home inspection application was to dynamically generate the final inspection report. The main question for a non-experienced PDA programmer was how to dynamically generate documents on the PDA. *Could the final report be generated as a Microsoft® Mobile™ Word, rich text format (RTF) or Hypertext Markup Language (HTML) formatted file?*

Project Scope

The professional project scope was to focus on researching, analyzing, and designing an object-oriented home inspection application for a portable device. Only the first three SDLC phases were required for the professional project: Requirement Analysis, Preliminary Design, and Design. The final application design was based on the captured system requirements and from the portable device's software and hardware capabilities. The project ending deliverable goals were to produce the "Requirements Definition" document, "Preliminary Design" report and the "Final Feasibility Design" document. These documents would be needed in order to complete the remaining four SDLC phases outside of project: Coding, Testing, Implementation, and Maintenance Phases.

Definition of Terms

AHIT® – American Home Inspection Training Institute

ASHI – American Society of Home Inspectors®

COTS – Commercial-off-the-Shelf

CCS – Cascading Style Sheets

ERD – Entity-Relationship Diagram

GUI – Graphical User Interface

HTML – Hypertext Markup Language

IDE – Integrated Development Environment

J2ME - Java™ 2 Micro Edition

J2SE - Java™ 2 Standard Edition

JVM - Java™ Virtual Machine

OO – Object Oriented

OOAD – Object-Oriented Analysis and Design

OOD – Object-Oriented Design

OOP – Object-Oriented Programming

OS – Operating System

PC – Personal Computer

PDA – Personal Digital Assistant

PDF – Portable Document Format

RAM – Random Access Memory

RDMS – Relational Database Management System

ROM – Read-only Memory

RTF - Rich Text Format

SD™ – Secure Digital

SDK - Software Development Kit

SDLC – Systems Development Life Cycles

UML™ - Unified Modeling Language

WYSIWYG - What You See Is What You Get

XML - Extensible Markup Language

Summary

The project focused on designing an object-oriented PDA home inspection application. The object-oriented analysis and design of the home inspection PDA application would be directed at converting Feel Good Home Inspection LLC's costly and inefficient paper system into an automated inspection system. A lot of additional project research time was needed to learn about programming PDAs. The final project outcome would be signified by the completion of the Requirement Definition document, "Preliminary Design" report and the "Final Feasibility Design" document. Each these documents should be the final output from the SDLC Requirement Analysis, Preliminary Design, and Design Phases.

Chapter Two: Review of Literature & Research

Overview of Project Literature and Research

Designing the home inspection PDA application required extensive research in two fields. The project involved combining the home inspection field knowledge with the PDA technology in order to design the home inspection application. The owner provided the research direction for a better understanding of the home inspection profession. This included literature and websites from respectable and well known home inspection organizations, completed final inspection reports as well as personal experience. The intent was to provide the author with enough home inspection knowledge to be able to design the inspection application.

Secondly, intensive research was required to learn about programming applications for the PDA since the author did not have any experience with coding or operating a PDA. The latest PDA technology and Microsoft® Windows® Mobile™ 5.0 OS release happened to be released while the research was being conducted. This resulted in getting the latest information from the Internet and newly published PDA magazines.

Specific Literature and Research Relevant to Project

Most of the home profession information came from ASHI. Their Standards and Practice (American Society of Home Inspectors, Inc.®, 2004) and Home Inspection Report (Newcomer & Newcomer, 2003) provided the foundation of understanding what is documented during an inspection. The owner provided some copies of completed inspection reports so that the author could get a better feel of the handwritten information. Several informal conversations between the author and owner were needed to clarify home inspection questions.

Researching how to program a Windows® based PDA application was the second part of the project. Most of the literature could only be found on the Internet or in recently published magazine articles. This was due to the recent release of the Microsoft® Windows® Mobile™ 5.0 OS and the new PDA hardware features. Published books on these topics were not available during the research.

PDA Operations & Limitations

PDAs running the Microsoft® OS overthrew the Palm® OS for the first time according to the 2004 PDA shipment market share report (*Report: Windows CE top PDA OS in 2004*, 2005). This signified the starting downfall of the Palm® OS based PDAs which was attributed mostly to the Microsoft® based OS PDAs. By the second quarter of 2005, Microsoft dominated the shipment market share with 45.6% compared to Palm® OS 18.8% which was down 40.9% from the previous year (*PDA market swells Windows CE holds lead*, 2005). The new king of the PDA operating systems was very evident with the majority of Microsoft® base OS being offered in the local electronic sales compared to the Palm® OS.

The PDA has limited processing power compared to a personal computer. The typical PDA has anywhere from 300 MHz to 624 MHz processors compared to 2.4 to 3.2 GHz processors on the personal computer (PC); however, the higher processing speeds required more power. The slower PDA processing speed was intended to save power since the PDA typically runs on a battery. Other obvious limitations included display screen size and available memory storage.

The Dell™ Axim™ X51v PDA running the Microsoft® Windows® Mobile™ 5.0 OS has received very impressive reviews (Dumas, 2005). What made this PDA popular was the latest Microsoft® OS which supported several enhanced hardware features. The most noticeable

feature was the large 3.5 inch color TFT 16-bit touch sensitive LCD (*Axim™ X51v Details*, 2005). The screen resolution was increased to 240 x 320 at 65,536 colors (QVGA). The Intel® XScale™ PXA270 624MHz processor along with the Intel® 2700G graphics accelerator provided very quick performance and graphics updates. The read only memory (ROM) was increased from 128 megabytes (MB) to 256MB to support the new OS persistent storage feature and 64MB random access memory (RAM) is built-in. Unique expandable memory storage features include CompactFlash™ Type II and Secure Digital (SD™) slots. Integrated 802.11b and Bluetooth® technologies are available for networking. An optional removable 2200mAh Lithium-Ion rechargeable extended battery could be purchased to provide seven hours of operation time (Brown, 2005). Three year limited and accidental warranties could be purchased from Dell™ at a very low price.

The Microsoft® Windows® Mobile™ 5.0 OS redesigned the method data and programs were loaded into memory. Previous Microsoft® operating systems loaded both data files and programs into volatile RAM. This meant that a power loss would cause data file and program losses. The new operating system addressed this issue by reconfiguring the memory usage simulator to the personal computer (Brown, 2005). Volatile RAM was now used just for running programs, which at the same time, increased the amount of memory available by removing the data file storage. Now the 64MB of RAM was available for active programs and 22MB of the RAM was taken for the operating system.

The PDA ROM size was doubled from 128MB to 256MB (196MB usable) and used the very fast non-volatile Flash™ memory technology. This meant that data files written to ROM would not be lost upon power loss or hard resets. At the same time the Flash™ technology reduced the power drain on the PDA. The Flash™ ROM has slower read/write speeds than the

volatile RAM, but it was much faster than accessing data stored on compact storage expansion cards. The revamping of Microsoft® Windows® Mobile™ 5.0 operating system's memory management along with converting the ROM to Flash™ chip technology dramatically decreased power consumption and secured the data files.

The Microsoft® Windows® Mobile™ 5.0 OS was made more efficient than the previous OS. This resulted in faster processing and extended battery life. ActiveSync 4.0 has been greatly improved to allow easier and faster connections to the desktop. Vast improvements were made to maintaining the network connections, file conversions, and error handling. ARM based processor application installations are supported with ActiveSync 4.0, including to the Visual Studio 2005 .NET PDA emulator (De Herrera, 2005).

PDA Object-Oriented Programming Options

There were issues with Java™ programming on a handheld device running the Microsoft® Windows® Mobile™ 5.0 OS. The main one being that Microsoft® does not include a Java™ Virtual Machine (JVM) with their Mobile™ operating systems. There are several JVM third party vendors that offered a JVM for the older Microsoft® Mobile™ 2003 Second Edition OS but not the latest version. It was surprising that the Java™ vendors were not prepared with a fully functional JVM that would work with the Microsoft® Windows® Mobile™ 5.0 OS.

Further investigation revealed that programming methodology really depended upon the chosen PDA operating system. Rich Hall's article *What is Windows Mobile?* in *Smartphone & Pocket PC* magazine's Buyer's Guide 2006 quickly covered the four types of handheld devices that are supported by Microsoft® Windows® Mobile™ 5.0 operating system (Hall, 2005). They included the PDA (also known as Pocket PC), Pocket PC Phone Edition, Mobile Smartphone and the Mobile Portable Media Centers. Mr. Hall's article explained how the familiar Windows®

operating system allows users to easily operate the different type of devices. The Microsoft® Windows® Mobile™ 5.0 OS was developed as the new foundation platform for all handheld devices.

Microsoft® answered the development community's request to provide one common platform for all handheld devices and one programming IDE. On November 7, 2005, Microsoft® launched the long awaited Visual Studio 2005 .NET version 8.0 IDE. The latest Visual Studio 2005 .NET IDE included Visual Basic, Visual C++, Visual C# and Visual J# programming languages. These languages could produce applications that ran on the latest or legacy Microsoft® Windows® operating systems. Visual Studio 2005 .NET was available in four editions: Standard, Professional, Team Systems and Express Editions. All four Visual languages were offered separately in the low cost Express Editions, which were targeted towards the hobbyists, amateurs and small businesses (Russell, 2005). The .NET Compact Framework 2.0 was incorporated into the Microsoft® Visual Studio 2005 .NET IDE which provided handheld device programming support. Microsoft® Mobile™ 5.0 OS included the improved .NET Compact Framework 2.0 which has several positive features: improved developer productivity, threading classes that were more similar to the full .NET Framework, simplified communication classes, improved compatibility with the full .NET Framework, increased runtime performance, expanded cryptography support, enhanced XML classes, and increased number of supported devices.

The Microsoft® Mobile™ 5.0 Software Development Kit (SDK) could be added to Visual Studio 2005 .NET to provide the capability to write managed and native applications running on the Microsoft® Mobile™ 5.0 OS. The SDK kit included Pocket PC Device Emulator images and skin files, headers, libraries, interface definition language (IDL) files, managed reference

assemblies, documentation, Test Authenticode Certificates, tools, sample configuration files, and sample code. Mobile applications could be easily tested using the portable device emulator. ActiveSync 4.0 provided the capability of installing PDA applications on the emulator. Internet connection was available to the emulator for testing Internet based applications. The improved mobile emulator provided easier and faster software application testing. The object-oriented home inspection application could be developed by using Microsoft® Visual Studio 2005 .NET C++ IDE with the Mobile™ 5.0 SDK.

Visual Studio 2005 .NET provided a designer view which had the capability of displaying the handheld device's physical frame along with the screen (Struys, 2005). PDA screen development was made easier with the what-you-see-is-what-you-get (WYSIWYG) designer view. The wide selection of form controls allowed the author to quickly create screen layouts and interface to databases. Both portrait and landscape modes are supported which allowed the author to verify proper screen layout in either mode.

PDA Database Options

The PDA home inspection application required the capability to store a lot of information on to nonvolatile memory. One of the owner's requirements was to implement the application on a PDA running the Windows® Mobile™ operating system with the total COTS PDA software licensing fees under $300. Thus the research was focused on programming IDEs and database solutions that were compatible with the Windows® Mobile™ operating system. Both XML and RDMS data storage technologies for portable devices were researched.

XML uses standardized formatted text component tag identifiers to describe a non-relational database schema (Benz, 2003). Implementing XML technology would be a good and free local data storage solution for a PDA, but the Microsoft® Visual Studio 2005 .NET

Compact Framework 2.0 only provides a basic set of XML classes (Wildermuth, 2005). Advance features like schema verification and XPath which provide data selection capability within a XML document were not incorporated within the Compact Framework (Wildermuth [2005]; *Extensible Markup Language* [2005]). The entire XML formatted file would have to be read into volatile RAM before the data information could be processed. Reading the file and saving the file would take a long time due to the PDA's slower processors. A great amount of data loss could occur from power loss or if the PDA operating system locks up and requires rebooting.

Another PDA database solution was to implement a RDMS that supported ANSI SQL. This option would allow the application to make a connection to the database that resided in PDA's nonvolatile memory. Frequent home inspection data could be quickly inserted or modified through the database connection. The iAnywhere® UltraLite™ relational database product directly addressed this issue by providing an affordable and small footprint database with very powerful relational database features (*UltraLite* [2005]; *SQL Anywhere Studio – UltraLite Deployment Technology* [2005]). Selecting only the database features needed on the PDA allowed for the smaller database footprint. Programming support was provided for Microsoft® Visual Studio .NET, Visual C++, and other IDEs. Database access can be done through ADO.NET or ODBC 3.5 (iAnywhere® Solutions, Inc. [2004]; iAnywhere® Solutions, Inc. [2003]).

The author would benefit from the free SQL Anywhere® Studio graphical database tools to create and deploy the client-server databases. Databases can be deployed for a very large selection of computer and handheld devices operating systems (*SQL Anywhere – Broad Platform Support*, 2005). Synchronization software can be installed with the databases to support

client/server applications. SQL Anywhere® Studio includes several graphical database tools for database design, reverse-engineering, database management, browser, query plan viewer, query editor, integrated stored procedure debugger, profiler and synchronization monitoring.

The Sybase® iAnywhere® UltraLite™ relational database with Sync 9.0 offered several of the desired benefits of the larger enterprise databases but at a very affordable rate of $119 per license (*SYBASE® eShop UltraLite*, 2005). iAnywhere® also offers the SQL Anywhere® Studio 9.0.2 for Windows for only $399 which included several one seat licenses for server, network seat, Adaptive Server Anywhere Personal Database with synchronization, and for the UltraLite™ with synchronization (*SYBASE® eShop SQL Anywhere Studio*, 2005). The studio package would satisfy the current PDA database requirements and future server requirements with one very affordable price. Additional specific licensing seats can be purchased as needed.

Dynamic Report Generation Options

American Society of Home Inspectors® provides the most rigid home inspection policies and standards on home inspections, thus, the owner closely followed ASHI's lead (American Society of Home Inspectors, Inc.®, 2004). The owner preferred to follow the home inspection industry leader in order to provide a better home inspection report for the customer and to avoid potential legal issues. All home inspections were documented using the preprinted AHIT® Home Inspection Report which cost $12.50 each (Newcomer, 2003). These blank carbonless reports contain several built-in pages to provide copies to all interested parties: paying client, home inspector and the realtor. The first page contained the Inspection Contract which ensures that the paying client has read and understood the AHSI Code of Ethics and Standards of Practices (American Society of Home Inspectors, Inc.®, 2004), financial liability of the inspector, fee payment, time and date of the inspection, and the property address. The PDA

home inspection application would need to print out the contract, receipt and Code of Ethics and Standards of Practice documents before the inspection could start. Then the final inspection report would be generated and printed after the inspection.

The owner of Feel Good Home Inspections LLC planned on creating his own contract and receipt forms and inserting them into the AHIT® Code of Ethics and Stands of Practices Portable Document Format (PDF) file. To view and print the high quality PDF document would require the installation of the free Adobe® Reader® for Pocket PC 2.0 (Adobe Systems Incorporated [2004]; *Adobe Reader* [2005]). Surprisingly, Adobe® Acrobat® and third party vendors do not offer a PDF creator application for the PDA operating systems. This could be a sign that the PDA is not powerful enough to handle PDF conversions. This means that a created final inspection report could not be converted to a PDF on the PDA. The customer originally wanted this option to provide a locked PDF home inspection report to the realtors and customer. The system requirements were changed to have the conversion done on a separate computer away from the inspection site. The recommendation was to use the more affordable pdfFactory Pro PDF converter for just $99.95 (FinePrint Software, 2005).

The only possible third party PDA report application found was Report CE® by SYWARE® Inc. which typically builds reports from Microsoft® Visual CE® or Microsoft® SQL Server CE database (*Report CE®*, 2005). Microsoft® database licensing fees were over ten times the price of iAnywhere®'s fees, thus, Microsoft®'s databases were not considered for the home inspection application. Data from the iAnywhere® UltraLite™ database could be formatted and imported into Report CE®'s report forms. The downfall was that the static report forms did not have the flexibility to generate a dynamic report.

Other professional home inspection PDA COTS products could only provide static reports or very crude and limited dynamic reports. The requirement was to produce a lively report using different font sizes, text color, logo graphics, horizontal bars, and photos. Microsoft® .NET Compact Framework 2.0 did not come with a powerful set of classes for creating rich text documents. Microsoft® Pocket Word was very weak and somewhat useless compared to the more powerful Microsoft® Office Word application. Dynamically creating or modifying Word documents on the PDA were not possible solutions.

The decision was made to create a simple set of classes to create Hypertext Markup Language (HTML) formatted reports. The report could then be viewed and printed from the NetFront™ v3.3 for Pocket PC Internet browser since this application supported several Internet technologies like HTML 4.01,Cascading Style Sheets (CSS) CSS1 and partial CSS2, DHTML, JV-Lite™ 2, Adobe® Reader ® and Macromedia® Flash™ (*NetFront™ Browser v3.3 for Pocket PC*, 2006). Microsoft's Mobile® Internet Explorer offered very little support for the higher Internet technologies, so the $29.80 NetFront™ v3.3 would be set as the default Internet browser (*NetFront v3.2 with JV-Lite2 Rel1.10*, 2006). Dynamically creating HTML formatted reports should be a good alternative until the PDA processing and software technologies matures.

Summary of Known and Unknown Project Topics

The author did not have any previous experience with conducting a professional home inspection, so the home inspection profession had to be investigated thoroughly in order to provide enough insight to design the home inspection application. It was impossible and unrealistic for the author to become a professional home inspector in the short amount of time available. The owner was held responsible for providing home inspection details as needed.

The personal digital assistant was selected as the portable device by the owner. Neither the owner nor the author had any experience with, or knowledge about, PDAs. Additional research was needed to learn how to use the PDA and more about the PDA hardware and operating system options.

The author had previous experience programming with object-oriented C++ and Java™ languages with various IDEs. Unfortunately, all developed applications were made to run on either a server or a personal computer and not a PDA.

Project Contribution to Field

Designing an object-oriented home inspection application for the PDA researched the latest available technologies available today. Recent technological advancements in handheld devices have radically improved programming options and features. This project addressed the design decisions made to create a C++ object-oriented application for a PDA running the Microsoft® Windows® Mobile™ 5.0 operating system. The project investigated the choices of object-oriented languages, database storage systems, and report generation technologies available for the Microsoft® Windows® Mobile™ 5.0 OS based PDA.

Chapter Three: Project Methodology

Research Methods Used

The management approval board consisted of the author and the owner of Feel Good Home Inspections LLC. The software author was solely responsible for ensuring that the project remained on track since the owner was not familiar with managing a software project. Time management was regulated according to the project task plan. Each of the three SDLC phases was broken down into separate tasks, and then specific start dates and duration days were assigned to each task. Changes to the task plan were only allowed with the approval of the management board. The task plan was referred to on a daily basis to ensure that the project remained on track.

Software Development Life Cycle Model

SDLC model phases were used to manage the project's milestones and deliverables. The scope of the professional project followed the first three Software Development Life Cycle (SDLC) model phases: Requirement Analysis, Preliminary Design, and the Design Phase. The remaining four SDLC phases are to be done outside of this project. These include the Coding, Testing, Implementation, and Maintenance Phases. These seven SDLC phases were chosen from the author's previous experiences with a SDLC model. Each of the seven SDLC phases represented the state of the software project.

Phase I: Requirement Analysis

The main goal of this phase was to discover the owner's requirements for the home inspection application and record those requirements in the "Requirements Definition" document. The system requirements were discovered and documented into functional and non-functional requirements. The PDA home inspection system diagram was included to define the

hardware and network. The project Task Plan was included within this document to officially post the project's three SDLC phases and their tasks. The direction of the project was heavily reliant on the information within this document. The "Requirements Definition" document was initially created in the Requirement Analysis phase; however, it could be modified with proper approval during the Preliminary Design or the Design Phases. Management approval of the "Requirements Definition" document was required before the project continued to Phase II.

Phase II: Preliminary Design

The outcome of the Preliminary Design Phase was the "Preliminary Design" report. This document defined the hardware specification; object-oriented programming (OOP) language, programming and database IDEs, third party COTS software, and the project expenses. The Project Task Plan was updated as needed at the end of this phase. Management approval of the "Preliminary Design" report signified the completion of the Preliminary Design Phase.

Phase III: Design

The SDLC Design Phase output was the "Final Feasibility Design" document. The contents of this document incorporated the following UML™ and database design aides: Use Case System Diagram, Use Case Scenarios, Sequence Diagrams, Static Class Diagrams and the Database Entity-Relationship Diagram. These software design aids are to be used by the author in the SDLC Coding Phase. Management approved the "Final Feasibility Design" document and authorized the SDLC Coding Phase to start in the future. The end of the Design Phase signified the end of the professional project.

Specific Procedures

SDLC Phase Approval

A phase completion could only be authorized by the management board during the SDLC software review meeting. Supporting phase deliverables were presented at each phase meeting in order to get final phase approval from management. The author created and was responsible for maintaining the deliverables. He also presented and explained the deliverables to the owner. Achieving management approval ensured that all research or design documentation has met the final goals for that particular phase. The management board could reject the proposal to complete the phase and then document the reasons for the rejection. Each reason for rejection would then be addressed before the next Management Approval meeting. The second portion of the management approval form decided whether the project would be allowed to start the next SDLC phase. The phase deliverables could receive approval but starting the next phase could be rejected due to documented reasons. This option allows the project to be stopped or suspended due to feasible analysis reasons. Each of the three project phases received management approval to end the phase without any issues or concerns and to start the next phase.

Maintaining "Requirements Definition" Document

The "Requirements Definition" document was the final goal output from the SDLC Requirement Analysis Phase. The document incorporated the project's background, scope, functional and nonfunctional requirements, task plan and the system architecture layout. Changes to this document were allowed with the approval from the management board; however, detailed changes had to be documented in the revision block section of the document.

Development of UML™ & ERD Graphical Design Aids

The development of the UML™ and database graphical design aids in the SDLC Design phase required several iterations. This was required in order to properly document new ideas and requirements between the software and the database. The first design aid generated was the Use Case System diagram that identified all actors and major system tasks. The Use Case Scenarios were then developed to describe the process of conducting an inspection by using the PDA home inspection application. The ERD diagram was then generated to document the required data to be stored. Detailed information that needed to be captured during a home inspection job was documented in the diagram. The ERD went through several updates in order to identify the required inspection information. This diagram was then used to create the general overview of the Static Class diagrams. A few iterations were required before the Static Class overview diagrams solidified. These diagrams were then used to generate the Sequence Diagrams and the Detailed Static Class Diagrams. Several iterations were required to create a good correlation between the UML™ and the ERD diagrams. A color code scheme was applied to the ERD, Use Case System, Static Class and Sequence Diagrams in order to improve the searching and readability of the graphical design aids (see Appendix A). For example, items relating to the person object were colored light blue in each of the graphical design aids. This was a tremendous help due to the shear volume of graphical design aids that were generated.

Format of Deliverables

The three project deliverables included the “Requirements Definition” document, “Preliminary Design” report and the “Final Feasibility Design” document. Each of these Microsoft® Office Word 2003 documents contained several itemized deliverables and were under the revision control of the author. Refer to Section 3.5 for more detailed information about

the deliverables. UML™ graphical design aids and the ERD were created with Microsoft® Office Visio® Professional 2003.

Phase Deliverables

The professional project deliverables were to provide coding direction to the author in the SDLC Coding Phase. All project deliverables were incorporated into the corresponding SDLC phase documents.

Phase I: Requirement Analysis

- "Requirements Definition" document - *Completed*
    a. Functional Requirements and Constraints
    b. Non-functional Requirements and Constraints
    c. System Architecture Layout
    d. Project Task Plan

Phase II: Preliminary Design

- "Preliminary Design" Report - *Completed*
    a. Portable Device Requirements
    b. System Hardware Requirements
    c. Development Software
    d. Commercial-off-the-shelf Software
    e. System Expenses

Phase III: Design

- "Final Feasibility Design" Document - *Completed*
    a. Use Case System Diagram
    b. Use Case Scenario

c. Sequence Diagrams

d. Entity-Relationship Diagram

e. Static Class Diagrams

Resource Requirements

The most important resource requirement was the cooperation and assistance of Terry Ryan, the owner of Feel Good Home Inspections LLC. His vast home inspection knowledge was frequently tested throughout the design process and would be needed to complete the remaining SDLC phases. The three deliverable documents were created with Microsoft® Office Word and Visio® on the author's workstation.

The "Preliminary Design" report identified several resources needed in order to complete the remaining SDLC phases. Software development would need $627 to purchase licenses for Microsoft® Visual Studio 2005 .NET Visual C++ Express Edition IDE, SQL Anywhere® Studio, NetFront™ v3.3 Internet browser, pdfFactory Pro, Belkin Pocket PC OS Printing Utility and Resco Multimedia Suite for PDA. Approximately $1,100 would be needed to purchase the development Dell™ Axim X51v 624MHz PDA with full warranties, SanDisk® 1 GB Extreme II SD™ card, SanDisk® 1 GB Extreme® III CompactFlash® card and a Belkin Bluetooth® wireless USB printer adapter.

Outcomes

Several project management and design methods were produced as project outcomes. The most important method defined was how the SDLC phases were used to manage the project along with the project task plan. Each phase had deliverables to produce and get authorization from the management board. The SDLC phase review, completion and authorization to continue to the next phase were outlined. The rules for maintaining the "Requirements Definition"

document were defined. Lastly, the procedure used to develop the UML™ & ERD graphical aids were explained.

Summary

The Requirement Analysis, Preliminary Design, and Design SDLC Phases were implemented to design an object-oriented home inspection PDA application. The management authorization board insured that all phase deliverables were completed on time. Project time management was controlled by referring to the task plan on a daily bases to ensure the project was kept on schedule. The "Requirements Definition" document was maintained by the management board to ensure the project's requirements and task plan were accurate. Identified technical solutions were provided as deliverables within the "Preliminary Design" report. The acceptance of the "Final Feasibility Design" document was the last project deliverable and signified the end of the professional project.

Chapter Four: Project History

Project Beginning

The professional project was to design an object-oriented home inspection PDA application. The author only had a limited amount of time in order to research the home inspection profession and identify the system requirements. In-depth research on the PDA technology was required for the author to get up to speed with this type of portable device. The technological PDA findings were documented in the "Preliminary Design" report during the SDLC Preliminary Design Phase. Major PDA discoveries included the hardware features, operating system, supported OOP languages and IDEs, relational database, and automatic report generation. These recent technologies were used to design the home inspection PDA application. The latest PDA operating system and C++ programming IDE provided a more efficient method for developing PDA applications with database support. Once the PDA specifications were defined, the standard object-oriented and database graphical design aids were created during the SDLC Design Phase.

Project Management

The software design project was managed using the first three SDLC phases: Requirement Analysis, Preliminary Design, and Design. All three phases were started and completed on time without any issues. The management board also approved the continuation to the next phases, with the exception of the Design Phase, which was put on pause. Both parties agreed to wait with the Coding Phase until the author had earned his Masters degree.

The following SDLC phase's project timelines are listed below.

- Requirement Analysis Phase (November 21, 2005 to November 22, 2005)
- Preliminary Design Phase (November 23, 2005 to December 23, 2005)

- Design Phase (January 3, 2006 to February 21, 2006)

Some of the project tasks within SDLC Preliminary Design Analysis Phase did not follow the specified allotted time. The author did not feel that modifying the individual task duration times was necessary since some tasks took less time while others took a little bit longer. The unit of time duration was days, so that provided a cushion of working longer hours during the day. Some tasks like researching the home inspection profession and identifying the PDA took less time than expected. This extra time was used to start the next task earlier. Investigating PDA database solutions took longer than the expected six days. There were several issues that limited the selection of the data storage system. Several different research paths were taken in order to find an affordable, compatible, reliable and fast database system. A total of eight very long days were spent before identifying the iAnywhere® UltraLite™ RDMS. The author recognized that there were some PDA database technology issues early on with the research and immediately increased the working day hours. The compromise in hours resulted with the successful completion of the SDLC Preliminary Design Phase according to the predetermined schedule.

The “Requirements Definition” document was the master document used to manage the project. The detailed documented requirements removed any assumptions between the owner and the author. This document was referred to frequently to ensure that the project requirements were being met. The official project task plan was incorporated in this document and referenced on a daily basis.

Significant Project Events / Milestones

"Requirement Definition" Document

The deliverable output from the Requirement Analysis Phase was the "Requirements Definition" document. The requirements were quickly captured over the allotted two day period. On the first day, the author conducted an in-person interview with the owner. A set of typed questions were provided to the owner three hours before the scheduled interview meeting. This provided the owner with some time to think about the questions and come up with some solid answers. The questions were grouped in sections about the company, home inspection reports, handheld devices, project planning and costs. Both parties had a good understanding of each others project interests and concerns. To the author's surprise, the owner was insistent on having the PDA as the handheld device. The author was thinking the PC Tablet would be a better solution. The smaller physical size and lighter weight of the PDA was more attractive to the owner. The one hour and twenty-two minute in-person interview was very successful in getting the project started. The author typed up and distributed the meeting notes after the meeting.

The in-person PDA home inspection application brainstorming meeting occurred the following day. The main goal of the meeting was to define specific system requirements. The one hour and ten minute meeting provided additional requirements on the PDA configuration, application and defined the inspection process. That day the author compiled the information from the two meetings and created the "Requirements Definition" document. The author used this document to manage project progress and to ensure that the application analysis and design met the predefined requirements. It was discovered that the requirements needed to be modified through out the SDLC phases. These modifications would be needed for better requirements clarifications or PDA technology limitations. Developing the document also taught the system

owner about capturing requirements for a software project and made him carefully evaluate the financial restrictions. Creating, reviewing and authorizing the "Requirements Definition" document within two days provided a quick management oversight role to the owner. He clearly understood the SDLC phases and the purpose of each of the three project deliverable documents.

"Preliminary Design" Report

The "Preliminary Design" report provided the technological solutions that needed to be used to develop the PDA home inspection application. In a way, the document was an extension to the "Requirements Definition" document. The "Preliminary Design" provided the technological requirements for developing the application. Each defined technology solution was the result of the research and analysis for that particular topic. The financial costs for the developer's and owner's systems were summarized at the end of the document. This provided enough data for both parties to make a feasible financial decision. Essentially, the report documents the restrictions on the future technological project development. The management authorization board reviewed the presented material and decided that implementing these technologies would result in success. The owner felt more at ease knowing exactly what the project costs would end up being. The "Preliminary Design" report contents also revealed the complexity that it would take to design and program the PDA home inspection application.

"Final Feasibility Design" Document

The "Final Feasibility Design" document contained UML™ and ERD design aids that would be frequently referenced by the author in the SDLC Coding Phase. Some of these aids provide a detailed base structure for developing the C++ classes and the relational database, while the other aids focus more on the home inspection actors and processes. The author was surprised by how much the Use Case Scenarios were really dependent on the owner's method of

inspections. Frequent impromptu conversations were needed to create the scenarios. The owner became very comfortable with the Use Cases and the ERD but felt that the Static Class Diagrams were somewhat over his head. The author then spent additional time teaching the owner about the construction and the purpose of the Static Class Diagrams and how they related to the other UML™ design aids. The owner did, however, understand that the "Final Feasibility Design" document was the "blueprint" for creating the application.

Project Plan Changes

Five requirements were added, removed or changed due to discoveries during the Preliminary Design Phase. The significant change occurred on December 13, 2005. The non-functional requirement "Allow Feel Good Home Inspection LLC future growth and add additional home inspections with no software license fees" was changed to "Allow Feel Good Home Inspection LLC future growth by limiting per inspector software license fees to under $300." The owner created the original requirement based on the current COTS expensive yearly home inspection PDA fees. One of the main purposes of the project was to avoid paying a third party home inspection PDA vendor hundreds of dollars for one PDA license. This requirement was rewritten to allow purchasing legal PDA software licenses needed to create one PDA home inspection system. The total PDA software licensing cost ended up being $199 which was well under the owner's $300 feasible limit.

Three requirements were added during the Preliminary Design Phase. On December 14, 2005 the following functional PDA contract and receipt requirements were added: "The PDA application should print out the ASHI® Code of Ethics.", "Automatically fill-in blank contract form information.", and "Automatically fill-in blank receipt form information." These requirements were discovered while discussing the home inspection process.

The last requirement modification involved the removal of the functional final report requirement “The PDA application should be able to convert the completed report to a PDF format file.” This requirement was eliminated since there was no available COTS PDF converter that could run the PDA (refer to section 2.2.4 – Dynamic Report Generation Options). Developing a PDA PDF file converter application was out of the project scope.

Project Goal Evaluation

The owner and the author evaluated the project goals during the SDLC software review meetings. The Requirement Analysis Phase goal was to produce the “Requirements Definition” document. This document was thoroughly reviewed to ensure that all of the home inspection application requirements were captured. The identified requirements provided the baseline for the design goals. Requirements, additions or changes to the “Requirements Definition” document would be allowed during the remaining SDLC phases. This document was frequently referenced throughout the project and provided the necessary requirement guidance and restrictions.

The Preliminary Design Phase resulted in the completion of the “Preliminary Design” report. This report provided the details on which technologies to implement, PDA specifications, third-party software applications, and cost analysis. The technologies chosen were explained to the owner. He also reviewed the projected cost analysis and approved the findings within the “Preliminary Design” report.

The final home inspection design goal was achieved by completing the “Final Feasibility Design” document. This document would guide the author through the SDLC Coding phase. Object-oriented design aides were incorporated into this document to guide the C++, database and GUI layout development. The evaluation of the “Final Feasibility Design” document was

done by verifying that the requirements in the “Requirements Definition” document were met. Both the owner and author approved the contents of the “Final Feasibility Design” document.

What Went Right / Wrong

The owner selected the PDA as the portable device which surprised the author; he was thinking either the tablet-PC or laptop would have been chosen as the portable device. The small and very portable size of the PDA ended up being the more logical device for the home inspection application. The home inspector needed to have a small, lightweight device to carry around on the jobsite for several hours. The author was unfamiliar with programming or even operating a PDA. These issues were overcome with additional research on the PDA and programming options. Several questions arose that needed to be answered in order to better understand programming the PDA.

The author also had to overcome his lack of knowledge regarding the home inspection profession. It was not feasible for the author to take several home inspection courses and become a certified professional, which would take over a year and cost thousands of dollars. To get around this barrier, the owner became the professional advisor and the author was required to learn some of the basic home inspection procedures and documentation. This arrangement worked out well for both parties. The author knew about the general home inspection process and the owner provided additional detailed information as needed.

Designing the home inspection application for a PDA relied heavily upon the communication between the owner and the author. Communication between the two was very good which was directly related to the availability of both parties. The only time that communications were not possible was due to holidays and vacations which were

understandable. In-person and informal meetings resulted in the best form of communication since both parties could present supporting information.

The only downfall of the project was the lack of a COTS PDF file converter for the PDA. Apparently, the PDA does not have enough processing power to convert a text or HTML file into a PDF file. For this reason, the requirement for creating a PDF file on the PDA was eliminated from the "Requirements Definition" document. The processing power limitations will be overcome as the PDA processor technology advances.

Impact of Project Variables

By far, the largest project variable was selecting the PDA as the portable device. This was the most significant project variable since it directly steered the project's direction. The optional programmable devices included the laptop, tablet-PC, PDA, or the mini-computer. The owner selected and supported his reasons why he went with the PDA on the first day of the Requirement Analysis Phase.

The second big impact project variable was choosing the Microsoft® Visual Studio 2005 .NET C++ IDE to program the PDA running the latest Microsoft® Mobile™ 5.0 operating system. The newly released IDE by Microsoft® specifically targeted the PDA processors. The new IDE provided the object-oriented C++ requirement and at the same time identified the PDA operating system to be the Microsoft® Mobile™ 5.0.

Identifying the data storage system was the most troublesome milestone to achieve that had a huge impact upon the project. The lower PDA processing power provided very little third party COTS PDA database solutions. Most PDA applications relied upon the troublesome and tedious file database systems which the author wanted to avoid for speed and reliability reasons. The discovery of the UltraLite™ database by iAnywhere® had a huge impact on the project.

This database has a small footprint for space limited devices like the PDA and a very powerful database engine similar to large scale databases. The UltraLite™ database was developed to share information between the client PDA and server side databases. The UltraLite™ database will be able to extract and store home inspection data reliably and quickly.

Findings/Analysis Results

PDA Limitations and Programming Issues

Programming a PDA required a different perspective than programming a personal computer. The main differences between these electronic devices stem from the processing power, memory management, and screen size. Powerful personal computers provided the ideal situation of fast and powerful processing, large monitor screen size, and a much larger amount of memory and disk space, while the PDA was much more limited in all of these areas and did not have a hard drive. Data storage and applications were instead stored on portable Flash® cards which are currently limited to 2GB of memory space. PDA requires more programming memory awareness to insure that there is no overflow of the allotted space and to write more efficient code since the processing speed is considerably slower. More responsibility would fall upon the author to monitor the application memory usage and processing speed during development.

Another unusual programming factor is the small size of the PDA touch screen. Both the author and owner would have to experiment with GUI screen layouts in order to find the best method to collect and display large amount of home inspection information. Neither of them was familiar with running applications on the PDA, so some GUI screen experimentation would be required during the SDLC Coding Phase.

Both the speed and data storage limitations were particularly disturbing to the author since he was use to programming applications for computers and servers. The lower PDA processors

speeds resulted in less power from the battery, so the author used this statement in determining which PDA and hardware to use on the project. The owner stated that it took up to four hours to inspect a very large or complex house using the blank reports. The author hoped that the PDA home inspection would reduce the inspection time by at least 25%, so that meant that the previous inspection should only take three hours to conduct. The worst case scenario was set at three hours plus a 25% allowance for the aging battery reduction in power storage. The new worst case four hour PDA runtime was used as a reference to purchase the fastest PDA on the market. Research results found that the Dell™ Axim™ with the 2200mAh Lithium-Ion rechargeable extended battery could provide seven hours of worst case scenario runtime (Brown, 2005). The results were very attractive since the seven hours of runtime could actually support two average sized house inspections without recharging. In reality, the owner planned on recharging the PDA while driving to the next job site or replacing the battery with a fully charged one before the inspection. The Dell™ Axim™ 624MHz PDA offered the maximum RAM and ROM sizes, physically larger and higher resolution screen, SD™ and CompactFlash™ memory card slots, and unprecedented three year limited and accidental warranty. Being able to purchase long term warranties were emphasized by the owner and were put in to the system requirements.

Windows® Microsoft® Mobile™ 5.0 OS changed the previous memory management methods in order to conserve battery energy and to provide non-volatile memory for applications and data. These changes allowed support for the increased non-volatile PDA ROM size to 256MB (196MB usable). This would be the safe storage place for the PDA home inspection application and the RDMS.

The SD™ memory port was reserved for transferring digital photos from the owner's camera. These photos would be able to be transferred to the PDA by inserting the camera's memory card in the PDA's SD™ memory port and using the Resco Explorer v5.38 COTS application. Microsoft®'s PDA Explorer had several poor reviews and the Resco Explorer 2005 provided several good PDA features for networking, file management registry modifications and photo viewing (Goldstein, [2005]; *Resco Media Suite (v5.38)*, [2005]).

The CompactFlash™ port was reserved to store data files. The very fast SanDisk® 1GB Extreme® III CompactFlash® card would resemble a large hard drive for the PDA. This would be the storage place for the generated final inspection reports, photos and other documents. The 1GB size should be adequate for storing the business files; however, the CompactFlash™ card could be replaced in the future to provide more storage space.

The owner planned on using his business printer with the Belkin Bluetooth® wireless USB printer adapter. The PDA would be able to use the Bluetooth® wireless communication to send print jobs to the printer. An automotive power converter would provide the AC power to the printer.

Programming the PDA

The immaturity of third party JVMs for the new Microsoft® Mobile™ 5.0 OS was too big of a risk to choose Java™ as the object-oriented language. The JVM search was conducted in December, 2005 and no final release of JVMs could be found. Microsoft® Mobile™ 5.0 OS included major changes with the operation of the PDA hardware and memory usage. These major changes apparently required several changes within the JVM. Another concern was that the third party vendors appeared to have their own flavor of the Java™ 2 Micro Edition (J2ME).

The author was familiar with the Java™ 2 Standard Edition (J2SE) classes and wished to use the J2ME subset for programming the PDA.

The logical decision was to choose Microsoft® Visual Studio 2005 .NET C++ as the object-oriented language solution for the PDA home inspection application. Microsoft® standardized the handheld device platform with the Microsoft® Mobile™ 5.0 operating system. Then the .NET Compact Framework 2.0 was incorporated into Visual Studio 2005 .NET to provide a common programming IDE for various types of handheld devices. Developing PDA screens was simplified with the WYSIWYG designer view. The new graphical designer allowed the development of the PDA screens to be done in either portrait or landscape mode. Screen control tools could be setup to access data from a database. Visual Studio has emerged as a single source IDE which attracts more programmers. Another project benefit was that the author was already familiar with Microsoft® Visual C++ IDE.

PDA Database

Intensive research for PDA data storage resulted in choosing the iAnywhere® Solutions UltraLite™ relational database to store the home inspection data. The findings were a big relief after investigating the PDA operating systems, C++ Framework, and PDA processing speed limitations. Both processing speed and volatile data issues steered the decision to go with the RDMS.

Visual Studio 2005 .NET Compact Framework 2.0 included limited XML classes for incorporating XML technology onto a PDA; however, there were several negatives for using the XML based file database system. The most critical evaluation stems from the long time that it takes for the PDA to open, read, write or modify information with the XML file. The slower PDA processor speeds naturally would delay the opening and closing of several XML files

during the inspection process. This brought up another issue requiring the XML processing classes to be either created or purchased from a third party vendor. Changes to the database design would require more coding changes to the XML structure and in the application. Another XML issue was unexpected data loss in PDA memory from the PDA OS locking up or power loss while the XML data file was open in volatile RAM. Data loss would be highly undesirable and would require the inspector to re-enter the lost data. The last downfall came from the fact that XML was a non-relational file database system. Implementing a RDMS would help with the more complex home inspection ERD design. Using ANSI SQL commands to manipulate the data within the RDMS would help with the complexity and provide faster data storage.

Implementing the powerful, but small footprint, iAnywhere® UltraLite™ relational database would provide data loss and integrity protection. Changes in the database schema would also limit the ANSI SQL changes needed in the C++ code. Faster database read/write speeds would result from the database services running in the background. Selecting the iAnywhere® UltraLite™ relational database product also promoted future growth for the home inspection application into a client-server product. The included synchronization software would allow data to be transferred between the UltraLite™ client databases to a possible server database. Lastly, the author could benefit from using one graphical database development tool, SQL Anywhere® Studio, to create and deploy the client-server databases onto several different platforms.

Dynamic Final Inspection Report Generation

The Microsoft® Mobile™ 5.0 OS used the improved .NET Compact Framework 2.0 classes; however, there were no advanced features for creating richly formatted text documents. The final inspection report would be generated from a custom HTML classes and displayed on

the PDA with a third-party Internet browser. NetFront™ v3.3 for Pocket PC Internet browser supports several Internet coding technologies similar to the PC Internet browsers, unlike Microsoft®'s Internet browser for the PDA that comes with the Microsoft® Mobile™ 5.0 OS. These advanced Internet technologies would be used to create the colorful formatted text and photographs.

One of the system requirements was to be able to create a locked PDF home inspection report for the realtors and customer. Extensive investigation revealed that there were no PDF software conversion tools available for the PDA, so the system requirement to convert the final report on the PDA was dropped from the PDA home inspection requirements. The recommended suggestion was to convert the HTML formatted final inspection report on a personal computer using the pdfFactory Pro PC application.

Summary of Results

The professional project used the SDLC Requirement Analysis, Preliminary Design, and Design Phases to manage the software development. A project task plan defined the dates and durations for each phase and task. The task plan was used to manage the project tasks to ensure that the project remained on schedule. The management authorization board approved the three phase deliverables and declared the phases completed.

The Requirement Analysis Phase deliverable was the "Requirements Definition" document which provided the project requirements and task plan. Defining the Windows® based PDA as the handheld device sent the project direction in unknown technological territory for the author. Additional research was needed to address the technological direction for PDA application development. The "Requirements Definition" document served was frequently referenced

thorough out the phase to ensure that the application, hardware, software, and financial restrictions were followed.

The deliverable outcome of the Preliminary Design Phase was the "Preliminary Design" report. This report was the technological findings that would be needed to program a PDA running the Microsoft® Mobile™ 5.0 OS. The report provided specific technological project restrictions for designing and developing the PDA home inspection application. The Microsoft® Mobile™ 5.0 OS was selected as the base programming platform. The object-oriented programming language and IDE was identified as Microsoft® Visual Studio 2005 C++. iAnywhere® Solutions UltraLite™ relational database was selected as the powerful PDA database solution. Lastly, the dynamic inspection report generation would be done through custom C++ classes that would create the formatted text HTML document. The report also identified third-party COTS solutions that would provide additional services for the PDA. All of these technological project restrictions would be referenced in the SDLC Design and Coding Phases.

The "Final Feasibility Design" document was the last phase deliverable. The SDLC Design Phase focused on creating the UML™ and database entity-relationship diagram design aids. The UML™ design aids provided different perspectives on the application requirements for the author. Both the application C++ and database design were detailed with the Use Case System diagram, Use Case Scenarios, Sequence diagrams, Static Class diagrams and the ERD. The contents of the "Final Feasibility Design" document would be the used as the coding restrictions in next Coding and Testing Phases.

The research and analysis has provided a programming methodology for a PDA handheld device running the Microsoft® Mobile™ 5.0 OS. Special considerations, such as, PDA

processing speeds and memory management were used in selecting PDA hardware and operating system. The processing speed and data storage limitations also weighed in heavily upon selecting the iAnywhere® Solutions UltraLite™ relational database to store the immense amount of data.

Chapter Five: Lessons Learned and Next Evolution of the Project

Lessons Learned

The most significant lesson learned by both parties was the education each received about the other's specialty. Neither the author nor the owner originally knew anything about the other's specialty. Most of the learning effort fell on the author in gaining a better understanding of the home inspection profession. Learning all of the fine details of the profession was not the goal; however, enough details about the inspection process, data collection and information presentation was needed in order to design the home inspection application. In addition, this was the first time the owner helped design a software application, so additional time was needed for training him on the project design process and the software design aids.

The author also relearned a valuable lesson. He started the project under the assumption that the portable device would be a PC tablet; however, the owner declared that the portable device was to be a PDA. The in-person interview with the owner quickly corrected the author's assumptions. Conducting the in-person interview to capture the system requirements should never be avoided since it can provide invaluable information for the project.

What Should Have Been Done Differently

At least one additional month to learn about the home inspection process, data collection and generating the final reported should have been allotted. The author underestimated the amount of knowledge and the inspection details needed to be able to design a professional home inspection application. Continuous specialty inspection training was needed over many years to become a proficient home inspector, so it was unreasonable to expect the author to learn about the home inspection details in such a short time. A better "hands on" approach to learning would have greatly benefited the author but it was not possible due to scheduling issues. The problem

was that most of the owner's customers were realtors which must be present during the inspection. Realtors try to push the inspectors to work faster since they do not want to be there for three to fours hours. Having the author continually asking questions would slow down the inspection and could compromise the customer's satisfaction.

Deciding to use the PDA as the portable device also put the author in unfamiliar territory. The author did not know how to operate a PDA or how to design and program a PDA application. At least one additional month should have been added to research the PDA operating systems and OOP languages. Instead, the extra time needed was obtained from working late nights and weekends.

Project Expectations

The project goal was to research and analyze object-oriented programming for Windows® based PDAs. The research was based on the requirements for the professional home inspection PDA application. It was found that the latest Microsoft® Microsoft® Mobile™ 5.0 operating system greatly improved the PDA operation and reliability over the previous legacy operating systems. The Mobile™ 5.0 OS was redesigned to support several types of handheld devices, thus, allowing the programming community to become familiar with one OS for all devices. The previous PDA memory management was redesigned in order to store data into non-volatile memory and to support larger RAM and ROM sizes. PDA processing speed performances were increased while at the same time increasing the battery life. Microsoft® Microsoft® Mobile™ 5.0 OS was chosen as the platform baseline due to its significant PDA improvements.

Both Java™ and C++ object-oriented languages were investigated for the Windows® based PDA. Java™ was ruled out since there were no finalized JVM available for the newly

released Microsoft® Mobile™ 5.0 OS. This was the main reason for ruling out Java™ as the OO language for the project. The OO language defaulted to C++.

Microsoft® quickly released Visual Studio 2005 .NET after releasing the new operating system. For the first time, Visual Studio 2005 .NET incorporated handheld programming support. Now programmers could use one IDE to program applications for computers and handheld devices. Microsoft® clearly made it known that they wanted programmers to develop applications for handheld devices by providing one OS platform and a familiar programming IDE. The Visual Studio 2005 .NET C++ IDE was chosen as the tool to generate the OO C++ code for the project. Visual Studio also included handheld simulators to decrease the testing time.

The PDA's speed limitation, compared to a personal computer, was a major factor in selecting the PDA. The expected inspection process was projected to take three hours to complete. The selection of the PDA was based upon a full load runtime of at least seven hours. This would allow two average inspections to be conducted with a fully charged battery. The most powerful and popular Dell™ Axim™ X51v PDA met the battery runtime requirements. This PDA also provided the fastest processing speed, largest RAM and ROM sizes, higher resolution color touch screen, wireless adapters and extended warranties.

The professional home inspection application needed to store a large amount of information. Both XML and relational database solutions were researched and analyzed based upon the PDA memory and speed limitations. Visual Studio 2005 .NET Compact Framework 2.0 included limited XML classes managing the data files and creating or purchasing more advanced classes were found to be outside of the project scope. Another negative for implementing XML on the PDA was with the slow file handling speed. The goal was to reduce

the inspection time, so having the inspector wait for the XML file handling was determined to be unacceptable. The final decision was to go with a relational database for the PDA.

iAnywhere® Solutions UltraLite™ relational database was selected as the database technology for the home inspection project. The very small footprint and powerful UltraLite™ database was specifically made to be deployed onto handheld devices and was supported by the Visual Studio 2005 .NET IDE. Implementing the relational database would provide faster data read/write speeds, data loss prevention and built-in data integrity protection. The author would benefit from the using SQL Anywhere® Studio graphical database development tool since it also supported client-server databases and deployment to several different platforms. This could be very helpful if the professional home inspection application eventually evolves into a client-server application.

Rich text final inspection reports could not be generated due to Microsoft® .NET Compact Framework 2.0's weak rich text class support and Microsoft® Pocket Word's lack of programming interface. The PDA report generation findings resulted in developing HTML C++ classed in order to dynamically generate a highly formatted document. The custom class would also incorporate other higher Internet technologies, like CCS and JV-Lite™ 2, in order to simplify the process of generating a professional looking report. This decision required the purchasing of NetFront™ v3.3 Internet browser since Microsoft's Mobile® Internet Explorer offered very little support for these higher Internet technologies. The requirement for converting the final inspection report into a locked PDF was dropped since there were no COTS PDF conversion applications for the PDA. Developing the PDF converter was determined to be outside of the project scope.

The project goal of designing an object-oriented home inspection application for a portable device was met. The "Final Feasibility Design" incorporated the targeted portable Personal digital assistant device, identified integrated development environment programming and database development applications, third party software applications, and the object-oriented design aids. A lot of detail was put into the OOD design aids: Use Case System Diagram, Use Cases, Sequence diagrams, Static Class Diagrams, Graphical User Interface Flow Diagram, Static Class Diagrams, and the Database Entity-Relationship Diagram. The completion of the "Requirement Definition" and the "Final Feasibility Design" signified the completion of the SLDC Design Phase which was also the end of the professional project.

Next Stage of Project Evolution

The last three phases, Testing, Implementation, and Maintenance SDLC, will be completed outside of the project. The author planned on going through a quick iteration of the first two SDLC phases before continuing with the Coding Phase. A year has passed since the research was conducted, therefore, the technological advances of the handheld devices should be re-evaluated since some high-end handheld gaming devices have emerged into the market. These gaming devices are really miniature computers that are capable of running the 3D graphical PC games. The intent would be to identify some handheld miniature computers in the Requirement Analysis Phase in order to create a new "Requirements Definition" document. The home inspection associations have recently been selling electronic reference books that run on personal computers. These applications could run on the PDA application along with the PDF file converter program. The Preliminary Design Phase would research and analyze the new miniature computers, dynamic report generation and COTS utilities. The analysis findings would then be documented in the "Preliminary Design" report. The original design plan would

still implement the OO C++ language and IDE but implement the iAnywhere® Solutions client database instead of the UltraLite™. The management authorization board would review the miniature computer findings and determine at that time whether to implement the PDA or the miniature computer project.

The three phase deliverable documents are to be used to develop the C++ code and the relational database during the Coding Phase over an eight week period. Another eight weeks would be spent in the Testing Phase. The owner wishes to spend the first two weeks testing the application in an office environment to get familiar with the PDA and the home inspection application, then spend six weeks testing the software application in the field. The first two home inspections will be done on non-paying customer homes to allow the inspector extra time to carefully test the application in the field. The remaining six weeks would be done on paying customer homes in conjunction with completing the handwritten inspection report. One inspector will run the PDA home inspection application while a second inspector fills out the handwritten home inspection report. The completed written report would then be compared to the final home inspection report generated by the application. This comparison would insure that all of the standard inspection items were conducted in addition to the new inspection points added by the owner. All software problems would be resolved during the Testing Phase. The home inspection PDA application would then be implemented over a one week time period in the final SDLC Implementation Phase. The last SDLC maintenance phase would continue until new features are needed, which would cause a quicker iteration through the SDLC phases.

Both the owner and author agreed to wait until six months into the Maintenance Phase before pursuing the marketing of the home inspection application. Any software issues should appear and be resolved during the first six months. After the first four months, both parties

would then focus on getting the home inspection application certified by the American Society of Home Inspectors® organization. Certification by the largest and most respected home inspection organization would provide great credibility to the home inspection PDA application; while at the same time attract certified home inspectors.

Conclusions and Recommendations

The current PDA processing power and application storage has been the most limiting factor for applications. This situation was very evident with the limited capabilities with the Windows® and COTS PDA applications. Even Microsoft® Visual Studio 2005 .NET Compact Framework 2.0 offered very limited classes for developing applications for handheld devices. The advancements of the PDA technologies should be watched closely for significant improvements. An alternative solution would be to implement the emerging gaming miniature computers to host the home inspection application. These devices would resolve the processing and data storage limitations and also be able to run PC applications. The major concern would be with the device's cost, reliability, durability and extended warranties.

The PDA performance and data reliability were significantly improved from the Microsoft® Mobile™ 5.0 OS. The operating system rewrite increased the OS performance, created non-volatile memory for applications and data, allowed support for larger RAM and ROM memory, provided higher resolution touch screen support and provided a much improved synchronization program. The Microsoft® Mobile™ 5.0 OS was also designed to be the base platform for four different types of handheld devices These changes provided a huge upgrade from the previous PDA technology, so the Microsoft® Mobile™ 5.0 OS was chosen as the default platform. Microsoft® answered the handheld programmer communities by incorporating handheld device support into Visual Studio 2005 .NET. The IDE also included a graphical PDA

simulator for quickly testing compiled applications. Visual C++ was chosen as the object-oriented language and iAnywhere® Solutions UltraLite™ relational database was selected as the database technology for the home inspection project. The UltraLite™ database provided a very small footprint size, fast data storage and data integrity. The SQL Anywhere® Studio became the database IDE of choice. This graphical database development tool provided the option of deploying server, personal or lightweight databases for client-server applications. The author would greatly benefit from the compatibility between the OS, programming and database IDEs.

Summary

The professional project required more research on the home inspection profession than originally expected. This, compounded with additional PDA technologies and programming, strained the original task timeline. The project's tasks never fell more than three days behind; however, it required a lot of additional time on behalf of the author. At least one month of additional time should have been allotted for researching the home inspection profession and the PDA technology. All project phase deliverables were delivered on-time and incorporated in the "Requirements Definition" document, "Preliminary Design" report and the "Final Feasibility Design" document.

The high-end database design application was based on selecting the Dell™ Axim™ X51v 624MHz PDA with the Microsoft® Mobile™ 5.0 OS. Object-oriented C++ software development for the PDA would be accomplished with Microsoft® Visual Studio 2005 .NET. The huge amount of inspection data would be managed by the iAnywhere® Solutions UltraLite™ relational database. This database provided several powerful relational database features for the PDA deployment. The SQL Anywhere® Studio could also deploy databases

with synchronization software to support a future professional home inspection client-server application.

Works Cited

Adobe Systems Incorporated. (2004). *Adobe Reader® for Pocket PC*. Retrieved December 14, 2005, from http://www.adobe.com/products/acrobat/pdfs/95004413_PPC_FAQ_V4.pdf

Adobe Reader. (2005). Retrieved December 14, 2005, from http://www.adobe.com/products/acrobat/readerforppc.html

American Society of Home Inspectors. (2005). Retrieved December 14, 2005, from http://www.ashi.org

American Society of Home Inspectors, Inc.®. (June 13, 2004). *The Standards of Practice and Code of Ethics of THE AMERICAN SOCIETY OF HOME INSPECTORS®*. Retrieved December 14, 2005, from http://www.ashi.org/documents/pdf/standards.pdf

Axim™ X51v Details. (2005). Retrieved December 12, 2005, from http://www.dell.com/content/products/productdetails.aspx/axim_x51v?c=us&cs=19&l=en&s=dhs

Benz, Brian, & Durant, John R. (2003). *XML Programming Bible.* New York, NY: Wiley Publishing, Inc. 3-28.

Brown, Adama D. (September 20, 2005). *Review – Dell Axim X51V*. Retrieved December 12, 2005, from http://www.brighthand.com/default.asp?newsID=2670&review=Dell+Axim+X51v

De Herrera, Chris. (September, 2005). Windows Mobile 5.0: A Look Under the Hood. Retrieved December 12, 2005, from http://www.pocketpcmag.com/_archives/sep05/directions.aspx

Dumas, Diane. (November, 2005). New Windows Mobile 5.0 Pocket PCs and Smartphones!. *Smartphone & Pocket PC Buyers Guide 2006*, 13-16.

Extensible Markup Language. (December 11, 2005). Retrieved December 12, 2005, from http://en.wikipedia.org/wiki/Xml

Goldstein, Hal. (November, 2005). Best Software Awards 2005. *Smartphone & Pocket PC Buyers Guide 2006*, 59.

Hall, Rich. (November, 2005). What is Windows Mobile?. *Smartphone & Pocket PC Buyers Guide 2006*, 6-12.

iAnywhere® Solutions, Inc. (2004). *SQL Anywhere® Studio*. Retrieved December 13, 2005, from http://www.ianywhere.com/downloads/datasheets/sql9datasheet.pdf

iAnywhere® Solutions, Inc. (2003). *SQL ANYWHERE® STUDIO AND MICROSOFT .NET SUPPORT AT A GLANCE*. Retrieved December 13, 2005, from http://www.ianywhere.com/downloads/datasheets/sql_net.pdf

Jones, A. Russell. (November 7, 2005). *DevX.com*. Official Microsoft Visual Studio/SQL Server/BizTalk Launch Day Has Arrived. Retrieved December 2005, from http://www.devx.com/dotnet/Article/29695

NetFront™ Browser v3.3 for Pocket PC. (2006). Retrieved January 3, 2006, from http://www.access-company.com/products/netfrontmobile/browser/32_wm.html

NetFront v3.2 with JV-Lite2 Rel1.10. (2006). Retrieved January 3, 2006, from http://www.pocketgear.com/software_detail.asp?id=18366

Newcomer, Roy & Newcomer Scott. (2003). *Home Inspection Report*. Retrieved December 14, 2005, from http://www.ahit.com/products/reportforms/Reportwboxes.pdf

PDA market swells, Windows CE holds lead. (August 2, 2005). Retrieved December 12, 2005, from http://www.windowsfordevices.com/news/NS2332906960.html

FinePrint Software, LLC. (2005). *pdfFactory*. Retrieved December 4, 2005, from http://www.pdffactory.com/products/pdffactory

Report CE® 4. (2005). Retrieved December 14, 2005, from http://www.syware.com/products/reports_ce.php

Report: Windows CE top PDA OS in 2004. (February 15, 2005). Retrieved December 12, 2005, from http://www.windowsfordevices.com/news/NS5200181621.html

Resco Media Suite (v5.38). (December 15, 2005). Retrieved December 15, 2005, from http://resco.net/pocketpc/mediasuite/default.asp

Richter, Charles. (1999). *Designing Flexible Object-Oriented Systems with UML*. Indianapolis, Indiana: Macmillan Technical Publishing.

SQL Anywhere – Broad Platform Support. (2005) Retrieved December 13, 2005, from http://www.ianywhere.com/products/supported_platforms_overview.html

SQL Anywhere Studio – UltraLite Deployment Technology. (2005). Retrieved December 13, 2005, from http://www.ianywhere.com/products/ultralite.html

Struys, Maarten. (November 2005). *MSDN Library*. Using Visual Studio 2005 to Design User Interfaces and Data for Device Applications. Retrieved December 2005, from http://msdn2.microsoft.com/en-gb/library/aa454904.aspx

SYBASE® eShop SQL Anywhere Studio. (2005). Retrieved December 13, 2005, from http://eshop.sybase.com/eshop/buy_detail?id=72250&prodId=7446&bid=19160

SYBASE® eShop UtraLite. (2005). Retrieved December 13, 2005, from http://eshop.sybase.com/eshop/buy?p=SQL+Anywhere+UltraLite+with+Sync&v=9.0&l=English&t=Windows+x86&resubmit=Select&b=144&s=Software&id=19160

UltraLite. (2005). Retrieved December 13, 2005, from http://www.ianywhere.com/products/ultralite.html

Wildermuth, Shawn. (January 9, 2005). *Choosing a Database for Compact Framework Applications*. Retrieved December 12, 2005, from http://www.devsource.com/print_article2/0,1217,a=142316,00.asp

Appendix A

Example Color Coding Scheme