HRI Rental Property Manager

By

Erica Hunter

Submitted to
the Faculty of the Information Technology Program
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology

University of Cincinnati
College of Applied Science

May 2007
HRI Rental Property Manager

by

Erica Hunter

Submitted to
the Faculty of the Information Technology Program
in Partial Fulfillment of the Requirements
for
the Degree of Bachelor of Science
in Information Technology

© Copyright 2007 Erica Hunter

The contents of this document are under copyright of the author. It may not be reproduced and distributed in whole or in part without the written permission of the author.

___________________________________________________ __________________
Erica Hunter        Date

___________________________________________________ __________________
Robert Schlemmer, Faculty Advisor     Date

___________________________________________________ __________________
Hazem Said, Ph.D. Department Head     Date
Acknowledgements

I would like to give special thanks to Professor Robert Schlemmer, Associate Professor and Technical Advisor, and Sean Winfield, CAGIS Sr. Computer Analyst, for supporting, advising and assisting with the development of this project. I would like to give special thanks to Stephen Hunter Jr., owner of Hunter Realty Investments, LLC, for challenging me and allowing me to be creative with the design concept of this project. Finally, I would also like to give special thanks to my loving parents, Stephen and Charlotte Hunter, for believing and supporting me throughout all my endeavors in my entire life.
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acknowledgements</td>
<td>i</td>
</tr>
<tr>
<td>Table of Contents</td>
<td>ii</td>
</tr>
<tr>
<td>List of Illustrations</td>
<td>iii</td>
</tr>
<tr>
<td>Abstract</td>
<td>iv</td>
</tr>
<tr>
<td>1. Statement of the Problem</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Definition of the Need</td>
<td></td>
</tr>
<tr>
<td>2. Description of the Solution</td>
<td>5</td>
</tr>
<tr>
<td>2.1 User Profile</td>
<td>5</td>
</tr>
<tr>
<td>2.1 Design Protocols</td>
<td>6</td>
</tr>
<tr>
<td>3. Deliverables</td>
<td>10</td>
</tr>
<tr>
<td>4. Design and Development</td>
<td>10</td>
</tr>
<tr>
<td>4.1 Budget</td>
<td>11</td>
</tr>
<tr>
<td>4.2 Timeline</td>
<td>11</td>
</tr>
<tr>
<td>4.2.1 Senior Design I Accomplishments</td>
<td>12</td>
</tr>
<tr>
<td>4.2.2 Senior Design II Accomplishments</td>
<td>13</td>
</tr>
<tr>
<td>4.2.3 Senior Design III Accomplishments</td>
<td>13</td>
</tr>
<tr>
<td>4.3 Software</td>
<td>14</td>
</tr>
<tr>
<td>4.4 Hardware</td>
<td>14</td>
</tr>
<tr>
<td>5. Proof of Design</td>
<td>14</td>
</tr>
<tr>
<td>5.1 User Interface Design</td>
<td>15</td>
</tr>
<tr>
<td>5.1.1 Main Form</td>
<td>15</td>
</tr>
<tr>
<td>5.1.2 Property Form</td>
<td>16</td>
</tr>
<tr>
<td>5.1.3 Tenant Form</td>
<td>19</td>
</tr>
<tr>
<td>5.1.4 Contractor Form</td>
<td>20</td>
</tr>
<tr>
<td>5.2 Work Orders</td>
<td>21</td>
</tr>
<tr>
<td>5.3 Expenses</td>
<td>22</td>
</tr>
<tr>
<td>5.4 Rental Income</td>
<td>23</td>
</tr>
<tr>
<td>5.5 Reporting Functionality</td>
<td>25</td>
</tr>
<tr>
<td>5.6 Data Repository</td>
<td>32</td>
</tr>
<tr>
<td>5.7 Add and Edit Functionality</td>
<td>32</td>
</tr>
<tr>
<td>6. Testing Plan</td>
<td>33</td>
</tr>
<tr>
<td>7. Conclusion and Recommendation</td>
<td>33</td>
</tr>
<tr>
<td>7.1 Conclusion</td>
<td>33</td>
</tr>
<tr>
<td>7.2 Recommendation</td>
<td>33</td>
</tr>
</tbody>
</table>
# List of Figures

<table>
<thead>
<tr>
<th>Figure Number</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1: Software Cost Comparisons</td>
<td>1</td>
</tr>
<tr>
<td>Figure 2: Schedule E 1040 Tax Form</td>
<td>4</td>
</tr>
<tr>
<td>Figure 3: HRI Rental Property Manager Outline</td>
<td>7</td>
</tr>
<tr>
<td>Figure 4: Database Table Design</td>
<td>8</td>
</tr>
<tr>
<td>Figure 5: Database Queries</td>
<td>9</td>
</tr>
<tr>
<td>Figure 6: Project Budget</td>
<td>11</td>
</tr>
<tr>
<td>Figure 7: Timeline</td>
<td>12</td>
</tr>
<tr>
<td>Figure 8: Main Form</td>
<td>15</td>
</tr>
<tr>
<td>Figure 9: Property Form</td>
<td>16</td>
</tr>
<tr>
<td>Figure 10: Mortgage Form</td>
<td>17</td>
</tr>
<tr>
<td>Figure 11: Record Insert Confirmation</td>
<td>18</td>
</tr>
<tr>
<td>Figure 12: Tenant Form</td>
<td>19</td>
</tr>
<tr>
<td>Figure 13: Contractor Form</td>
<td>20</td>
</tr>
<tr>
<td>Figure 14: Work Orders Form</td>
<td>21</td>
</tr>
<tr>
<td>Figure 15: Expense Payment Form</td>
<td>22</td>
</tr>
<tr>
<td>Figure 16: Expense Viewer</td>
<td>23</td>
</tr>
<tr>
<td>Figure 17: Rent Payment Form</td>
<td>24</td>
</tr>
<tr>
<td>Figure 18: Income Viewer</td>
<td>25</td>
</tr>
<tr>
<td>Figure 19: Reports Menu Option</td>
<td>26</td>
</tr>
</tbody>
</table>
Figure 20: Active Tenants Report  26
Figure 21: Property Report  27
Figure 22: Expense Summary Report  28
Figure 23: Income Summary Report  29
Figure 24: Complete Work Orders Report  30
Figure 25: Incomplete Work Orders Report  30
Figure 26: Expired Lease Report  31
Figure 27: Vacant Units Report  32
Figure 28: Code To Pass Unique Identifier  35
Figure 29: Parent and Child Table Association  36
Abstract

*HRI Rental Property Manager* is a desktop real estate application designed to track rental properties, tenant information, work orders, income, expenses and contractor specialties for Hunter Realty Investments, LLC. The company’s current mechanisms of tracking this information utilizes paper mediums, such as expense workbooks, along with Microsoft Excel spreadsheets. This application provides HRI the ability to organize their rental income and expenses in one place saving a significant number of hours at tax time. The window based application works well for owners with a few rental units as it does for management of extensive properties. This application has been designed utilizing the following three components: Visual C#.Net 2005, Microsoft Access and Crystal Reports. A multi – form tab displayed interface, designed in Visual C#.Net, will give the user the impression of filing information in an organized manner. Microsoft Access will allow for easy data storage, quick retrieval through the interface, and backup. Crystal Reports provides the user with quick compiled summary data. Hunter Realty Investments, LLC will continue to expand and own income rental properties at a rapid pace. HRI Rental Property Manager will provide the owner with an efficient mechanism to stream-line workflow activity, increase productivity, and perform business analysis quickly and effectively utilizing pre-developed reports.
HRI Rental Property Manager

1. Statement of the Problem

1.1 Definition of the Need

Many home based income rental property owners struggle with inefficient paper-based or spreadsheet systems to manage rental property records. Others use personal finance software only to find that these applications fall short for property management. There are many real estate applications on the market today, but most of these specialty software applications charge by the number of units a property owner may have. The more units or properties a real estate investor owns, the higher the cost for a property management system. See figure 1 for a cost comparison analysis of other property management software.

<table>
<thead>
<tr>
<th># of Units</th>
<th>Rent Manager</th>
<th>Rent Right</th>
<th>Landmaster</th>
<th>Landlord Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-10</td>
<td>$149.99</td>
<td>$249.00</td>
<td>$395.00</td>
<td>$74.99</td>
</tr>
<tr>
<td>11-50</td>
<td>749.50</td>
<td>399.99</td>
<td>395.00</td>
<td>147.99</td>
</tr>
<tr>
<td>50-80</td>
<td>1200.00</td>
<td>874.00</td>
<td>695.00</td>
<td>N/A</td>
</tr>
<tr>
<td>&gt;80</td>
<td>1200.00+</td>
<td>874.00+</td>
<td>695.00+</td>
<td></td>
</tr>
</tbody>
</table>

Figure 1: Software Cost Comparisons

Sources: (4,8)
There are several reasons why investors have problems managing properties. As a property investor, he has to adhere to all standard legal obligations of owning property, including the taxes, mortgage, insurance and any other expenses. In addition, he also needs to be aware of the condition of the property at all times and maintain a safe living environment for tenants. There is also the investment of time in managing the property, such as doing repairs, following up with tenants if rent isn’t paid on time, seeing that all rules are being followed, and settling disputes. This is not always an easy task, therefore, he has to manage, track, and maintain accurate records efficiently. Investors with multiple properties also have the responsibility of corresponding with multiple tenants. With a 10 to 1 ratio of tenant to investor, respectively, many tenants are left unattended leaving them dissatisfied and angry (7). Small time investors who continue to use spreadsheets or paper documents find it time consuming when performing analysis across various investment properties (2). It is also difficult to produce various detailed reports, such as tenant payment histories, income reports, late rent, profit/ loss, expense summaries, work orders, and depreciation reports, which are needed for year end tax computations (6). Having the proper software tool will enable an investor better organization and to be better equipped with handling the affairs of their business.

Hunter Realty Investments, LLC (HRI) is a home-based business fairly new to the Real Estate Rental Industry. This small company is operated and managed by one owner. HRI owns several rental income producing properties. The owner is currently managing eight units which are comprised of a single family, duplex, and a multi – unit. HRI is rapidly expanding its real estate portfolio.
The owner is currently managing these properties and tenant records utilizing Microsoft Excel and an expense workbook. These mechanisms have limitations such as the inability to produce reports needed by the owner. It requires a significant amount of time to utilize the pre-built financial functions of MS Excel. There is a learning curve for the owner.

By using Excel and a paper-based system to handle the affairs of HRI, the owner runs into the following problems:

- **Data tracked by numerous worksheets and excel files**
  - Currently the owner tracks tenant records and details, property and unit details, Income and expense reports, work orders, and phone numbers. See figure 2 for Schedule E Tax form needed to track year-end computations.

- **Excessive Time Consumption for Business Analysis**
  - The owner tracks how well each property is performing by tracking unit occupancies and vacancies.

- **No Reporting Functionality**
  - The current system is missing any reporting capabilities. HRI would like pre-developed summary reports.

- **No Data Repository**
  - Data is tracked in various MS Excel workbook worksheets and expense workbooks.

- **No User Interface**
  - HRI is expanding its real estate portfolio, therefore more unit records will need to be maintained and tracked with appropriate data and accurate records. There is constant toggling between numerous workbooks (2).
Figure 2: Schedule E 1040 Tax Form
Source: (9, p. 150)
2. Description of the Solution

The owner of HRI is not technical or computer savvy. This is why the owner and other real estate investors handle the affairs of their businesses utilizing a paper medium system. Therefore to ease the frustrations of a non-technical user my application has been designed and developed with a more simplistic approach. To develop this application I used my programming and database knowledge. This user friendly, sophisticated application eliminates computational errors, streamline workflow processes, and increase business productivity so that the owner can handle tasks effectively and efficiently. The main features of this application are to include the following:

- **Reporting Capabilities**
  - This application will utilize Crystal Reports to provide a reporting feature that will allow the owner to print pre-developed reports quickly and efficiently.

- **User Interface Design**
  - This application will provide a means for the user to quickly and easily add and track property details, tenant information, income, expenses, work orders, and contractor details.

- **Data Repository**
  - This application will provide a centralized location where all data will be stored and retrieved through user interface and reports.

2.1 User Profile

There is only one primary user for this desktop application. The landlord or real estate investor is essentially the owner who bears the ultimate responsibility for any property he or she owns. A landlord is the owner of a house, apartment, condominium or land which is rented or leased to an individual or business, who is called the tenant.
The landlord or real estate investor will use this application to handle the daily responsibilities of a property owner. Property owners have to adhere to many legal obligations therefore the design of my application enables the owner to manage, track and maintain accurate records efficiently. This application allows the owner to enter and update property information, tenant details, work orders, contractors, income and expenses quickly. This application also provides the user to view various summary reports so that he can perform some business analysis.

2.2 Design Protocols

HRI Rental Property Manager is a windows-based single user-application which utilizes the following three major components: Visual C#.Net, Microsoft Access and Crystal Reports. The main graphical and multi-form tab display interfaces were created using Visual C#.Net. The windows form designer in this .Net developer tool provides the fastest drag-and-drop creation for rich desktop applications. In addition the In-Place Menu Editor enables quick and easy creation of main menus and right-click context menus. The toolbox form components save time and alleviate the need for complex coding. Grid views will serve explorer and navigational purposes to the next and previous records. Pre-defined and filtered combo boxes provide the user with quick and easy item selections. Each form will provide the ability to add and edit various types of records. Through these forms the user can enter property details, tenant details, work orders, contractor information, expenses, payment information and printing reports. The outline of the application can be found below. (See figure 3. Page 14)
Figure 3: HRI Rental Property Manager Outline

A Microsoft (MS) Access database is used to serve as the backend system to store all data that is entered and retrieved through the user interface. MS Access user friendly interface allows for quick and easy table and query design. The windows-based desktop
database application is designed for single user applications and is much cheaper than other database management systems. HRI database structure is comprised of 15 tables and 8 queries. The queries are used to retrieve data through the application various summary reports. See figure 4 below for the relational database design.

Figure 4: Database Table Design

All reports that are pre-designed for the HRI Rental Property Manager were developed utilizing Crystal Reports. Crystal Reports can access data from the most widely used databases and can integrate data from multiple databases within a single report, and it has been endorsed as the standard program for reporting with over 300 third
party business solutions. Crystal Reports is a design application for creating powerful and compelling reports that transform data from virtually any data source, into meaningful information. This tool is integrated with the front-end user interface of HRI Rental Property Manager where all reports are executed. In addition, these reports are linked to the backend MS Access database where data will be retrieved from its various database queries. See figure 5 for various report queries.

![Database Queries](image_url)

**Figure 5: Database Queries**
3. Deliverables

In order to effectively define the deliverables for this project, the HRI Rental Property Manager application must:

- Have add/edit/modify capabilities.
- Track expenses paid for per property.
- Track rental income received per tenant.
- Track work orders per property/unit.
- Provide a centralized location for data storage.

In addition to these functions, the application consists of an effective user interface design and provides reporting capabilities. The following lists the details of these functions:

- **User Interface Design** - The interface comprises of multiple forms. These forms track property information, tenant details, work orders, contractor details and occupation specialty.

- **Reporting Capabilities** – This application has several pre-designed reports. These reports include the following: active properties list, a list of lease expirations, active tenant summary list, complete and incomplete work orders, expense summary report, and income summary report.

4. Design and Development

The next sections describe the overall budget and project timeline with hardware and software costs.
4. 1 Budget

The cost to develop and implement this application is described below in figure 6 project budget. This table includes items from the hardware and software sections. The total retail cost of this project is $1,444.00. All hardware and software applications were provided by the developer; therefore the total incurred cost to complete this project is $0.00.

<table>
<thead>
<tr>
<th>Items</th>
<th>Description</th>
<th>Estimated Cost</th>
<th>Incurred Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crystal Reports x</td>
<td>Provided by Developer</td>
<td>$458.00</td>
<td>$ 0.00</td>
</tr>
<tr>
<td>Microsoft Access 2003</td>
<td>Provided by Developer</td>
<td>338.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Microsoft Visual Studio .Net 2003</td>
<td>Provided by Developer</td>
<td>579.00</td>
<td>0.00</td>
</tr>
<tr>
<td>40 GB Portable Drive</td>
<td>Provided by Developer</td>
<td>69.00</td>
<td>0.00</td>
</tr>
<tr>
<td><strong>Total Estimate Cost:</strong></td>
<td></td>
<td><strong>$1444.00</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Total Incurred Cost:</strong></td>
<td></td>
<td></td>
<td><strong>$0.00</strong></td>
</tr>
</tbody>
</table>

Figure 6: Project Budget

4. 2 Timeline

There were a number of tasks involved in completing this project. Figure 7 page 12 displays a more detailed timeline of tasks that were performed over the past several months.
4.2.1 Senior Design I Accomplishments

During the initial research, many hours were spent understanding the daily tasks and legal obligations involved with owning rental property. I researched and downloaded other competitive rental property software to analyze the logic and flow of this type of software. I downloaded web and desktop applications for small and midsize businesses. This research aided me in the design for HRI Rental Property Manager. I accomplished the following During Senior Design I:

- Analyzed desktop and web-based rental property software.
- Researched and gathered client’s needs.
- Structured database and application design.
Developed a proposal and oral presentation.
Presented proposal to technical advisor for approval.

4.2.2 Senior Design II Accomplishments

Senior Design II started off with a challenging start. I have not used Visual Studio.Net for over a year. It took a significant amount of hours to understand how to use this tool and review programming syntax. I accomplished the following during Senior Design II:

- Developed and tested relational database design.
- Developed and tested various user interface designs.
- Prepared and presented Design Freeze.

4.2.3 Senior Design III Accomplishments

Remaining deliverables, continual application testing and minor modifications were accomplished during Senior Design III:

- Developed and tested remaining user interface designs.
- Developed queries for Crystal Reports.
- Designed and developed pre-defined Crystal Reports.
- Tested entire application.
- Made minor modifications to application.
- Clean up code and interface.
- Document and prepare final report and presentation.
- Presented final project.
4. 3 Software

This project was developed using the following software applications:

*Crystal Reports X*: Reporting tool used in conjunction with MS Access and MS Visual Studio .Net to easily access end user information requirements.

*Microsoft Access 2007*: Relational database application used to create tables, relationships, and queries that this project will need.


4. 4 Hardware

The hardware that is used for this project is a 40 GB portable hard-drive for backing up the database and .Net application.

**Minimum System Requirements:**

This project needed the following minimum system requirements in order to run the software applications mentioned above (1, 10):

*Processor:*
A computer with at least 600 MHZ or faster processor.

*Operating System:*
Microsoft Windows 2000 with Service Pack 4
Microsoft Windows XP Home Edition with Service Pack 2
Microsoft Windows Professional*
Microsoft Windows Server 2003*

*Memory:*
A computer with at least 192 MB of RAM or more.

5. Proof of Design

The next section shows in detail how deliverables of this project were fulfilled and the logistics of HRI Rental Property Manager.
5.1 User Interface Design

One of the deliverable requirements is to provide an interface comprised of multiple forms that track property, tenants, work orders and contractor details. The following sections explain how this was accomplished.

5.1.1 Main Form

The main form is the initial screen and it is loaded when the application is executed. This form serves as the parent form that is comprised of several picture button controls which gives the user the ability to easily navigate to all other forms. This includes the property, tenant, contractor, work orders, income and rent forms. The form has mouse-over features to determine which form the user is currently accessing. This is done in the form properties for “ToolTipText”. In addition, the user has the ability to access several menu options located at the top of this form. These menu options are Add/Edit, Reports and Expense/Income. I will discuss these menu options in later sections. (See Figure 8)

Figure 8: Main Form
5.1.2 Property Form

The property form is comprised of a multiple tab display. This consists of a main tab and sub tabs. The main tab is the “Properties” tab. This tab is bound to the properties table in a MS Access database. This is how data is stored and retrieved through the user interface. On the left side of the properties tab is a data grid, bound to the properties table in MS Access, which only displays the property address. This ensures that the user is viewing or editing data for the appropriate property. The purpose of the tab display is to give the user the impression of filing data into its appropriate folders. This is similar to filing data into a filing cabinet. This concept also applies to the tenant, contractor and work order forms. (See Figure 9.)

![Property Form](image)

**Figure 9: Property Form**
Each sub tab consists of links and grids bound to the appropriate tables in MS Access. The sub tabs include mortgage details, insurance details, tenant details and units. To add a mortgage to a property, the user will click the link “Add New Mortgage”. This will open a new form for the user to enter the details of a new mortgage. The unique identifier of the property form for the selected property is passed to this form. (See figure 10.)

Figure 10: Mortgage Form
The user will save the entered information by clicking the blue icon that resembles a floppy disk. A message box will appear that the record has been inserted. (See figure 11.)

![Record Inserted!](image)

**Figure 11: Record Insert Confirmation**

This updates and associates unique identifiers to the appropriate table in MS Access. If user needs to add more than one mortgage for a given property, the user can click the yellow icon that resembles a plus sign. This will clear the text boxes of the previous entered information. The “Close” link will close the current form and show the previous form opened. The “Refresh” link will display the data grid with the information entered for that mortgage. This concept applies to the insurance, tenant and unit sub tabs.

The property form includes option buttons located on the left side of the form for easy filtering, active, inactive and all properties. When the property form loads, all active properties are displayed by default. All data grids have edit, delete, sort and column reordering capabilities. When the user edits any data grid, the save button, floppy disk icon, must be clicked to save any changes. This button will display a message box that the information has been updated. To add new properties to this application, the user must click the yellow icon that resembles a plus sign. This will clear all text boxes for the main tab and prepare the properties table to append a new record. To delete a property, the user must click the button icon that resembles a red x. Add, delete and save buttons on the main tab forms are all data bound. This does not require extensive coding. All sub
tabs add, delete and save buttons are not data bound therefore requires additional coding to program the logic behind these controls.

5.1.3 Tenant Form

This form is accessed by clicking on the button control that resembles two people on the main form. The tenant form has the same concept as the property form. It is bound to the “tenantpersonalinformation” table in MS Access. When the user associates a tenant to a property from the property form, this tenant will display in the data grid on the tenant form. The user has the ability to associate emergency contacts, occupants, vehicles, employer and lease terms for the selected tenant. (See Figure 12.)

![Tenant Form Image]

Figure 12: Tenant Form
5.1.4 Contractor Form

This form is accessed by clicking on the button control that resembles a worker with a hard hat on the main form. The contractors form serves the purpose of storing and editing contact information as well as various occupation specialties. The occupation specialties were derived from other property manager applications. This form is bound to the “contractor details” table in MS Access. This creates a list of contractors used for assigning work orders. The user has the ability to add any additional information in the notes sub tab. (See Figure 13.)

Figure 13: Contractor Form
5.2 Work Orders

This form is accessed by clicking on the button control that resembles a stack of papers on the main form. The work orders form is comprised of one main tab. This tab allows the user the ability to enter and edit new and current work orders. The purpose of work orders is for the landlord to track all services performed, when they were performed and who performed the service on a given property and/or unit. This is useful when tracking expenses. This form is comprised of filtering combo boxes, a data grid and check boxes to indicate complete or incomplete work orders. This form is bound to the “work orders” table in MS Access. The user has the ability to assign contractors to a specific work order. This fulfills the deliverable to track work orders per property and/or unit. (See Figure 14.)

![Figure 14: Work Orders Form](image)
5.3 Expenses

The expense payment form fulfills the deliverable to track expenses paid per property. This form is accessed by clicking on the button control that resembles a check book on the main form. The user has the ability to submit significant information such as Date Paid, Amount Paid, Paid To, Payment Type, Property and Unit. This form allows the user to categorize the type of expense which is helpful when entering information on the Schedule E tax form. This information is pertinent for year-end taxes. When a user selects the appropriate property the units drop-down will populate only those units associated with that property. (See Figure 15.)

Figure 15: Expense Payment Form
HRI Rental Property Manager also provides an additional form to view all expenses logged. This is viewable through the Expense Viewer. The user can access the viewer by clicking on the Expense/Income option in the menu bar. The Expense Viewer gives the user the ability to filter all expenses per property. This drop-down is populated by creating a data source to the expense table in MS Access. (See Figure 16.)

![Expense Viewer](image)

Figure 16: Expense Viewer

5.4 Rental Income

The rental payment form fulfills the deliverable to track rental income received per tenant. Like the expense form, the user has the ability to submit significant information such as Date Received, Amount Paid, Paid To, Payment Type, Property, Unit and Tenant. This form allows the user to categorize the type of income which is helpful when entering information on the Schedule E tax form. This information is pertinent for
year-end taxes. When a user selects the appropriate property the unit and tenant drop-downs will populate only those units and tenants associated with that property.

(See Figure 17.)

![Figure 17: Rent Payment Form](image)

HRI Rental Property Manager also provides an additional form to view all rental income received. This is viewable through the Income Viewer. The user can access the viewer by clicking on the Expense/Income option in the menu bar. The Income Viewer gives the user the ability to filter all income received per tenant. This drop-down is populated by creating a data source to the rent table in MS Access. (See Figure 18 page 25)
5.5 Reporting Functionality

This application includes several pre-designed summary reports. This fulfills the reporting functionality requirement. All reports are exportable and printable. To access reports, the user selects the Report option from the menu bar located at the top of the main form. This will give the user the ability to select and run any of the following reports: Active Tenants, Properties by Type, Expense Summary, Income Summary, Expired Lease Notifications, Vacant Units, Complete and Incomplete Work Orders. (See Figure 19 page 26.)
The Active Tenants report displays all tenants that are currently active in within the system. This information is retrieved by connecting the report’s data source to the “Active Tenant List” query in MS Access. See Figure 20 for report.
The Properties by Type report displays all active properties within the system. It is grouped by the different types of properties a real estate investor may own. This information is retrieved by connecting the report’s data source to the “Property List” query in MS Access. See Figure 21 for report.

![Figure 21: Property Report](image)

The Expense Summary report displays all expenses that have been entered within the system. All expenses are grouped by categories. The report displays a running total for all expenses. This information is retrieved by connecting the report’s data source to the “Expense Summary” query in MS Access. See Figure 22 on page 28 for report.
The Income Summary report displays a history of all income received per tenant. All income is grouped by categories. This information is retrieved by connecting the report’s data source to the “Income Summary Report” query in MS Access. See Figure 23 on page 29 for report.
The following two reports display complete and incomplete work orders. This summary report gives the owner the ability to view pending and completed tasks for a given property. All pending work orders are grouped by high, normal and low priorities. The report data is retrieved by connecting the report’s data source to the “Completed Work orders” and “Incomplete Work orders” query in MS Access. See Figure 24 and 25 on page 30 for sample reports.
Figure 24: Complete Work Orders Report

Figure 25: Incomplete Work Orders Report
The Expired Lease Notification report displays all tenants whom leases are going to expire within 65 days of the lease expiration date. This allows the owner ample time to send out notifications for gathering information from those tenants who will renew and not renew their current leases. This information is retrieved by connecting the report’s data source to the “Lease Expirations” query in MS Access. See Figure 26 for report.

```
<table>
<thead>
<tr>
<th>Name</th>
<th>Property Address</th>
<th>Unit #</th>
<th>LeaseStartDate</th>
<th>MoveInDate</th>
<th>MoveOutDate</th>
<th>Days Left</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retta Halls</td>
<td>123 Hampton St</td>
<td>2802</td>
<td>05/03/2007</td>
<td>05/03/2007</td>
<td>06/08/2007</td>
<td>12</td>
</tr>
<tr>
<td>Antonio Wardell</td>
<td>5400 Walkers Ln</td>
<td>1901</td>
<td>05/22/2007</td>
<td>06/22/2007</td>
<td>07/22/2007</td>
<td>55</td>
</tr>
<tr>
<td>Leslie Gaines</td>
<td>5300 Marion st</td>
<td>101</td>
<td>05/24/2007</td>
<td>06/24/2007</td>
<td>07/24/2007</td>
<td>57</td>
</tr>
<tr>
<td>Juanita Gilbert</td>
<td>4900 Farmers Ln</td>
<td>4001</td>
<td>05/24/2007</td>
<td>06/24/2007</td>
<td>07/24/2007</td>
<td>57</td>
</tr>
</tbody>
</table>
```

Figure 26: Expired Lease Report

The Vacant Units report displays all properties and their vacant units. This allows the owner to determine how well a property is performing. This information is retrieved by connecting the report’s data source to the “Vacant Units” query in MS Access. See Figure 27 page 32 for report.
5.6 Data Repository

The Microsoft Access database application serves the purpose of storing all data in one centralized location. The database structure ensures data integrity amongst records submitted and retrieved thru the user interface. This fulfills one of the deliverable requirements.

5.7 Add and Edit Functionality

All forms include two button controls. One button resembles a yellow plus sign. This button prepares a new record to be added to the database. The other button resembles a floppy disk. This button will behave in either two ways. It may update the current record selected by the user or it may insert a new record into the appropriate table.
in MS Access. All records in a data grid are editable. In order to save the edited record, the user must click the save button.

6.0 Testing Plan

For the project to be successful, testing has been performed throughout every module of the application. The testing plan involved testing all functional requirements. The following key functions are defined below:

- Interface used to add, modify, and save data to the appropriate database tables.
- Data integrity.
- Parameters passed to appropriate forms.
- Tool strip and menu buttons opening appropriate forms.
- Reports displaying correct data.
- Error handling capabilities.
- Required field inputs.
- Sorting and viewing data for all.

This application is tested by the HRI real estate investor as well as other rental investors. Careful planning and logic flow were very critical during the development phase. The user requires reliable and fast access to data. Since all database communications must pass through the connectivity component, bugs or flaws in the data access architecture or implementation can cause performance and support problems. The user reliance on these solutions to return consistent and accurate results from database queries, high-performance data connectivity is an important component needed to build an efficient system.

7.0 Conclusion and Recommendation

7.1 Conclusion

Hunter Realty Investments will continue to expand and own income rental properties at a rapid pace. Although this company is one owner-operated, its current
mechanisms used to track, manage, and maintain data information were proved to be inefficient. The development of HRI Rental Property Manager covers application programming and database design. This cost-effective tool will provide the owner with a better equipped application to streamline workflow activity, increase productivity, and perform business analysis quickly and effectively utilizing pre-developed reports.

7.2 Recommendation

The challenging aspect of this entire project is the proficiency of programming in C#.Net and how to use the Visual Studio.Net environment. I have not used Visual Studio or programmed in over a year. The only experience I ever utilize these tools were in Contemporary Programming I and II classes. Therefore, I needed to take a mini course to refresh my memory. This allowed me to successfully complete this application.
Appendix A

Throughout the HRI Rental Property Manager many sub tabs contain links to open child forms in order to append records to various database tables. When the user clicks the link to open a child form this event captures the unique identifier from the parent form and passes it to the child form. This will ensure data integrity. (See Figure 28).

```
private void linklblMortgage_linkClicked(object sender, LinkLinkClickedEventArgs e)
{
    FrmMortgage2 x = new FrmMortgage2();
x.bidIDTextBox.Text = this.bidIDTextBox.Text;
x.EdiParent = this.ParentForm;
x.Show();
}
```

**Figure 28: Code to Pass Unique Identifiers**

When the user submits a new record into a database table from a child form the parent form unique identifier is captured and inserted into the appropriate database field. This is how specific records are associated with records from parent forms. (See Figure 29 page 36.)
private void mortgageDetailsBindingNavigatorSaveItem_Click(object sender, EventArgs e)
{
   DataRow drNewMortgageRow = hRIDataset.MortgageDetails.NewRow();
   drNewMortgageRow["mtgCompany"] = txtboxMtgCompany.Text;
   drNewMortgageRow["mtgLoanNumber"] = txtboxMtgLoanNumber.Text;
   drNewMortgageRow["mtgOriginDate"] = mtcOriginationDate.Text;
   drNewMortgageRow["mtgType"] = mtcTypeComboBox.Text;
   drNewMortgageRow["mtgLoanCost"] = txtboxMtgLoanCost.Text;
   drNewMortgageRow["mtgBuildingCost"] = txtboxMtgBuildingCost.Text;
   drNewMortgageRow["mtgYearsFinanced"] = txtboxMtgYearsFinanced.Text;
   drNewMortgageRow["mtgPayment"] = mtcPaymentTextBox.Text;
   drNewMortgageRow["mtgDateClosed"] = mtcDateClosed.Text;
   drNewMortgageRow["mtgAmountFinanced"] = txtboxMtgAmountFinanced.Text;
   drNewMortgageRow["mtgLoanType"] = mtcLoanTypeComboBox.Text;
   drNewMortgageRow["mtgPercentage"] = txtboxMtgPercentage.Text;
   drNewMortgageRow["mtgContactName"] = txtboxMtgContactName.Text;
   drNewMortgageRow["mtgContactPhone"] = txtboxMtgContactPhone.Text;
   drNewMortgageRow["blID"] = int.Parse(txtblIDTextBox.Text.ToString());
   hRIDataset.MortgageDetails.Rows.Add(drNewMortgageRow);
   hRIDataset.Validate();
   this.mortgageDetailsBindingSource.EndEdit();
   this.mortgageDetailsTableAdapter.Update(this.hRIDataset.MortgageDetails);
   MessageBox.Show("Record Inserted!");
}

Figure 29: Parent and Child Table Association
References

1. “Checkout Software”

   Personal Interview. October 25, 2006


   Personal Interview. October 20, 2006.


