An IS Capstone Project: The Mywick Property Management System

Martha Myers
Computer Science and Information Systems Department
Kennesaw State University
1000 Chastain Road
Kennesaw, GA 30144
mmyers@kennesaw.edu

ABSTRACT

This paper describes a project suitable for use in an upper division course requiring the completion of analysis, design and implementation of a software system. It is especially useful for illustrating multiple cycles through the development process, as well as for integrating key concepts from a number of fundamental knowledge clusters in a typical undergraduate IS program of study. These knowledge clusters include database management, project management, programming principles, and system analysis and design. In the pilot class, students employed use case analysis as well as standard object-oriented techniques such as class diagrams and sequence diagrams. Students followed a modified version of the Team Software Process, with special attention paid to the postmortem following each of two cycles. Teaching tools included the use of a special email address allowing students to converse with the "client."

Keywords: iterative development, team process, software development, object-oriented analysis and design, database application, programming, and triage

1. BACKGROUND

The Mywick Residential Rental Property Management System case was designed for use in a capstone project course for undergraduate information systems (IS) majors. As such it is particularly useful in classes where a completed team software system is required. It was designed to elicit the development of a front-end data entry and reporting system, as well as a companion back-end database system. Students are expected to engage in analysis, design, and implementation using object-oriented techniques (Dennis, 2002). This case is also useful in teaching the need to set system priorities. The concept of triage (Yourdon, 1999) is better understood within the context of a case requiring that choices be made. While oriented towards a completed system, this case may also be useful in a standard database or system analysis and design course where students do not actually implement or write code.

2. THE CASE

2.1 Context

Ms. Jane Mywick is a real estate owner who has owned and rented residential units of duplexes and 4-plexes to individuals and to families for about 15 years. She currently owns 12 properties, each with 2-4 units. She expects that her real estate holdings may grow to as many as 15 properties, but very gradually over several years. Her tenants tend to be of modest income, many of whom receive rent subsidy from federal agencies or other resources, such as parents. Her philosophy towards collection of rent tends to emphasize collecting the rent for the current month, encouraging tenants to pay their rent on time and to avoid falling behind. Most of what follows has been gleaned from a single interview with Ms. Mywick. Follow-up questions may be sent by each team's leader to Ms. Mywick.

2.2 Problem Statement

Management of residential rental property requires tracking several aspects of income and expenses by tenant, by unit, by property and in aggregate. Ms. Mywick currently uses Quick Books to track expenses, but rents are tracked manually. As a result, she is unable to see where she stands with respect to unpaid rents and total income for tax filing. Several off-the-shelf rental property management packages are currently available but most try to satisfy all possible property management requirements. The result is a product that is overly complex for Ms. Mywick's purposes and actually adds to her work load. For example, several are oriented around an on-site property manager who enters payments and issues receipts on the spot. Ms. Mywick is an absentee property manager who receives checks from tenants via regular mail. Also Ms. Mywick often receives partial payments for rent. So

Ms. Mywick would like for you to develop a system to track rents.

In the past, Ms. Mywick has tried several methods for tracking rents including paper grids, one per unit, and a similar approach in Excel where one spreadsheet was generated for each unit. Samples of these documents may be found in Appendix A.

2.3 Basic Requirements – Data Capture

Ms. Mywick has asked you to design and implement a software system to track rental payments. Specifically she needs a system that will allow her to efficiently record rents received and track unpaid rent. The solution system should allow Ms. Mywick to record checks from tenants in a batch mode at various times over the course of a month. She also needs to record payments from the federal government Housing Assistance Program (HAP). Each month HAP sends her one check covering subsidies for all related units. An itemized list is enclosed with the HAP check showing how the subsidy amounts are distributed among the tenants. So Ms. Mywick must also be able to enter this information in a batch mode.

The system should allow her to enter rents owed and must handle recurring amounts based on a one-year lease, but allow for monthly adjustments. Also the system must provide for multiple sources of payments, like federal subsidy, multiple tenants each paying a portion, or parents helping out with some part of the rent. For example, rent for Chris Brown might be $753 per month. The HAP subsidy is currently $542 per month and Chris is responsible for the rest ($211). If Chris's income changes, then the HAP subsidy changes. So rather than paying $542, the subsidy might go down to $258. This then increases the amount owed by Chris to $495.

Only very basic information is required on tenants, like names (one or two), phone #s (up to 3), and deposit amount paid. Requiring too much information per tenant means too much data capture, and unnecessary work overload for Ms. Mywick. Probably first and last names of one or two people per unit are adequate.

Data entry of rental checks is usually done in batches by type of transaction. HAP sends a single check to Ms. Mywick at the beginning of each month. This one check covers all subsidized rent tenants. An itemized list of relevant tenants and amount provided for each is enclosed with the monthly HAP check. See Figure 1 for an example of HAP detail.

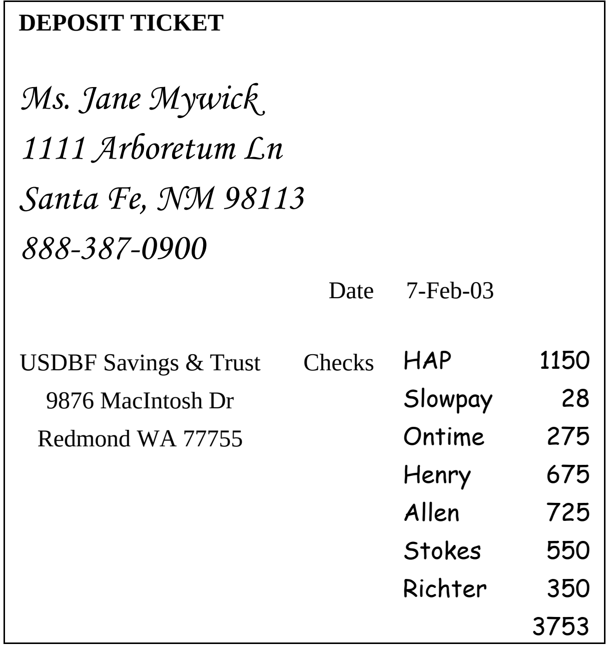
Ms Mywick visualizes a system that she would start and then be able to select "enter HAP payment for this month." The system should then cycle through the tenants that have subsidy payments due and should pre-populate subsidy amount fields (e.g., $542 in the above example) where possible. For example, if entering subsidy payments, the payment amount field should show the expected subsidy amount, as well as the total rent due. Also, if the expected amount is overridden, then there should be the ability to memorize that amount as the new recurring amount and recalculate the amount expected from the tenant.

Figure 1
Sample of HAP Detail

Tenant Name	Reference	Amt
Fred Slowpay	1201B Cedar Ln	300
Helen Ontime	3001A Oak Dr	450
George Bush	4453D Mimosa Rd	200
Aida Richter	4453C Mimosa Rd	200
	TOTAL	1150

Tenants pay with checks. Rent is due on the first and late after the fifth. Ms. Mywick usually batches 5-10 checks together in a single deposit to the bank. See Figure 2 for an example of a deposit.

Figure 2
Sample of Deposit Slip



DEPOSIT TICKET

Ms. Jane Mywick
1111 Arboretum Ln
Santa Fe, NM 98113
888-387-0900

Date 7-Feb-03

USDBF Savings & Trust
9876 MacIntosh Dr
Redmond WA 77755

Checks		
	HAP	1150
	Slowpay	28
	Ontime	275
	Henry	675
	Allen	725
	Stokes	550
	Richter	350
		3753

Ms. Mywick would like to be able to record the related tenant and check amount in a batch mode in this system. Because tenants often pay only part of the rent, Ms. Mywick needs to be able to run such a batch process multiple times per month, as checks arrive. It would be nice if each expected rent amount were displayed on the form where she enters the amount of each check from a tenant. A grid-like format would be ideal for this activity, as well as for the HAP check processing described above.

Data captured in this system varies in terms of its volatility. For example, the addresses of the units are static. Ms. Mywick has not bought any new properties for two years. Rents described in leases usually stay in place for one year, so that information, including tenant names and phone numbers, is more volatile than the address information, but still fairly stable month-to-month. The sources of rent, and distribution among the sources, are the most volatile. Ms. Mywick must be able to rely on the system remembering the last amount received via subsidy, but should be able to change that during any monthly keying of subsidy payments.

Allowing for other adjustments is also important. For example, Ms. Mywick might choose to write off some amount of unpaid rent from an otherwise good tenant. If a tenant has paid rent completely and on time for 9 out of the last 10 months, then Ms. Mywick might write off the lost rent in a previous month, particularly if the most recent payments are complete and timely. Or sometimes a tenant does work on the unit in exchange for rent. This may reduce the amount due or the past due amount. A comment field would be important to describe the reason for the adjustment, as it would appear in later reports.

2.4 Basic Requirements – Reporting

The system should allow Ms. Mywick to enter rents paid against rents owed. She must be able to track outstanding amounts due on the first of the month, as well as overdue rent from a previous month. She needs to know how much is due for this month, as well as how much was not paid from previous periods. It would be nice if this could be reported by period, like last month, or last 60 or 90 days.

Example reports include the current month payment history by unit showing rent due, payments made, and balance due. Exception reporting should show past due amount by unit, showing amount past due per period. Other reports should display total income by unit by month, and some ability to report by tenant rather than just by unit. This requires keeping track of current tenant as well as past tenants, but this is a nice-to-have and not essential.

2.5 Definitions

Tenant – one person or group of persons living together. Everybody in the group is liable for all the rent. So, for example, if one of two renters leaves in the middle of the night, the remaining person is responsible for all rent.

Unit – one single family dwelling. This is a unique mailing address such as 1201B Prospect Avenue.

Property – a structure that might include multiple units, that is, a duplex or a 4-plex. In the above example, 1201 Prospect Avenue represents the property; 1201B Prospect Avenue represents the unit.

2.6 Desirable Features

If time allows, Ms. Mywick would like to have the ability to export individual transactions to Quick Books with proper coding for producing Quick Books income reports. For effective use in Quick Books, additional information is needed to properly categorize the transaction. Provision is needed for entering appropriate information that is then used to produce a fully encoded Quick Books transaction. Tracking expenses might be a future feature but not nearly as important as effective income management.

Ms. Mywick will of course need to be able to enter the actual property and unit information initially when she installs the system, but afterwards, she will make infrequent changes to unit or property information stored in the database. Leases usually run for a year with few intervening changes. Thus, the user interface for these activities need not be as efficient as the interface for entering rental checks.

2.7 Case Summary

The most important challenge is to allow for efficient tracking of expected rent payments and actual payments received. Ms. Mywick must be able to track rents received and expected rents - by unit, by tenant, and based on a date range. Of particular note is that the amounts from any given tenant may change in any given month, most often in the middle of a lease agreement, due to changes in subsidy amounts. She must also be able to accept partial rent payments. Ms. Mywick needs flexibility in terms of accounting for the full rent from various sources, including subsidies, write-offs and special agreements with tenants.

2.8 Follow up Q&A 1

The following questions and answers (Q&A) are based on a follow-up interview with Ms. Mywick.

Q: Are there any single unit properties?
A: No.

Q: Would you like the ability for a lease document to be generated from a report that queries the tenant entry table and populates specified fields based upon the tenant entry form and unit specifications?
A: Not at this time.

Q: Would you like a form to add new units to the database
A: Yes.

Q: Should you have a field associated with the deposit paid field to apply the deposit to the last month's rent or a cash back feature upon properly approved move out?
A: Not to the last month’s rent, but possibly a cash back feature. It’s not a high priority.

Q: Should the application have the ability to charge standard utility or service amounts that are standard with each month's invoice, such as, cable billing, housekeeping, or trash charges at a standard rate for each month?
A: I don’t cover any of these utility or service charges. Rent is a flat amount. Tenants are responsible for utility bills.

Q: Will there be multiple users for the application, in which each user type will only be allowed to view certain

information and have the specific permissions associated with that user's position?
A: My partner may become involved but we will need the same capabilities (full).

Q: Should there be a secured login screen?
A: Not need.

Q: Does the application need to run on a multi tiered system where the database may reside on a server and the front end on each client?
A: No.

Q: What is the primary operating system that the application will be running on? And what type of network architecture will the system be?
A: This system will probably run on a stand alone PC, probably XP.

Q: Should there be a field that allows you to put a status associated with a Tenant, such as if a tenant is 2 months behind in payment they could be red flagged and an automatic notification could be posted to remind the user to cancel lease and evict them?
A: Don't know about a special "field" but I need a way to track payments that are behind. For example, a useful report would list all the tenants who are behind, the dollar amount behind, and the date of last payment.

Q: Should there be some sort of trigger to notify the user that a tenant's one year lease is up and they need to update there lease information?
A: A report that could be run might help with this.

Q: Will there be units with more than 4 properties?
A: No.

Q: How do we determine rent? Is it based on the unit or tenant?
A: It is based on the unit.

Q: Do you need to track the property/structure, or just the units, tenants and payments?
A: I need to be able to track by property, by unit, by tenant, AND by payment.

Q: Do you need to track expenses by property/structure?
A: I do not really need to track expenses with this system. I am much more concerned about tracking rents.

Q: Do you need the ability to export data for a backup?
A: Probably. Also, at some point, I might like to export data into Quick Books.

Q: Do you need to query data by: name, unit, and property/structure?
A: Yes.

Q: Do you need payment receipts?
A: No.

Q: Do you need to enter payments by unit, property or tenant?
A: Rents are entered by tenant, and also for subsidy payments of rents.

Q: For the login, do you want it to go to a switchboard-type menu to enter payments, add units, and other tasks?
A: That sounds about right.

Q: Is there a specific GUI style that you would prefer? Could you list some programs that you are familiar with where you are comfortable with the interface?
A: I'm familiar with most Windows applications (e.g., the Office Suite) plus Quick Books and a number of other software packages. I suggest you follow the MS standards.

2.9 Follow up Q&A 2

The following questions and answers stem from a second follow-up conversation with Ms. Mywick.

Q: Should we track what type of property (2 plex or 4 plex)?
A: Yes, this is one characteristic of a property.

Q: Should we include a unit description (Like a 2 bedroom apartment, with 1 bath, something to identify what type the unit is)?
A: Yes, but very brief. You might create radio buttons or check boxes.

Q: Should we include a billing address for tenant (example if the billing is going to the tenant's mom and dad should we have fields available to input this information)?
A: Optional, might be nice to have. Certainly tenant telephone numbers might be helpful. Of course we already know tenant address because we know where they live.

Q: Do you need to track who the deposit refund goes to?
A: This is not very important - optional.

Q: Does the deposit refund go to the person who paid it or the first or last person who leaves?
A: It will only be paid when the unit becomes empty. We will pay the person(s) remaining (if you implement that option).

Q: Do you want to know how many properties available?
A: Yes, we need to have several possible reports. A list of vacancies is one possible report. We also need some on-line retrieval of current vacancies.

Q: What is the current business environment? Is it a manual, Excel or other computerized system?
A: We keep the books for tax purposes in QuickBooks. We record the actual rents paid and expenses at the end of the month.

Q: Do we need to archive tenant information?

A: Yes, but not critical.

Q: Do we need to archive payment information?
A: Yes - it would be nice to run reports of rent history by unit. We also need to run rent history by tenant as long as the tenant is in residence.

3. ACKNOWLEDGEMENTS

The author gratefully acknowledges the support and insights provided by James Myers, as well as the enthusiasm and hard work provided by her spring 2003 CSIS 4830 class at Kennesaw State University.

4. REFERENCES

Cappel, James J. [2002] "A systems Analysis and Design Case: ABC Church," Journal of Information Systems Education, Vol. 12(4).
Cappel, James J., and Paul H. Schwager [2003] "Writing IS Teaching Cases: Guidelines for JISE Submission", Journal of Information Systems Education, Vol. 13(4).
Dennis, Alan, Barbara H. Wixom, and David Tegarden [2002] Systems Analysis and Design: An Object-Oriented Approach with UML. John Wiley & Sons, Inc.
Humphrey, Watts. S. [2000] Introduction to the Team Software Process. Addison Wesley Longman, Inc.
Yourdon, Edward [1999], Death March. Prentice-Hall, Inc., Upper Saddle River, New Jersey 07458.

AUTHOR BIOGRAPHY

Martha Myers, Ph.D. is professor of Computer Science and Information Systems at Kennesaw State University in Georgia. She has 15 years of experience in IT education and many years industry experience both as a consultant and management within the insurance industry. She has published in the *Journal of Computer Information Systems, Ubiquity, Communications of the Association of Information Systems,* and *Journal of International Information Management.* Research interests include IS career development and IS pedagogy.

aitp Association of Information Technology Professionals

No matter how sophisticated the technology, it still takes people! ™

EDSIG

Serving Information Systems Educators

STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the Journal of Information Systems Education have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

ISSN 1055-3096