

CS 4400 PROJECT (Fall 2008)

Apartment Complex Management System (Version 1.0)

This document contains a general description of the requirements for the project. Please note the version number since we are likely to revise it after student feedback. The deliverable for different phases are mentioned at the end of document.

1. INTRODUCTION

Four Point Property is a real estate company that has 2 apartment complexes, namely North East and South West. The apartments are located in Atlanta. Each apartment is headed by an apartment manager. The company has several contractors to maintain the apartments. In this project, we will develop software to help Four Point Property to manage the apartments. The users of the system are the managers, the tenants, and the contractors of the apartment complexes. The system allows tenants to submit the repair/ maintenance requests online and check for the requests' status. The manager of an apartment complex can approve or deny the pending requests. Upon the approval, the manager needs to choose the contractor who will perform the service. The general manager can view the summary reports about the apartment complexes. The detailed requirements for the tasks are mentioned in the tasks below.

2. TASKS

All Users

T1. User Login:

Description: a tenant, contractor, apartment manager or general manager can log in the system.

Input: uid, password.

Output: log into the system according to user type.

Apartment Manager

T2a. View Payments:

Input: unit number.

Output: due date, payment date, status (on time, late, not paid), #overdue days, amount. Ordered

by due date.

Constraints: only shows the next due date that a tenant has to pay, which means the tenant must make the payment in the order of month. When the user pays the listed payment, the next upcoming payment will be shown as long as it does not exceed the end date of the lease or the renewed lease. The status of payment can be on time, not paid (upcoming payment), overdue (not yet paid), or late (paid after the due date).

Reference: illustration 1.

Payment for Unit for South West Apartment

Tenant Name: *Avalon Butcher*

Lease Expired On: *3/31/20009*

Due Date	Payment Date	Status	#Overdue Days	Amount	Action
01/01/2009		Not paid	0	1000	Pay
12/01/2008	11/30/2008	On Time	0	1000	
11/01/2008	11/01/2008	On Time	0	1000	

Illustration 1

T2b. Enter payment:

Input: unit number and payment date. From the task T2a, when the user clicks on the pay button, the system will ask the user to enter the date of payment.

Output: updated payment.

Reference: illustration 1.

T3. List of Available Units:

Output: available date, #bed, #bath, floor, unit#. Ordered by available date.

Constraint: show the units that are currently available or will be available within two months.

Reference: illustration 2.

Current & Future Available Units for South West Apartment

Available Date	Unit	#Beds	#Baths	Floor
Now	1	1	1	1

1/1/2008	2	1	2	1
----------	---	---	---	---

Illustration 2

T4. List of Current Leases:

Output: unit info, tenant's name, #late payment (for the current lease), last payment, status (renewed, expired in one month, in progress), renew lease button. Ordered by unit number.

Constraints: can renew lease if expiration date is within 1 month of the current date.

Reference: illustration 3.

Current Lease Information for North East Apartment

Unit	Tenant	#Late & overdue Payments	Last Payment	Expiration Date	Action
1	Stefany Holmes	1	12/1/2008	12/31/2008	Renew
2	Michael Brokeman	2	12/3/2008	3/31/2009	

Illustration 3

T4b. Renew Lease:

Description: when the user clicks the renew link in task 4a, the system will show a window that lets the user enter the unit#, rate, and term of the renewed lease. The term can be 2 months, 3 months, or 6 months.

Input: unit#, rate, term.

Reference: illustration 3, 4.

Unit ▾

Rate

Term ▾

Illustration 4

T4c. Tenant's Info:

Description: when the user clicks on the name of a tenant in task T4a (illustration 3). The system

will show the detail information about the tenant.

Output: first name, last name, phone number, former address, email.

Reference: illustration 3.

T4d. Update Tenant's Info:

Description: the system allows the user to update the tenant information in task T4c.

Input: any new information (first name, last name, phone number, former address, or email).

Output: update tenant's information.

T5a. View Request:

Description: the manager of an apartment complex can view the list of requests made by the current tenants. The output shows the status of each request which can be pending, approved, or done. To approve a pending request, the user should choose a contractor prior to approving the request.

Output: ordered by request date

Constraints:

Reference: illustration 5.

Requests for Current Leases for North East Apartment

Request Date	Unit	Tenant	Service Type	Description	Status	Action	Serviced By
12/2/2008	2	Michael	Carpet Cleaning	Need to be cleaned	<i>Pending</i>	Approve Reject	ANZ v
2/1/2008	1	Stefany	A/C	No air	Approved		ANZ v
1/2/2008	2	Michael	Painting	Old Paint	Done		Monroe v

Illustration 5

T5b. Approve/Deny Request:

Input: if the user approves the request, then the user needs to select a contractor, otherwise, the default contractor will be chosen.

Output: updated request.

Constraints: the "serviced by" drop down list shows only the list of contractors that can perform

the requested service.
Reference: illustration 5.

Customer

T6a. Check Request Status:

Output: request date, service type, description, status, service date. Ordered by request date.
Constraints: only for the current lease.

Request Status for Michael Brokeman

Request Date	Service Type	Description	Status	Service Date
12/2/2008	Carpet Cleaning	Clean	Pending	
11/2/2008	Painting	Old	Done	11/10/2008

Enter Request

Service Type

Description

Illustration 6

T6b. Enter Request:

Input: service type, description.

Output: a new request with the request date.

Reference: illustration 6.

Contractor

T7a. View Pending Requests

Output: apartment, unit number, service type, request date. The output is ordered by request date.

Reference: illustration 7.

Pending Requests for ANZ

Request Date	Apartment	Unit#	Service Type	Description	Action
12/3/2008	South West	3	Plumbers	Not working	Update
12/2/2008	South West	3	A/C	No heat	Update
12/1/2008	North East	1	A/C	No air	Update

Illustration 7

T7b. Update Request Status

Description: after the user clicks on the update button in task T7a. The system asks the user to enter the date that the service will be performed. Then the request status will be changed to DONE. The request is removed from the list of pending requests.

Input: service date

Output: the request is updated with service date.

Constraints: the service date can not precede the request date.

Reference: illustration 7.

General Manager

T8. View Apartment Information for the given month:

Input: month/year

Output: list of apartments with the manager's name, #units, #available, # rented, #requested services for the given month, #late payments for the given month. The output ordered by apartment name.

Constraints: only for the selected month.

Reference: illustration 8.

Apartment Information For

Apartment	Manager	#Units	#Rented	#Requested Services	#Late Payments	#Overdue
North East	Jonathan	4	2	2	3	0
South West	Jessica	4	3	2	1	1

Illustration 8

T9. View Performed Services by Contractors:

Output: the list of contractors and the corresponding total number of services performed by each contractor for each apartment complex. Ordered by contractor name and apartment complex.

Reference: illustration 9.

Performed Service Information

Contractor	Phone	At	Total
ANZ	404-405-9111	North East	3
		South West	1
Metro Service	404-402-2700	North East	0
		South West	0
Monroe Ltd.	678-687-6781	North East	1
		South West	0
NBC Corp.	404-403-4700	North East	0
		South West	0

Illustration 9

3. GENERAL INFORMATION

GENERAL INFORMATION

GROUPS: Each group must have 3 or 4 members.

As a group, you will decide whether to complete the lightweight or heavyweight project options. The two options are identical for phases I and II, but differ in the deliverable for phase III. Note that the option of whether you wish to do heavy or light weight can wait until you get into phase III and as late as the final submission of Phase III.

Heavyweight Option

Groups choosing this option will demo a working implementation of their project to the TA. The implementation must include a Java or web-based GUI (Graphical User Interface) that uses JDBC (Java Database Connectivity), MySQL or ODBC (Open Database Connectivity) for database access. The SQL statements you create in phase II will be embedded inside your GUI.

Lightweight Option

Groups choosing the lightweight option will submit working SQL statements for each of the project tasks and demo the SQL statements to the TA. This option may be appealing to groups with little or no experience programming GUIs.

Oracle

We will provide you with access to the Oracle Database Management System on ACME. See the course web page for further information on how to access Oracle from the ACME command line or from a Java program.

DELIVERABLES

PROJECT PHASES

The three phases of the project cover the following tasks. Specific deliverables are defined for each of the three phases below.

PHASE	DESCRIPTION	DUE DATE
I	Analysis & Specification	Sept. 24
II	Design	Oct. 22
III	Implementation & Testing	Dec. 2
	Demonstration	Dec. 3-5

PHASE I (hard copy)

The deliverables include:

1. ER-Diagram (ER).
2. Information Flow Diagram.
3. A list of logical constraints that will be enforced. (Do not include data-type constraints, but rather semantic business logic related constraints).
4. Any assumptions made. Include explanations.

Notes:

1. The ER must capture the constraints of the system as much as possible whenever applicable, i.e. total participation, super/sub class, weak entities. If there is any drop down list in the system, it should be modeled by an entity.
2. The design must satisfy all the constraints. The students are allowed to make up additional assumptions and constraints as long as they do not conflict with the specified constraints and requirements. Those additional assumptions and constraints should be written in ER-Diagram.
3. You must turn in your report in hardcopy in class attached with a cover page that includes the names of all group members, their class section and their email address.

Useful Link for Phase I:

4. http://www.cc.gatech.edu/classes/AY2008/cs4400_spring/methodologyFall2002.ppt

PHASE II (hard copy)

1. Cover Page.

2. Copy of the E-R Diagram (either from phase I (with any revisions) or from the solution).
3. Copy of the Information Flow Diagram from phase I (either from phase I (with any revisions) or from the solution).
4. Relational Schema Diagram (with primary and foreign keys identified, referential integrity is shown by arrows).
5. Create Table statements, including domain constraints, integrity constraints, primary keys, and foreign keys.
6. SQL statements for each task (*follows the template in the phase II guideline*).

Notes: A set of SQL statements may be required in order to complete one task. However, in such cases, the last SQL statement should show the output according to the specification. If mentioned, the returned tuples must be ordered according to the specification. The last SQL should resemble the output as much as possible. Views and nested queries may be used to support the tasks. A nested query can be broken down into views to make the query more readable.

PHASE III

Prior to the demo, the TA will give a sample data set. The database has to be populated with this data set prior to the demo. **5% will be deducted from the grade, otherwise.**

LIGHTWEIGHT: write and test a set of working SQL statements to perform the tasks specified in the project description.

HEAVYWEIGHT: implements a working application with all functionality described in the GUI. Source code will be mailed to the respective TA who grades your project by the deadline.

Deliverables for Phase 3 are:

- Copy of the Create Table statements from phase II (with any revisions),
- Source Code (documented) for your System,
- A set of working SQL statements for all project tasks (*Lightweight Option*)
- A functional GUI with embedded SQL statements that accesses your database (*Heavyweight Option*)
- A system demo to one of the TAs (you will use SQLPLUS or MySQL (you need to setup your own) if you choose the light weight option.

