
SOFTWARE REQUIREMENTS SPECIFICATION

for

Condo Management System

Version 1.0 draft

Prepared by Mhaisdhune, Trupti Vishnu
Peixoto Costa Neto, Mario
Yohannan, Blessy
Zhang, Xuezhu

Texas State University

February 16, 2017

Contents

1	Introduction	4
1.1	Purpose	4
1.2	Problem statement	4
2	Overall Description	5
2.1	Product Perspective	5
2.2	Software Interfaces	5
2.3	Product Functions	5
2.3.1	Create Client	5
2.3.2	Edit Client	5
2.3.3	Remove Client	6
2.3.4	Create Condo	6
2.3.5	Edit Condo	6
2.3.6	Remove Condo	6
2.3.7	Create Unit	7
2.3.8	Edit Unit	7
2.3.9	Remove Unit	7
2.3.10	Create Tenant	7
2.3.11	Edit Tenant	8
2.3.12	Remove Tenant	8
2.3.13	Create Debt	8
2.3.14	Edit Debt	8
2.3.15	Remove Debt	9
2.3.16	Create Notice	9
2.3.17	Edit Notice	9
2.3.18	Remove Notice	9
2.3.19	Import Monthly Report	10
2.3.20	Emit Notices	10
3	Other Nonfunctional Requirements	11
3.1	Performance Requirements	11
3.2	Reliability Requirements	11
3.3	Robustness Requirements	11
3.4	Maintainability Requirements	11
3.5	Usability Requirements	11

4	Constraints	12
4.1	Design	12
4.2	Implementation	12

1 Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of the Condo Management System, a system designed to manage clients, condos, tenants, units and debts and provide organized records of each tenant debts and emit notices for those debts. It will explain the functions that the manager will be able to access, the interfaces of the system, what the system will do, the constraints under which it must operate and how it will react to external stimuli. This document is intended for both users and developers of the system and will be submitted to the customer for approval.

1.2 Problem statement

A law office works as a representative of several condominium administration companies. These companies may have several hundreds tenants who rents their apartments and it's very common for those tenants to skip rental and administration payments of the condo. The administration company has its own information system, but the law office only receives a monthly report with the up-to-date debts for all condos and units.

The job of the law office is to look at the reports and find the tenants who are in debt for more than a certain time and notify them that they should contact the law office to settle their debts or expect to be sued. But having several companies as clients - each of them having several hundreds of tenants - can lead to a tremendous amount of labor, it's just not doable.

Each client will provide a monthly debt report with condo names, units in debt, tenants in debt, and debts information and the law office will use that report to import the data into the database.

Then, the system shall provide a screen to help the law office to easily find who's in debt, for how long they are in debt and how much they owe. That same interface shall allow the law office to choose the tenants to be notified. After choosing, the system shall generate all the notices and the envelope labels for mailing as PDFs.

After notifying the tenants, they may contact to negotiate a settlement or to pay the full amount. Then, the system shall provide a feature to mark those debts as paid.

2 Overall Description

2.1 Product Perspective

The Condo Management System is a new web-based application and does not depend or is a part of any other system. It has one actor: the manager. The manager needs a compatible web browser and internet access to access the system.

2.2 Software Interfaces

PostgreSQL will be used as a Database Management System for the Condo Management System.

Compatible web-browsers: Chrome, Firefox and Safari.

2.3 Product Functions

2.3.1 Create Client

Description

The manager creates a client (Condo Administration Company), entering information such as name, contact name, contact phone, address, default fine percentage and default interest percentage for late payment.

Rationale

This function will make the client available to the system in order to create the condominium managed by this client. That will be necessary to import the debts and keep an organized record.

2.3.2 Edit Client

Description

The manager edits a client (Condo Administration Company), entering information such as name, contact name, contact phone, address, and default fine percentage and default interest percentage for late payment.

Rationale

This function will make possible to edit the client information whenever the data's changed or it was entered incorrectly. That's necessary to keep an organized record.

2.3.3 Remove Client**Description**

The manager removes a client (Condo Administration Company) from the system.

Rationale

This function will make possible to remove the client from the system if the client decide to cancel the contract with the law firm. That's necessary to keep an organized record.

2.3.4 Create Condo**Description**

The manager creates a condo, entering information such as name, address and chose the correct condo administration that manages it.

Rationale

This function will make the condo available to the system in order to create the units that belong to that condo. That will be necessary to import the debts and keep an organized record.

2.3.5 Edit Condo**Description**

The manager edits a condo, entering information such as name, address and chose the correct condo administration that manages it.

Rationale

This function will make possible to edit the condo information whenever the data's changed or it was entered incorrectly. That's necessary to keep an organized record.

2.3.6 Remove Condo**Description**

The manager removes a condo from the system.

Rationale

This function will make possible to remove the condo from the system whenever the client decides it won't manage the debts from that condo anymore. That's necessary to keep an organized record.

2.3.7 Create Unit**Description**

The manager creates an unit, entering information such as number, building name, select the correct condo that it belongs to, and the current tenant .

Rationale

This function will make the unit available to the system in order to assign the tenants and debts for that unit. That will be necessary to keep an organized record.

2.3.8 Edit Unit**Description**

The manager edits an unit, entering information such as number, building name, the correct condo that it belongs to, and the current tenant.

Rationale

This function will make possible to edit the unit information whenever the data's changed or it was entered incorrectly. That's necessary to keep an organized record.

2.3.9 Remove Unit**Description**

The manager removes an unit from the system.

Rationale

This function will make possible to remove the unit from the system whenever the client decides it won't manage the debts from that unit anymore. That's necessary to keep an organized record.

2.3.10 Create Tenant**Description**

The manager creates a tenant, entering information such as name, billing address, SSN, phone number.

Rationale

This function will make the tenant available to the system in order to manage his debts. That will be necessary to keep an organized record.

2.3.11 Edit Tenant**Description**

The manager edits a tenant, entering information such as name, billing address, SSN, phone number.

Rationale

This function will make possible to edit the tenant information whenever the data's changed or it was entered incorrectly. That's necessary to keep an organized record.

2.3.12 Remove Tenant**Description**

The manager removes a tenant from the system.

Rationale

This function will make possible to remove the tenant from the system whenever he's not needed anymore. That's necessary to keep an organized record.

2.3.13 Create Debt**Description**

The manager creates a debt, entering information such as the unit in debt, description, original amount, due date, debt type (condo fee or extra fee).

Rationale

This function will make the debt available to the system in order to emit notices for it. That will be necessary to keep an organized record.

2.3.14 Edit Debt**Description**

The manager edit a debt, entering information such as the unit in debt, description, original amount, due date, debt type (condo fee or extra fee).

Rationale

This function will make possible to edit the debt information whenever the data's changed or it was entered incorrectly. That's necessary to keep an organized record.

2.3.15 Remove Debt**Description**

The manager removes a debt from the system.

Rationale

This function will make possible to remove the debt from the system whenever it's not needed anymore or was incorrectly inserted in the system. That's necessary to keep an organized record.

2.3.16 Create Notice**Description**

The manager creates a notice, entering information such as the units that need to be notified and for which debts they are being notified.

Rationale

This function will make the notice available to the system in order to create the notice to be printed together with the envelope labels. That will be necessary to keep an organized record.

2.3.17 Edit Notice**Description**

The manager edits a notice, entering information such as the units that need to be notified and for which debts they are being notified.

Rationale

This function will make possible to edit the notice information whenever the data's changed or it was entered incorrectly. That's necessary to keep an organized record.

2.3.18 Remove Notice**Description**

The manager removes a notice from the system.

Rationale

This function will make possible to remove the notice from the system whenever it's not needed anymore or was incorrectly inserted in the system. That's necessary to keep an organized record.

2.3.19 Import Monthly Report**Description**

The manager imports the monthly debt report provided by each client. The report contains the expanded outstanding debt for each unit for all condos managed by the client (Condo Administration Company).

Rationale

This function will make possible to import the monthly report data into the system in order to reconcile the system data with the outstanding debts and provide the option to emit notices for those debts.

2.3.20 Emit Notices**Description**

The manager selects which units and debts to emit notices and the system generates notice letters and envelope labels.

Rationale

This function will make possible to inform a tenant of the outstanding debt and provide a chance for him to settle the debt before being sued or evicted.

3 Other Nonfunctional Requirements

3.1 Performance Requirements

The system response time shall not exceed 10 seconds for any user interaction event.

3.2 Reliability Requirements

The application's mean time between failures shall be at least 1 month.

3.3 Robustness Requirements

The application shall provide a meaningful error message and continue to operate properly when a human user provides incorrect inputs.

3.4 Maintainability Requirements

The average person-time required to make a minor enhancement (including testing and documentation update) shall not exceed one person week.

3.5 Usability Requirements

The application's user interface shall be clear to its users. Specifically, at least 70% of a statistically valid sample of users shall rate the overall user interface as being either very clear or clear on the following scale (very clear, clear, neutral, unclear, very unclear).

4 Constraints

4.1 Design

The performance of the system depends on the underlying server and the database system. Web page scaling depends on the device and the browser used.

4.2 Implementation

The system functionality depends on the ability to reliably transfer data over an active Internet connection. If the connection has high latency or bandwidth issues then the response time may be slower than desired.

Bibliography

- [1] “Ieee recommended practice for software requirements specifications,” *IEEE Std 830-1998*, 1998.