

# **Homeowners of America: Design and Design Test Plan Report**

Team 1: Tyler Dobbs, Casey Galloway, and Matt McJury

11/3/2015

**CIS 317.01**

**Analysis and Logical Design of Information  
Systems**

## Table of Contents

Introduction .....	2
Interfaces .....	2
Interface Structure Diagram.....	3
Interface Specifications .....	4
Screen Layouts .....	4 - 21
Report Layouts .....	21 - 32
Program Structure and Specifications .....	33
Structure Chart .....	33 - 44
HOA Structure Chart Page 1 Module Pseudocode .....	45 - 56
HOA Structure Chart Page 2 Module Pseudocode .....	56 - 67
HOA Structure Chart Page 3 Module Pseudocode .....	67 - 70
HOA Structure Chart Page 4 Module Pseudocode .....	70 - 77
HOA Structure Chart Page 5 Module Pseudocode .....	77 - 84
HOA Structure Chart Page 6 Module Pseudocode .....	84 - 90
HOA Structure Chart Page 7 Module Pseudocode .....	90 - 99
HOA Structure Chart Page 8 Module Pseudocode .....	99 - 108
HOA Structure Chart Page 9 Module Pseudocode .....	108 - 116
HOA Structure Chart Page 10 Module Pseudocode .....	116 - 124
HOA Structure Chart Page 11 Module Pseudocode .....	124 - 84
Appendix A – Interface Specifications	
Appendix B – Interface Test Plans	

# **Homeowners of America: Design and Design Test Plan Report**

Tyler Dobbs, Casey Galloway, and Matt McJury

## **Introduction**

Homeowners of America is an organization dedicated to handling the financial and administrative affairs of various homeowners associations all over the country. They have requested that we design and implement an information system for them that would increase their productivity and allow them to grow as a company and take in more and more associations to manage. Since we have already determined and understand the requirements for Homeowners of America, we are now able to move onto the next step in creating their system which is designing the system itself. This part of the process includes designing screen and report layouts as well as writing module-specific pseudocode.

## **Interfaces**

Interfaces are the first step to completing the design process when creating a system. In short, they are the screens and reports that allow for the user to interact with the system. This part of the process consists of creating layouts for the various screens and reports along with writing detailed specifications for each of the interfaces created. Defining how the interfaces work together is also included in this part of the process.

## **Interface Structure Diagram**

The interface structure diagram is a visual representation of how the all of the interfaces are related to each other and shows what processes from the requirements contribute to each interface. Figure 1 is the interface structure diagram for the system we designed for Homeowners of America. This structure diagram displays the screens and reports within our system and how the user is able to move from one interface to another. It also roughly shows how our interfaces are laid out in a tab fashion.

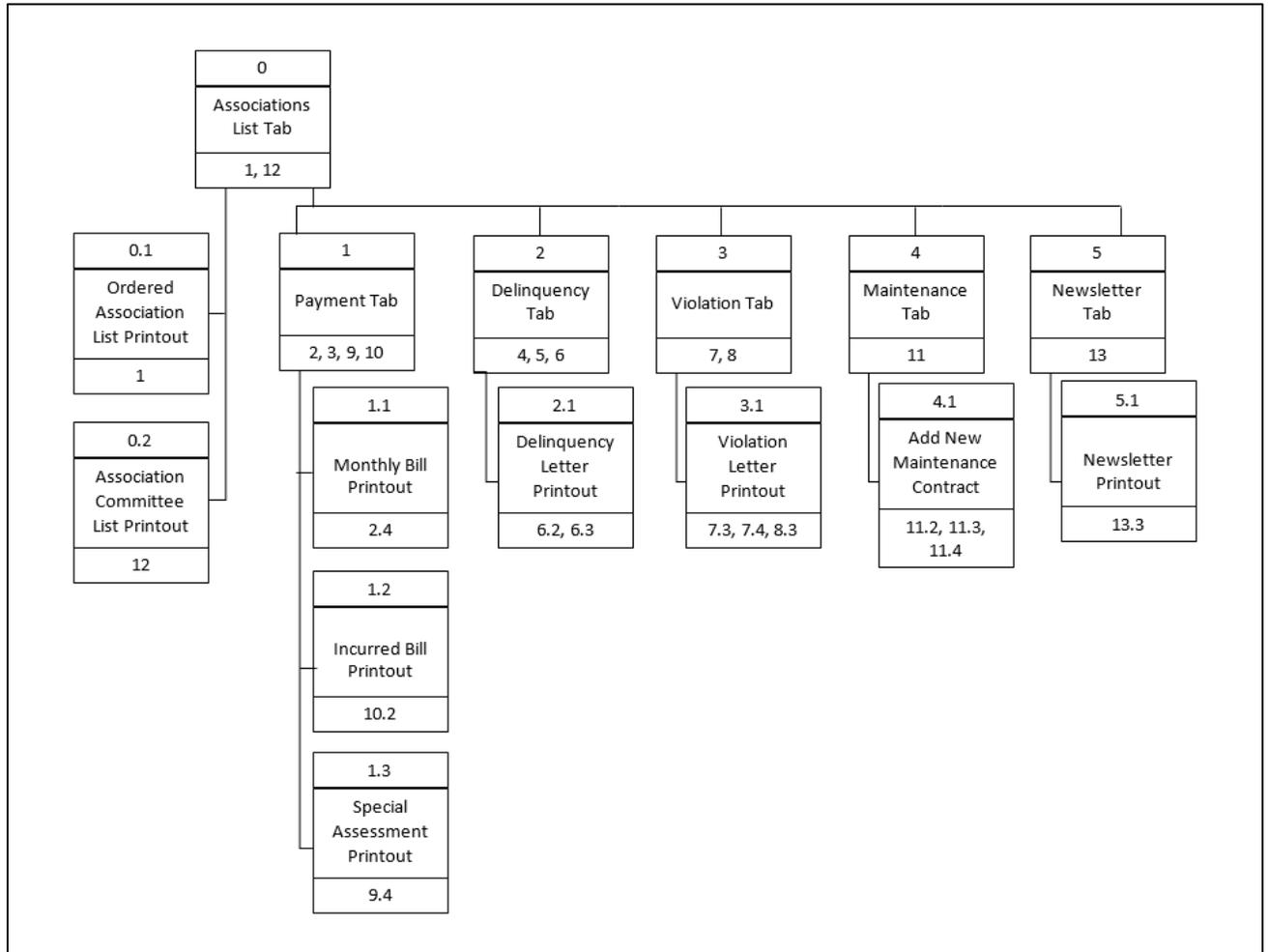


Figure 1: Homeowners of America Interface Structure Diagram

## Interface Specifications

Interface specifications are documents that provide detailed explanations about each particular interface. The specifications enable future designers to know exactly what someone was thinking when they created the interface originally. Specifications normally consist of a detailed narrative that explains who will use the interface, the purpose of the interface, the data flows involved, the fields to be fill-in by either the system or the user, a list of interface buttons, etc. There is also a spot to display an interface prototype and another section where a person writes out a brief test plan for the interface. Appendix A displays the specifications for our interfaces.

### **Screen Layouts**

Layouts allow a person to visualize what the interface is, hopefully, going to look like at the very end. The screen layouts provide an example of how each interface is going to appear on the computer screen. Due to the length of some of the screen layouts a smaller scale figure of the entire will be followed by the layout being broken down into various sections.

#### ***Associations List Tab Screen Layout***

This is a layout of the first tab or screen in our system. This screen incorporates various aspects that are associated with the homeowner associations managed by Homeowners of America and their corresponding association committees. This layout also provides a section where a new association can be added into the system. Figure 2 is the screen layout in its entirety while Figures 3 and 4 are section break-downs of the same screen.

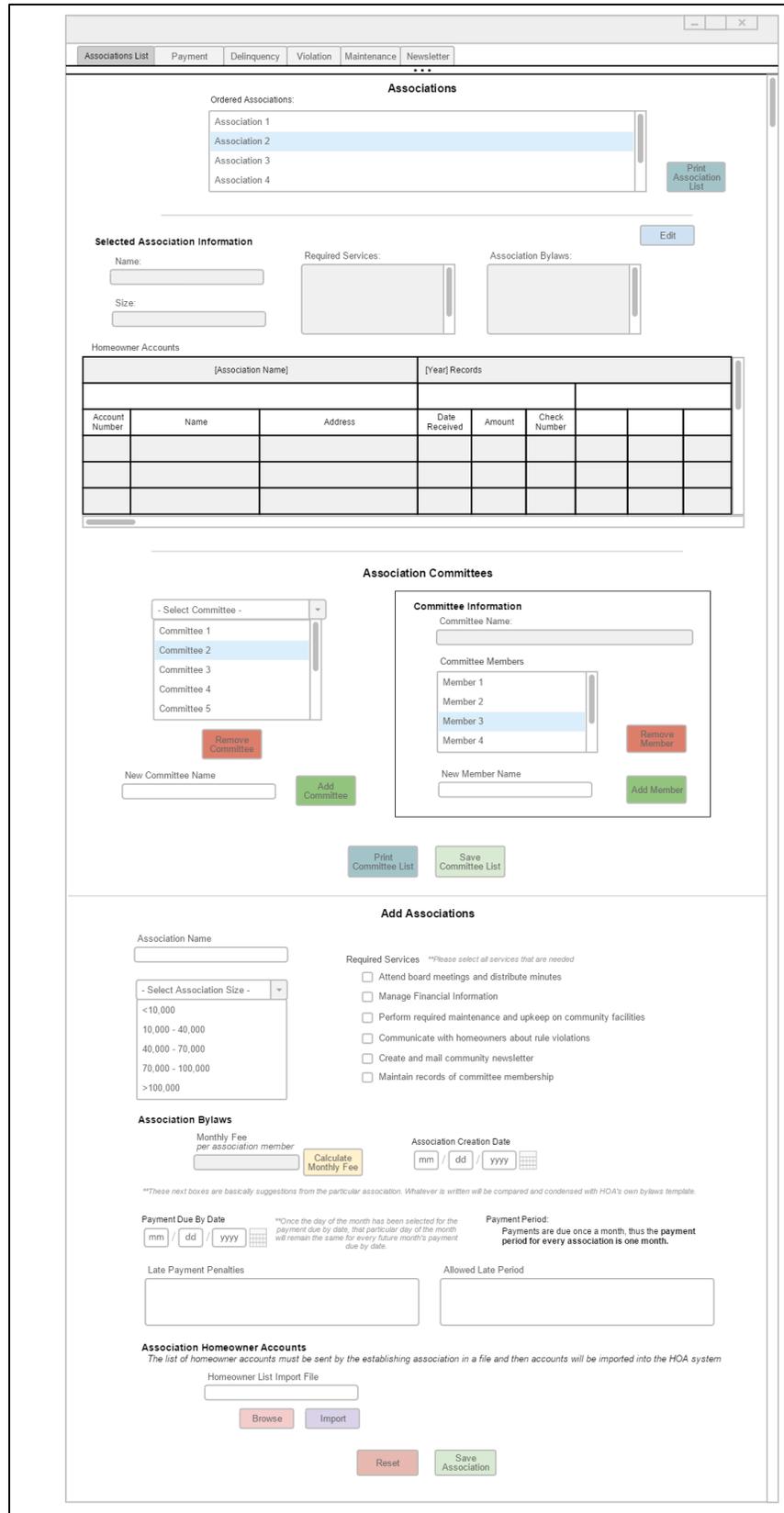


Figure 2: Associations List Tab Screen Layout

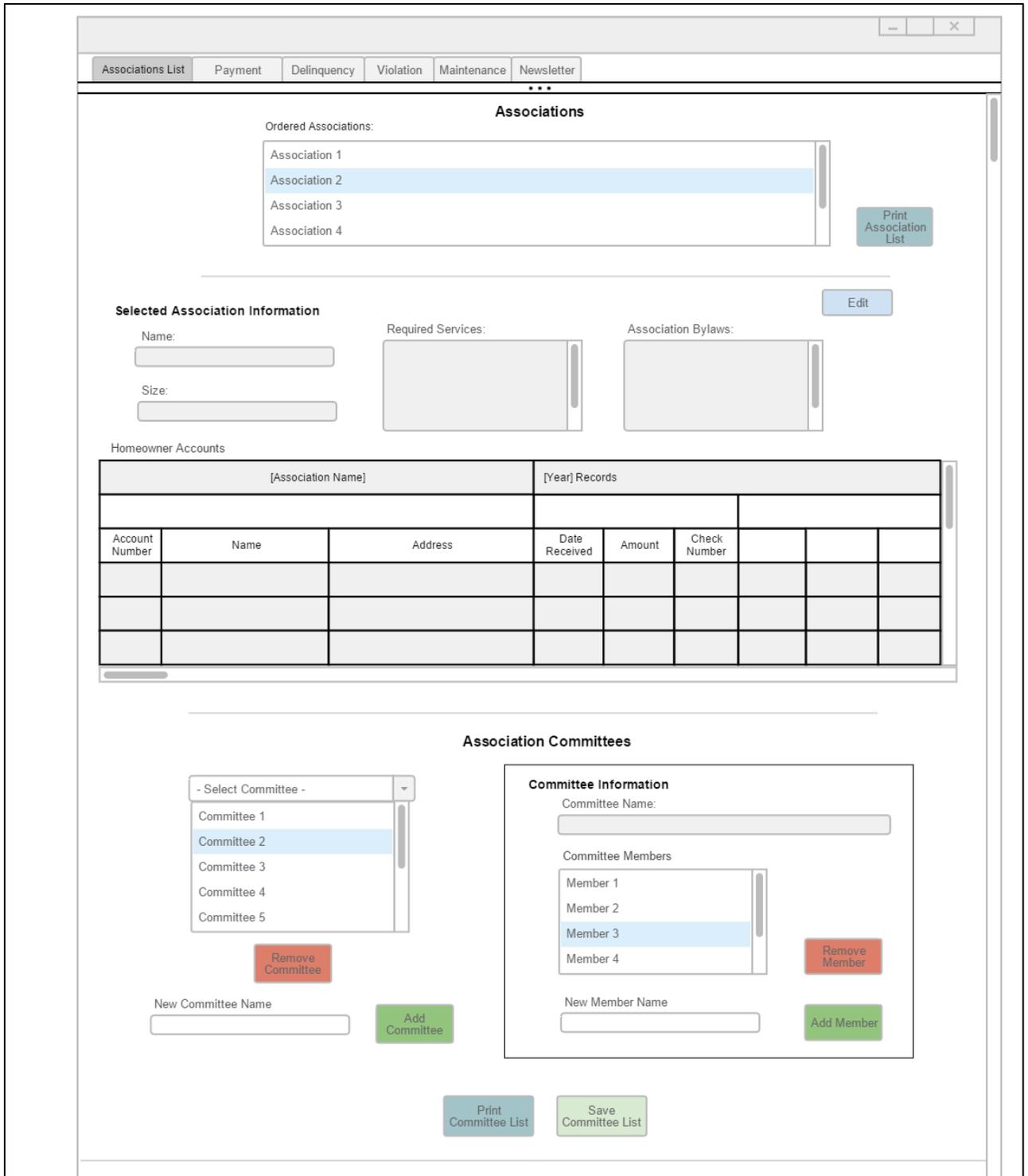


Figure 3: Associations List Tab - Existing Association Sections Screen Layout

### Add Associations

Association Name

- Select Association Size -

**Required Services** \*\*Please select all services that are needed

- Attend board meetings and distribute minutes
- Manage Financial Information
- Perform required maintenance and upkeep on community facilities
- Communicate with homeowners about rule violations
- Create and mail community newsletter
- Maintain records of committee membership

**Association Bylaws**

Monthly Fee  
per association member

Calculate  
Monthly Fee

Association Creation Date

 /  /

\*\*These next boxes are basically suggestions from the particular association. Whatever is written will be compared and condensed with HOA's own bylaws template.

Payment Due By Date

 /  /

\*\*Once the day of the month has been selected for the payment due by date, that particular day of the month will remain the same for every future month's payment due by date.

Payment Period:  
Payments are due once a month, thus the payment period for every association is one month.

Late Payment Penalties

Allowed Late Period

**Association Homeowner Accounts**  
The list of homeowner accounts must be sent by the establishing association in a file and then accounts will be imported into the HOA system

Homeowner List Import File

Figure 4: Associations List Tab - Add New Association Section Screen Layout

**Payment Tab Screen Layout**

This is a layout of the second tab or screen in our system. This screen enables Homeowners of America to efficiently and easily manage all financial information for the homeowner associations they manage or for themselves. This layout allows for the creation of different types of bills and also the processing of any payments made to them. Figure 5 is the screen layout in its entirety while Figures 6 – 8 are section break-downs of the same screen.

Associations List
Payment
Delinquency
Violation
Maintenance
Newsletter

Monthly Bill

**Create Monthly Bill**

Association Name

- Select Month -

---

**Process Bill Payment**

Date Received

mm

/

dd

/

yyyy

Payment Amount

Member Name

Account Number

Member Association

Check Number

Check if no Payment Coupon

Account Found:

Member Name

Member Association

Account Number

---

Incurred Bill

Association Name

**Association Fees**

Water Fee Amount	Trash Pickup Fee Amount
<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>
Gas Fee Amount	Maintenance Fee Amount
<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>
Electricity Fee Amount	HOA Fee Amount
<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>

**Incurred Fee**

Total Fee Amount

---

Special Assessment

**Create Special Assessment Bill**

Association Name

Special Assessment Bill Amount

Date Bill Received

mm

/

dd

/

yyyy

Chosen Due Date

mm

/

dd

/

yyyy

Special Assessment Reason

Charge per Homeowner

---

**Process Special Assessment Payment**

Date Received

mm

/

dd

/

yyyy

Payment Amount

Member Name

Account Number

Member Association

Check Number

Check if no Payment Coupon

Account Found:

Member Name

Member Association

Account Number

Figure 5: Payment Tab Screen Layout

Associations List | Payment | Delinquency | Violation | Maintenance | Newsletter

### Monthly Bill

**Create Monthly Bill**

Association Name [v] | - Select Month - [v]

Create and Print Bill

---

**Process Bill Payment**

Date Received: mm / dd / yyyy [calendar icon]

Payment Amount: [input]

Member Name: [input] | Member Association: [input]

Account Number: [input] | Check Number: [input]

Check if no Payment Coupon

Search Member Name [input] | Search [button]

Account Found:

Member Name: [input]

Member Association: [input]

Account Number: [input]

Reset [button] | Save Payment [button]

Figure 7: Payment Tab - Monthly Bill Section Screen Layout

### Incurred Bill

Association Name [v]

**Association Fees**

Water Fee Amount: [input] | Trash Pickup Fee Amount: [input]

Gas Fee Amount: [input] | Maintenance Fee Amount: [input]

Electricity Fee Amount: [input] | HOA Fee Amount: [input]

**Incurred Fee**

Total Fee Amount: [input] | Calculate Incurred Fee [button]

Reset [button] | Create and Save Bill [button] | Send Bill to HOA [button]

Figure 6: Payment Tab - Incurred Bill Section Screen Layout

### Special Assessment

**Create Special Assessment Bill**

Association Name

Special Assessment Bill Amount

Special Assessment Reason

Date Bill Received  /  /

Chosen Due Date  /  /

Charge per Homeowner

---

**Process Special Assessment Payment**

Date Received  /  /

Payment Amount

Member Name  Member Association

Account Number  Check Number

Check if no Payment Coupon

Account Found:

Member Name

Member Association

Account Number

Figure 8: Payment Tab - Special Assessment Section Screen Layout

### ***Delinquency Tab Screen Layout***

This is the third tab or screen in our system. This screen encompasses all the steps Homeowners of America takes in creating and printing delinquency letters. This layout lets Homeowners of America see the delinquent homeowner accounts while at the same time allowing them to create consolidated delinquency letters. Figure 9 is the screen layout in its entirety.

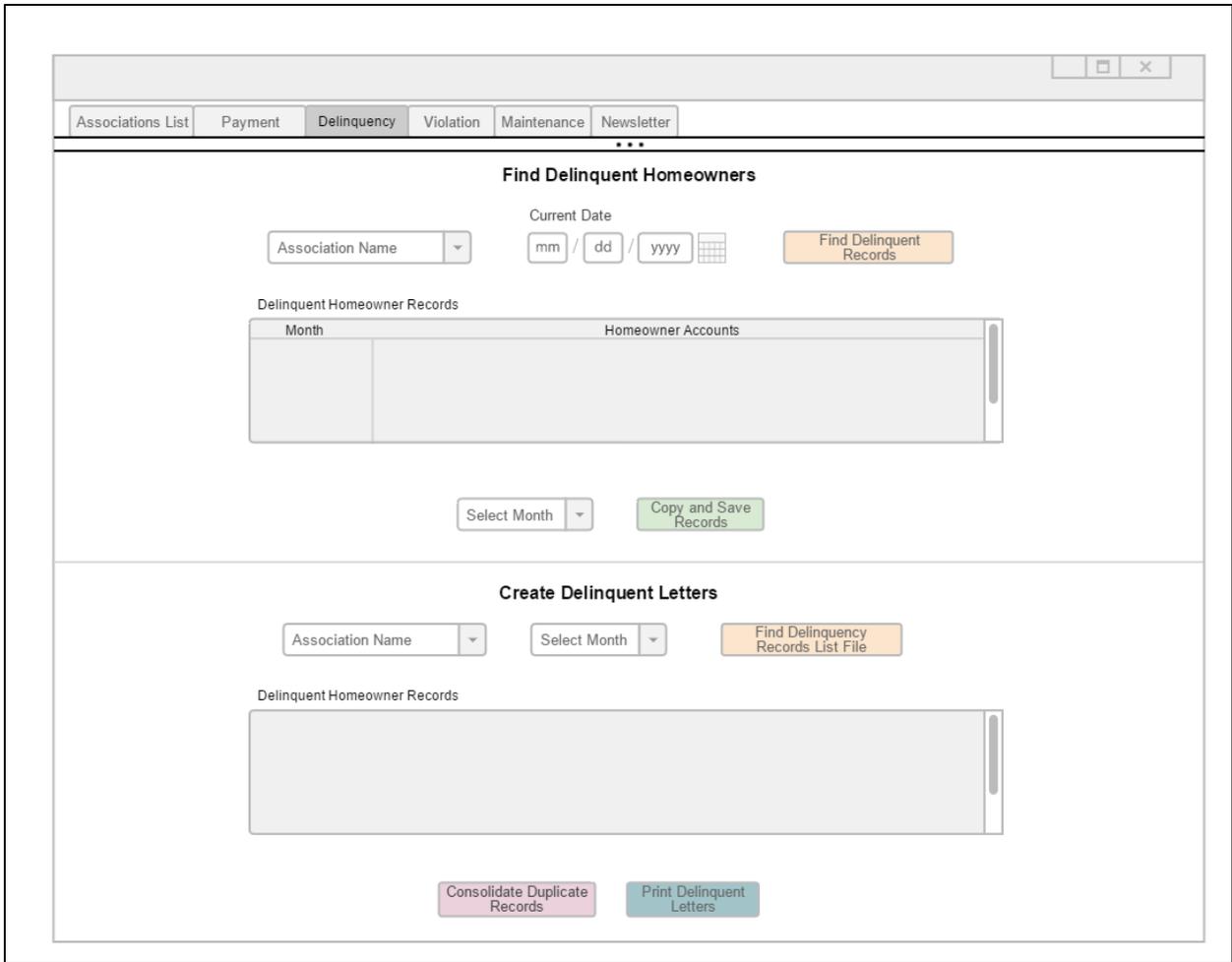


Figure 9: Delinquency Tab Screen Layout

### ***Violation Tab Screen Layout***

The fourth tab or screen in our system is the Violation Tab. This screen enables Homeowners of America to records complaints, make violation notices, and follow-up on those notices. This layout also allows for a complaint file to be edited if need be. Figure 10 is the screen layout in its entirety while Figures 11 - 13 are section break-downs of the same screen.

\_ □ ×
Associations List | Payment | Delinquency | **Violation** | Maintenance | Newsletter

### Complaint

/  /

Valid  
Violated CC&Rs

Invalid  
Reason(s) Complaint is Invalid

---

### Violation Notice

Use Above Complaint Information

Search for a different reported complaint

#### Complaint Recipient Information

**Penalty Warnings**  
These penalty warnings come the Association Board of Directors and can either be manually inputted or imported from an external file.

---

### Notice Tracking

/  /

Old Violation Notices (15 or more days past)

Notice Recipient	Number of Days Past	Notice File

#### Notice Follow-up

**Notice Compliance Status**

Satisfactory Compliance

Non-Satisfactory Compliance

**Follow-up Notice**

Non-Satisfactory Notice File

Non-Compliance Penalties  
These penalties come the Association Board of Directors and can either be manually inputted or imported from an external file.

/  /

Figure 10: Violation Tab Screen Printout

Associations List | Payment | Delinquency | **Violation** | Maintenance | Newsletter

**Complaint**

Association Name

Complaint From  Complaint Recipient  Date Complaint Received  /  /

Complaint

Determined Complaint Validity  Valid  Invalid

Violated CC&Rs  Reason(s) Complaint is Invalid

Figure 11: Violation Tab - Complaint Section Screen Layout

**Violation Notice**

Use Above Complaint Information  Search for a different reported complaint

**Complaint Information**

Association Name  Date Complaint Received  Complaint Validity  Violated CC&Rs

Complaint From  Complaint Recipient

**Complaint Recipient Information**

Name

Mailing Address

**Penalty Warnings**

These penalty warnings come the Association Board of Directors and can either be manually inputted or imported from an external file.

File Import

Figure 12: Violation Tab - Violation Notice Section Screen Layout

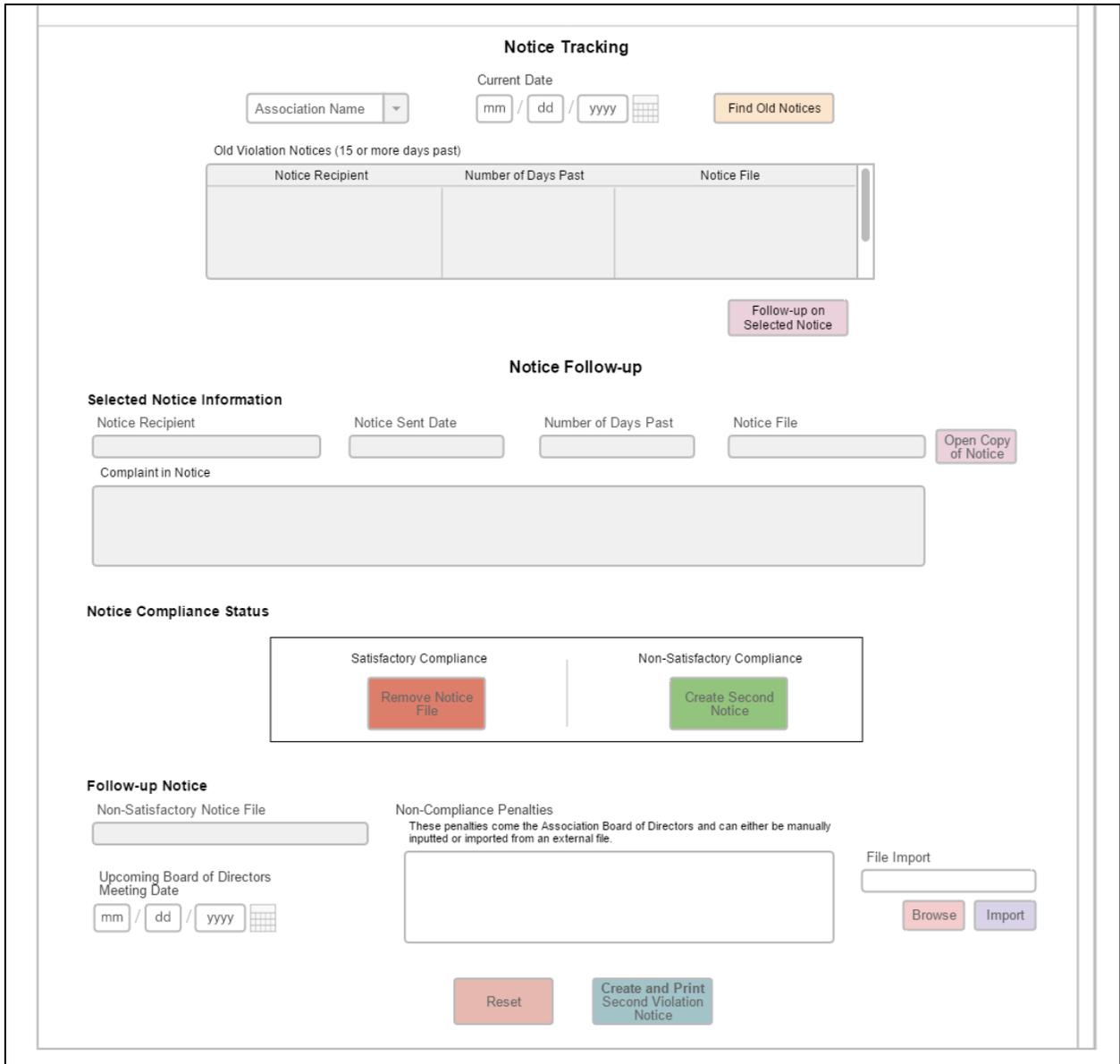


Figure 13: Violation Tab - Notice Tracking and Follow-up Section Screen Layout

**Maintenance Tab Screen Layout**

The Maintenance Tab is the fifth tab or screen in our system. This screen displays an interactive calendar with all of the maintenance contracts on it. It allows for a contract to be chosen from the calendar and edited. This layout also has a section to set a maintenance check-up reminder. There is also a subpage to add a new contract when a certain button is clicked. Figure 14 is the main screen layout in its entirety while Figures

15 – 18 are section break-downs of the same screen. Figure 19 is the screen that appears when then Add New Maintenance Contract button is clicked.

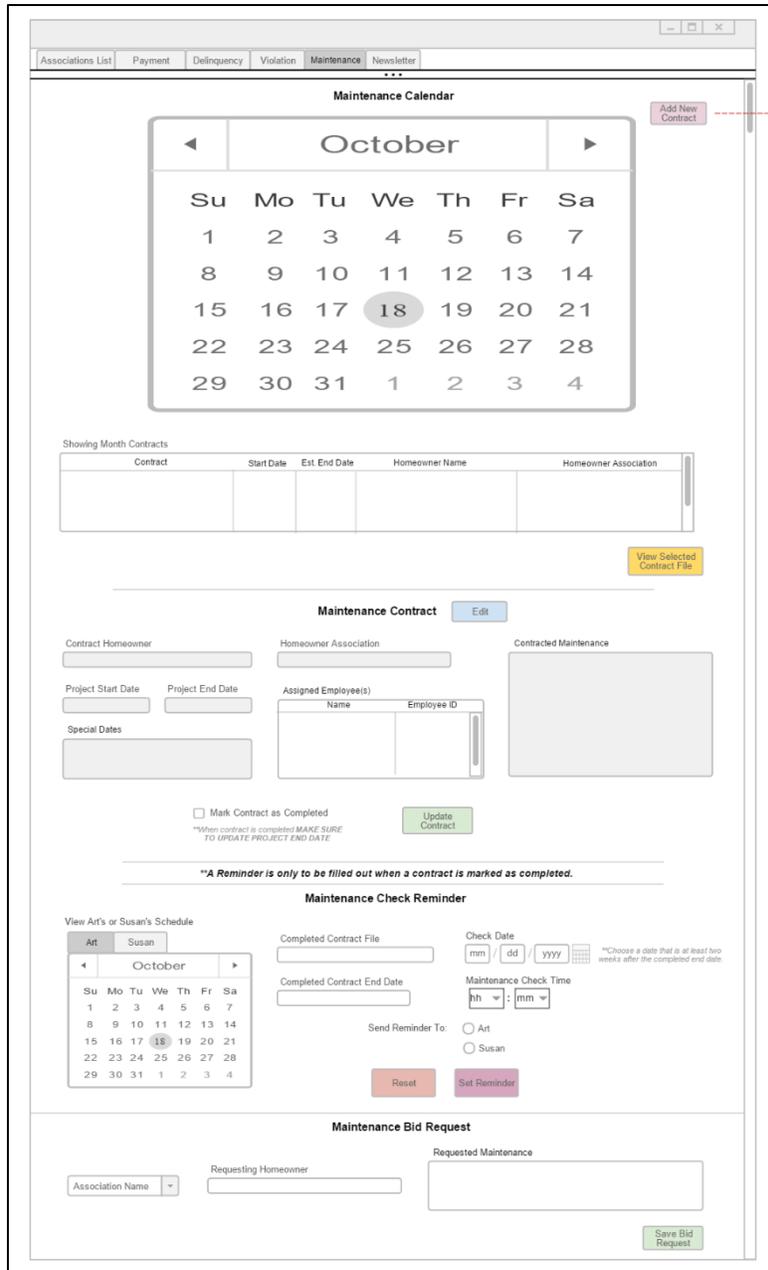


Figure 14: Maintenance Tab Screen Layout

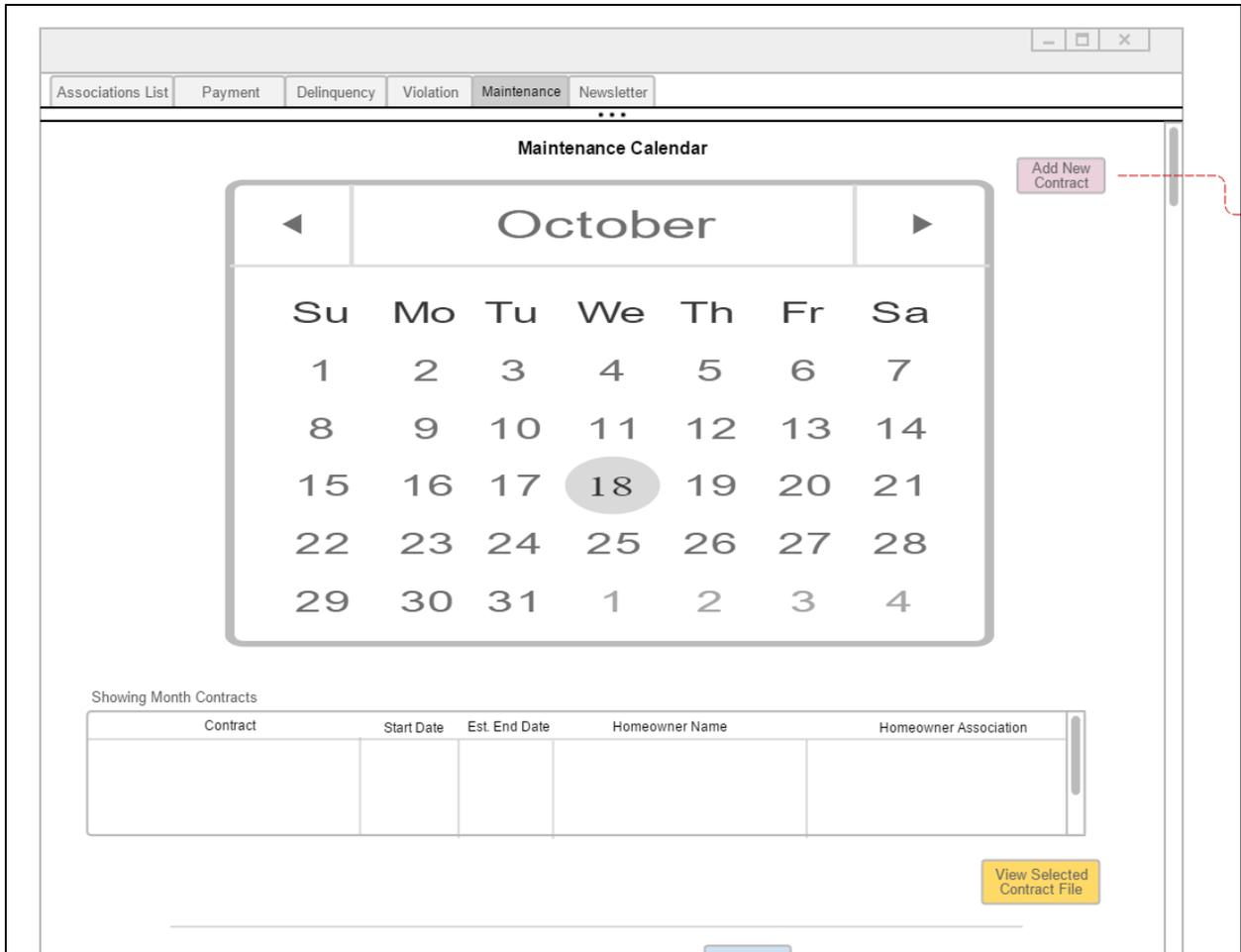


Figure 15: Maintenance Tab - Maintenance Calendar Section Screen Layout

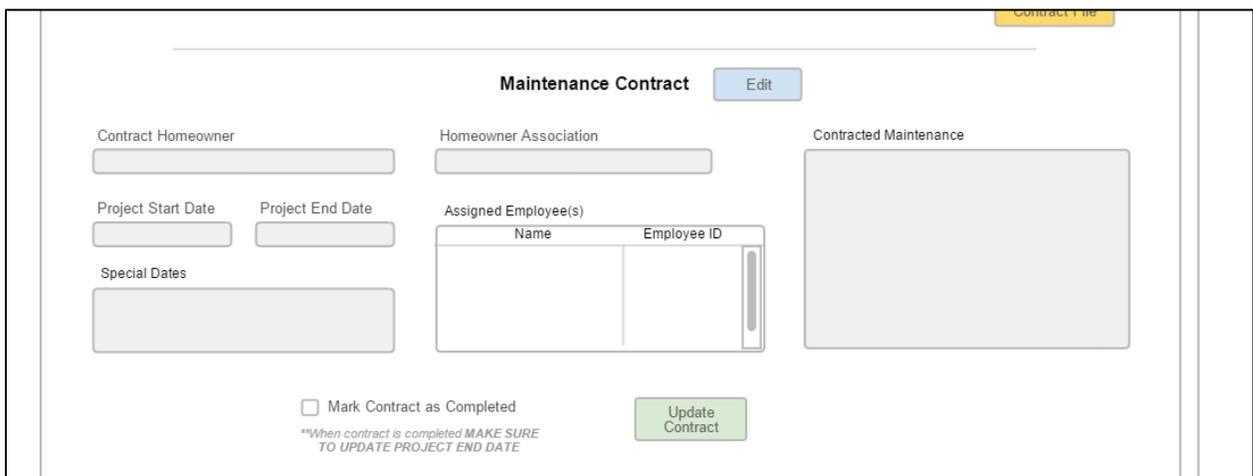


Figure 16: Maintenance Tab - Maintenance Contract Section Screen Layout

**\*\*A Reminder is only to be filled out when a contract is marked as completed.**

### Maintenance Check Reminder

View Art's or Susan's Schedule

Art Susan

October						
Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

Completed Contract File

Check Date  /  /  \*\*Choose a date that is at least two weeks after the completed end date.

Completed Contract End Date

Maintenance Check Time  :

Send Reminder To:  Art  Susan

Figure 17: Maintenance Tab - Maintenance Check Reminder Section Screen Layout

### Maintenance Bid Request

Association Name

Requesting Homeowner

Requested Maintenance

Figure 18: Maintenance Tab - Maintenance Bid Request Section Screen Layout

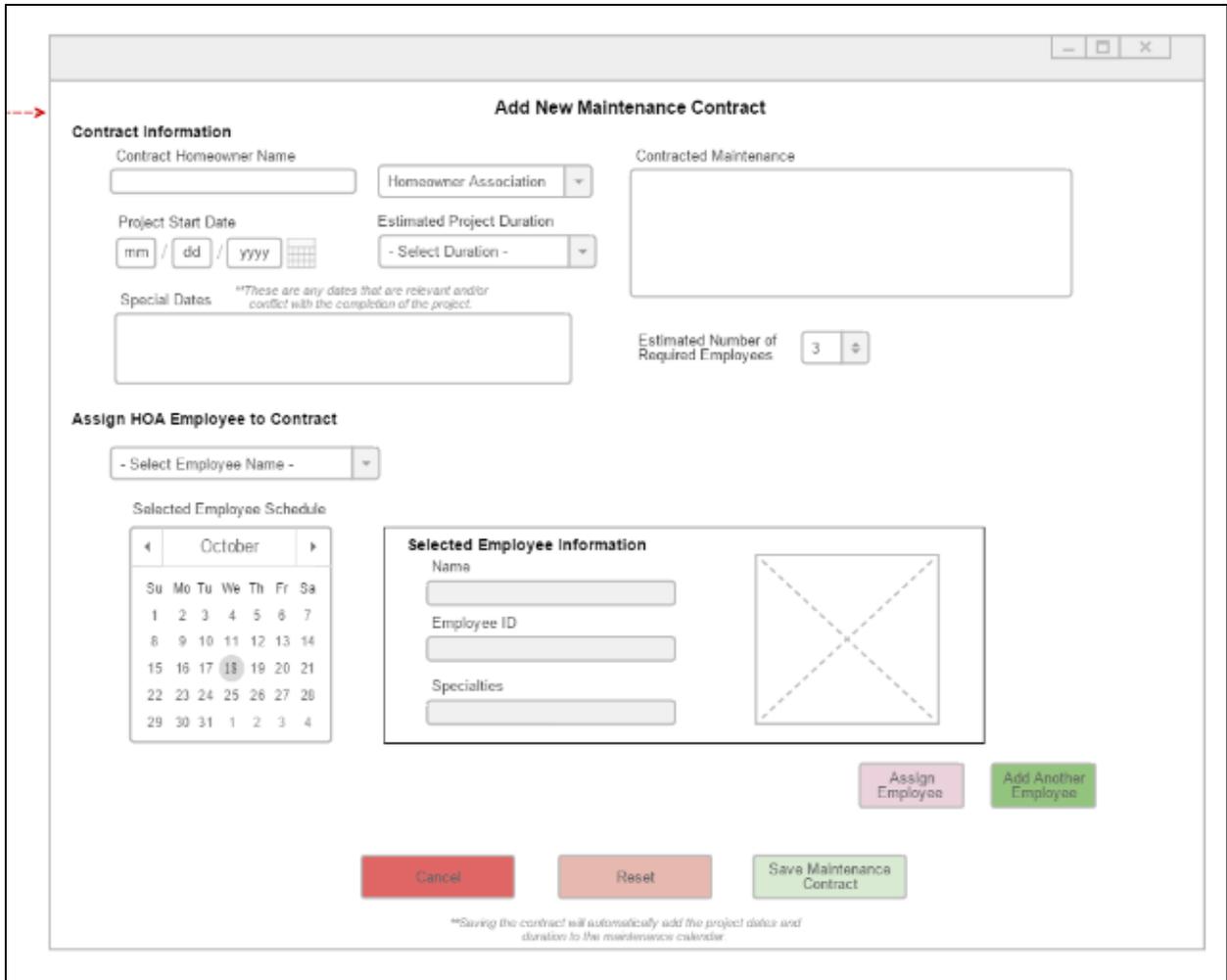


Figure 19: Add New Maintenance Contract Screen Layout

### Newsletter Tab Screen Layout

The last tab or screen in our system is the Newsletter Tab. This screen covers all aspects, according to Homeowners of America’s requirements, that have to deal with creating an association community newsletter. This layout allows for the creation of a newsletter contract, easy access to already existing newsletter contracts, and an upload section for the newsletter after it has been put together by Julie. Figure 20 is the screen layout in its entirety while Figures 21 - 23 are section break-downs of the same screen.

Associations List
Payment
Delinquency
Violation
Maintenance
Newsletter

### Newsletter Contract

**Newsletter Details and Requirements**

*Newsletter Sections (for Layout)*

Community News  
 Community Events  
 Births

Weekly Weather  
 Upcoming Dates  
 Obituaries

Community Activities  
 Birthdays  
 Other  
Please List

Layout Specifications

- Select Newsletter Cycle Duration -

Daily  
Weekly  
 Monthly

Design Specifications \*\*Include things such as preferred color scheme, fonts, sizing, etc.

Additional Comments

---

### Community Newsletter

**Contract Information**

Association

Community

Newsletter Cycle Duration

Newsletter Sections

Layout Specifications

Design Specifications

Additional Comments

**Newsletter Information**

All information that is to be on the newsletter when it prints (including text, images, etc) is to be sent to HOA in a file and then imported into the system.

Section Information File

Image Upload

Imported Files

---

### Newsletter Upload

Finished Newsletter Upload

Figure 20: Newsletter Tab Screen Layout

Associations List | Payment | Delinquency | Violation | Maintenance | **Newsletter**

### Newsletter Contract

Association Name [v] Community (if specified) [v]

**Newsletter Details and Requirements**

Newsletter Sections (for Layout)

- Community News
- Weekly Weather
- Community Activites
- Community Events
- Upcoming Dates
- Birthdays
- Births
- Obituaries
- Other

Please List [input]

- Select Newsletter Cycle Duration - [v]  
Daily  
Weekly  
Monthly

Layout Specifications [text area]

Design Specifications \*\*Include things such as preferred color scheme, fonts, sizing, etc. [text area]

Additional Comments [text area]

Reset Save Contract and Specifications

Figure 21: Newsletter Tab - New Newsletter Contract Section Screen Layout

### Newsletter Upload

Finished Newsletter Upload [input]

Browse Import Print Newsletter Save Newsletter

Figure 22: Newsletter Tab - Newsletter Upload Section Screen Layout

Figure 23: Newsletter Tab - Existing Community Newsletter Contract Section Screen Layout

## Report Layouts

Interface reports are things that are being outputted from the system, whereas the screen layouts are interface forms meaning they get information inputted to be used by the system. The report layouts show an example of what the report is going to look like. All of the reports in this system for Homeowners of America are printed out on paper.

### *Ordered Association List Printout Report Layout*

This report comes from the Associations List Tab screen. It displays all of the homeowner associations managed by Homeowners of America. The associations are ordered based on their association size from largest to smallest. This report also displays

the corresponding association size and required services for each association listed. The more specific details of the report can be found in the interface specifications in Appendix A. Figure 24 shows the general layout how the actual printed information on the report is supposed to appear when it is printed. When the report actually goes through the system all of the bracketed text fields will be filled in with the appropriate information.

<b>Ordered Association List</b>		
<b>Association Name</b>	<b>Size</b>	<b>Required Services</b>
[Association 1]	[Association 1 Size]	[Association 1 Required Services]
[Association 2]	[Association 2 Size]	[Association 2 Required Services]
[Association 3]	[Association 3 Size]	[Association 3 Required Services]
[Association 4]	[Association 4 Size]	[Association 4 Required Services]
[Association 5]	[Association 5 Size]	[Association 5 Required Services]
[Association 6]	[Association 6 Size]	[Association 6 Required Services]
[Association 7]	[Association 7 Size]	[Association 7 Required Services]

*Figure 24: Ordered Association List Printout Report Layout*

***Association Committee List Printout Report Layout***

This report also comes from the Associations List Tab screen. It displays all of the committees within a particular association and the member names of those committees. There is no specific order to how the committees are display. The more specific details of the report can be found in the interface specifications in Appendix A. Figure 25 shows the general layout how the actual printed information on the report is supposed to appear when it is printed. When the report actually goes through the system all of the bracketed text fields will be filled in with the appropriate information.

**Association Committee(s) List**

Association:  [Association Name]

[Committee 1 Name]  
    [List of Member Names]

[Committee 2 Name]  
    [List of Member Names]

[Committee 3 Name]  
    [List of Member Names]

[Committee 4 Name]  
    [List of Member Names]

[Committee 5 Name]  
    [List of Member Names]

[Committee 6 Name]  
    [List of Member Names]

*Figure 25: Association Committee List Printout Report Layout*

**Monthly Bill Printout Report Layout**

This report comes from the Payment Tab screen. It displays the monthly amount due by each homeowner in a particular association. The top half of the report is to be sent back to Homeowners of America with the monthly payment. The more specific details of the report can be found in the interface specifications in Appendix A. Figure 26 shows the general layout how the actual printed information on the report is supposed to appear when it is printed. When the report actually goes through the system all of the bracketed text fields will be filled in with the appropriate information.

<b>Homeowners of America</b>		
P.O. Box 999		
Big City, CA 92222		
800-555-5555		
<b>[Association Name] Homeowners Association</b>		
Monthly Dues Assessment		
[Homeowner Name]		
[Homeowner Address]		
Account [Account Number]		
Date	Charges and Credits	Amount
[Current Date]	[Month] Assessment	[Total amount due]
	Total Due .....	[Total amount due]
Return this portion with your check made payable to:		
[Association Name]		
.....		
Date	Charges and Credits	Amount
[Current Date]	[Month] Assessment	[Total amount due]
	Total Due .....	[Total amount due]
Next Annual Meeting [Next Meeting Date], [Next Meeting Time], [Next Meeting Location]		
[Late penalty] Late Charge if not paid by [Date Due By]		

Figure 26: Monthly Bill Printout Report Layout

***Incurred Bill Printout Report Layout***

This report also comes from the Payment Tab screen. It displays the incurred amount of association fees along with the fees themselves. This report is printed to Homeowners of America because they are responsible for paying the incurred. The more specific details of the report can be found in the interface specifications in Appendix A. Figure 27 shows the general layout how the actual printed information on the report is supposed to appear when it is printed. When the report actually goes through the system all of the bracketed text fields will be filled in with the appropriate information.

<b>Homeowners of America</b>	
P.O. Box 999	
Big City, CA 92222	
800-555-5555	
<b><i>[Association Name]</i> Homeowners Association</b>	
Incurred Bill Assessment	
Association Fees:	
Water Fee .....	<i>[Fee Amount]</i>
Gas Fee .....	<i>[Fee Amount]</i>
Electricity Fee .....	<i>[Fee Amount]</i>
Trash Pickup Fee .....	<i>[Fee Amount]</i>
Maintenance Fee .....	<i>[Fee Amount]</i>
HOA Fee .....	<u><i>[Fee Amount]</i></u>
<b>Total Incurred Amount</b> .....	<i>[Total Incurred Amount]</i>
The total amount due by the Homeowners of America is: <i>[Total Incurred Amount]</i>	

*Figure 27: Incurred Bill Printout Report Layout*

***Special Assessment Printout Report Layout***

This report also comes from the Payment Tab screen, and is very similar to the monthly bill printout. It displays the special charge due by homeowners in a particular association whenever Homeowners of America receives a special assessment fee. Like the monthly bill printout, the top portion is to be sent back to Homeowners of America with a payment. The more specific details of the report can be found in the interface specifications in Appendix A. Figure 28 shows the general layout how the actual printed information on the report is supposed to appear when it is printed. When the report actually goes through the system all of the bracketed text fields will be filled in with the appropriate information.

<b>Homeowners of America</b>		
P.O. Box 999		
Big City, CA 92222		
800-555-5555		
<b><i>[Association Name]</i> Homeowners Association</b>		
Special Assessment		
<i>[Homeowner Name]</i>		
<i>[Homeowner Address]</i>		
Account <i>[Account Number]</i>		
Date	Charges and Credits	Amount
<i>[Current Date]</i>	Special Assessment	<i>[Special Assessment Charge]</i>
	Total Due .....	<i>[Amount due]</i>
Return this portion with your check made payable to:		
<i>[Association Name]</i>		
.....		
Date	Charges and Credits	Amount
<i>[Current Date]</i>	Special Assessment	<i>[Special Assessment Charge]</i>
	Total Due .....	<i>[Amount due]</i>
Next Annual Meeting <i>[Next Meeting Date]</i> , <i>[Next Meeting Time]</i> , <i>[Next Meeting Location]</i>		
<i>[Late penalty]</i> Late Charge if not paid by <i>[Special Assessment Due Date]</i>		

*Figure 28: Special Assessment Printout Report Layout*

### ***Delinquency Letter Printout Report Layout***

This report comes from the Delinquency Tab screen. It informs a homeowner that they are late on payments and displays the delinquent amount that is to be paid by the homeowner. The more specific details of the report can be found in the interface specifications in Appendix A. Figure 29 shows the general layout how the actual printed information on the report is supposed to appear when it is printed. When the report actually goes through the system all of the bracketed text fields will be filled in with the appropriate information.

Homeowners of America  
P.O. Box 999  
Big City, CA 92222

*[Current Date]*

*[Homeowner Name]*  
*[Homeowner Address]*

Re: *[Association Name]*

Dear Homeowner,

It has come to our attention that you are *[Number of Days Late]* days late in the payment of your *[Delinquent Month]* dues. The board of directors has asked that we contact you regarding this matter.

Sections 4.6 and 4.7 of the *[Association Name]* CC&Rs detail your responsibility to make timely payments of your association dues. Section 4.7 specifically details the procedures the association may follow if you do not immediately make all back payments owed to the association. It is your responsibility to immediately address this matter.

Please send in your payment of *[Delinquent amount due]*:  
*[Total amount due]* monthly assessment  
*[Late penalty]* late fee

Please contact me at 800-555-5555 if you have any questions. Note that failure to promptly address this matter may result in the board levying additional fines against you or filing a lien against your property.

Sincerely,

*[Association Manager Name]*  
Association Manager

*Figure 29: Delinquency Letter Printout Report Layout*

### ***Violation Letter Printout Report Layout***

This report comes from the Violation Tab screen. It displays the complaint made against the homeowner, what CC&Rs they are violating, and informs them that they must address the issue quickly. The more specific details of the report can be found in the interface specifications in Appendix A. Figure 30 shows the general layout how the actual printed information on the report is supposed to appear when it is printed. When the report actually goes through the system all of the bracketed text fields will be filled in with the appropriate information.

<p>Homeowners of America P.O. Box 999 Big City, CA 92222</p> <p><i>[Current Date]</i></p> <p><i>[Homeowner Name]</i> <i>[Homeowner Address]</i></p> <p>Re: <i>[Association Name]</i></p> <p>Dear Homeowner,</p> <p>It has come to our attention that you are in violation of the CC&amp;Rs relating to <i>[Violated CC&amp;Rs]</i>. The board of directors has asked that we contact you regarding this matter.</p> <p><i>[Violated CC&amp;Rs specifics]</i>. <i>[Complaint that was reported]</i>. It is your responsibility to immediately address this matter.</p> <p>Please contact me at 800-555-5555 when you have resolved the above stated problem. Note that failure to promptly address this matter may result in the board levying a fine against you, or billing you as a result of the board hiring someone to resolve the issue.</p> <p>Sincerely,</p> <p><i>[Association Manager Name]</i> Association Manager</p>
---

*Figure 30: Violation Notice Printout Report Layout*

**Newsletter Printout Report Layout**

This report comes from the Newsletter Tab screen. It displays whatever information and images were sent to Homeowners of America corresponding with the contract specifications. The more specific details of the report can be found in the interface specifications in Appendix A. Figures 31 - 34 shows an example of what a multi-paged community newsletter may look like. When the report actually goes through the system all of the bracketed text fields will be filled in with the appropriate information.



Figure 31: Newsletter Printout Front Page Report Layout

PAGE 2

## Community Activities



[This story can fit 150-200 words.

One benefit of using your newsletter as a promotional tool is that you can reuse content from other marketing materials, such as press releases, market studies, and reports.

While your main goal of distributing a newsletter might be to sell your product or service, the key to a successful newsletter is making it useful to your readers.

A great way to add useful content to your newsletter is to develop and write your own articles, or include a calendar of upcoming events or a special offer that promotes a new product.

You can also research articles or find "filler" articles by accessing the World Wide Web. You can write about a variety of topics but try to keep your articles short.

Much of the content you put in your newsletter can also be used for your Web site. Microsoft Publisher offers a simple way to convert your newsletter to a Web publication. So, when you're finished writing your newsletter, convert it to a Web site and post it.]

Caption describing picture or graphic.

## Community Events

[This story can fit 100-150 words.

The subject matter that appears in newsletters is virtually endless. You can include stories that focus on current technologies or innovations in your field.

You may also want to note business or economic trends, or make predictions for your customers or clients.

If the newsletter is distributed internally, you might comment upon new procedures or improvements to the business. Sales figures or earnings will show how your business is growing.

Some newsletters include a column that is updated every issue, for instance, an advice column, a book review, a letter from the president, or an editorial. You can also profile new employees or top customers or vendors.]

Caption describing picture or graphic.

## Welcome New Community Members



[This story can fit 75-125 words.

Selecting pictures or graphics is an important part of adding content to your newsletter.

Think about your article and ask yourself if the picture supports or enhances the message you're trying to convey. Avoid selecting images that appear to be out of context.

Microsoft Publisher includes thousands of clip art images from which you can choose and import into your newsletter. There are also several tools you can use to draw shapes and symbols.

Once you have chosen an image, place it close to the article. Be sure to place the caption of the image near the image.]

[NEWSLETTER TITLE]

Figure 32: Newsletter Printout Page 2 Report Layout

[COMMUNITY NAME] PAGE 3

## Birthdays to Celebrate

[This story can fit 150-200 words.]

One benefit of using your newsletter as a promotional tool is that you can reuse content from other marketing materials, such as press releases, market studies, and reports.

While your main goal of distributing a newsletter might be to sell your product or service, the key to a successful newsletter is making it useful to your readers.

A great way to add useful content to your newsletter is to develop and write your own articles, or include a calendar of upcoming events or a special offer that promotes a new product.

You can also research articles or find "filler" articles by accessing the World Wide Web. You can write about a variety of topics but try to keep your articles short.

Much of the content you put in your newsletter can also be used for your Web site. Microsoft Publisher offers a simple way to convert your newsletter to a Web publication. So, when you're finished writing your newsletter, convert it to a Web site and post it.]

*"To catch the reader's attention, place an interesting sentence or quote from the story here."*

## Special and Upcoming Dates

[This story can fit 100-150 words.]

The subject matter that appears in newsletters is virtually endless. You can include stories that focus on current technologies or innovations in your field.

You may also want to note business or economic trends, or make predictions for your customers or clients.

If the newsletter is distributed internally, you might comment upon new procedures or improvements to the business. Sales figures or earnings will show how your business is growing.

Some newsletters include a column that is updated every issue, for instance, an advice column, a book review, a letter from the president, or an editorial. You can also profile new employees or top customers or vendors.]

## Other Section Header

[This story can fit 75-125 words.]

Selecting pictures or graphics is an important part of adding content to your newsletter.

Think about your article and ask yourself if the picture supports or enhances the message you're trying to convey. Avoid selecting images that appear to be out of context.

Microsoft Publisher includes thousands of clip art images from which you can choose and import into your newsletter. There are also several tools you can use to draw shapes and symbols.

Once you have chosen an image, place it close to the article. Be sure to place the caption of the image near the image.]



Caption describing picture or graphic.

Figure 33: Newsletter Printout Page 3 Report Layout

Homeowners of America

P.O. Box 999  
Big City, CA 92222

Phone: 800-555-5555  
Fax: 800-555-5544  
Email: someone@hoa.com

This is the weekly newsletter for the  
[Community Name] of [Association Name].

**\*\*Newsletters are subject to change based on  
the information provided by the home associa-  
tion.\*\***



## Obituaries

[This story can fit 175-225 words. If your newsletter is folded and mailed, this story will appear on the back. So, it's a good idea to make it easy to read at a glance.]

A question and answer session is a good way to quickly capture the attention of readers. You can either compile questions that you've received since the last edition or you can summarize some generic questions that are frequently asked about your organization.

A listing of names and titles of managers in your organization is a good way to give your newsletter a personal touch. If your organization is small, you may want to list the names of all employees.

If you have any prices of standard products or services, you can include a listing of those here. You may want to refer your readers to any other forms of communication that you've created for your organization.

You can also use this space to remind readers to mark their calendars for a regular event, such as a breakfast meeting for vendors every third Tuesday of the month, or a biannual charity auction. If space is available, this is a good place to insert a clip art image or some other graphic.]



Caption describing picture or graphic.

Figure 34: Newsletter Printout Last Page Report Layout

## **Program Structure and Specifications**

The program structure and specifications essentially creates instructions for programmers when the system is actually sent to be programmed. These programmer instructions consist of a structure chart which displays the entire system broken down into modules, pseudocode for each module that was created for the structure chart, and test plans for every module specification pseudocode along with an integrative test plan to make sure all the module specification pseudocodes work together.

### **Structure Chart**

The structure chart plays an important role which designing a system because it shows all of the code needed for the system broken down into simplified modules that are displayed in a hierarchical format. Since the structure chart encompasses all of the code for a system it can get very large very fast meaning that it usually has to be split up and separated onto multiple pages. Figures 35 – 41 shows the module breakdown of the system we designed for Homeowners of America and they also show how our structure chart was spread across multiple pages. There is also a key within each structure chart page in order to easily identify what the inputs and outputs of each module are.



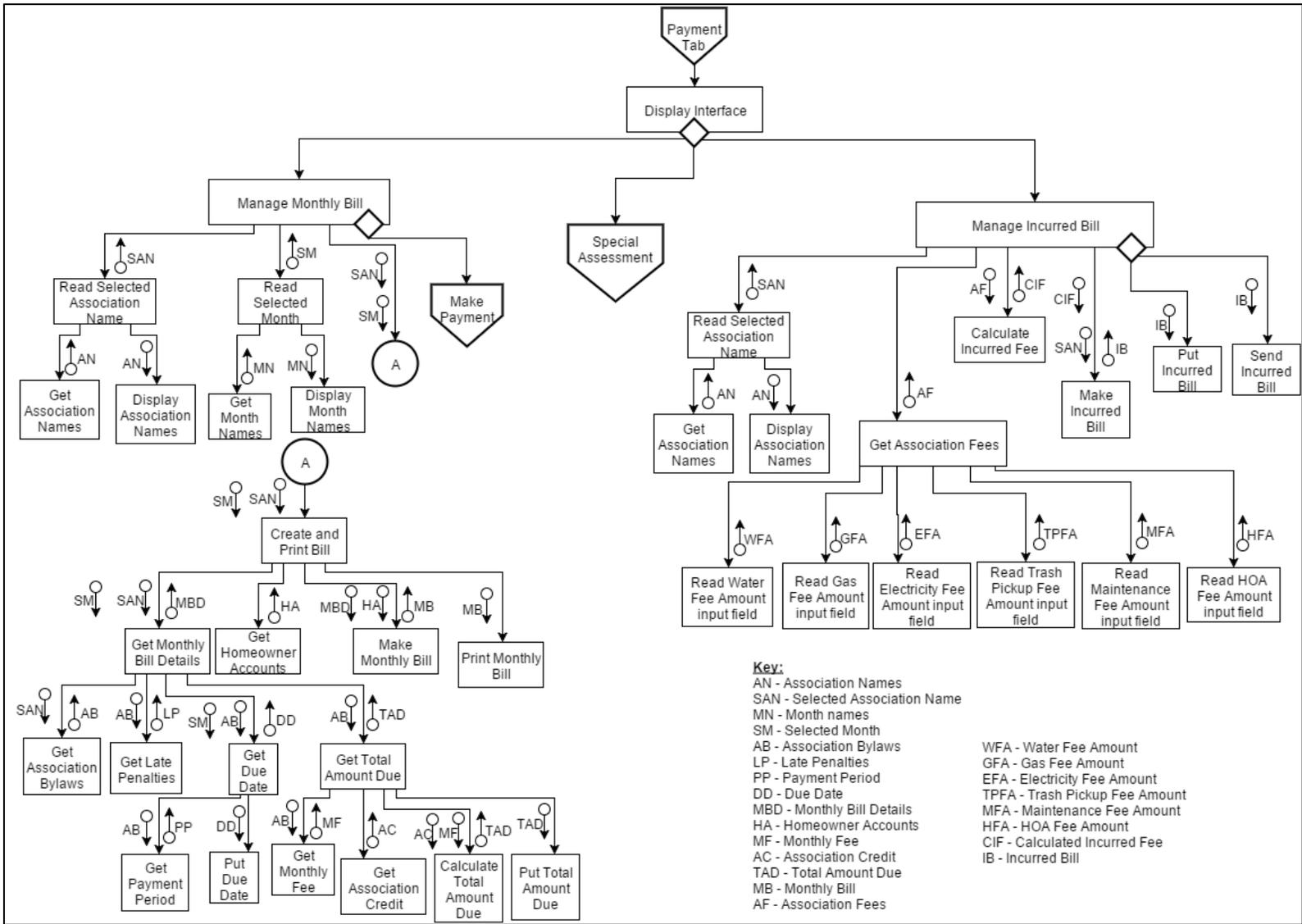


Figure 36: HOA Structure Chart Page 2 - Payment Tab Modules

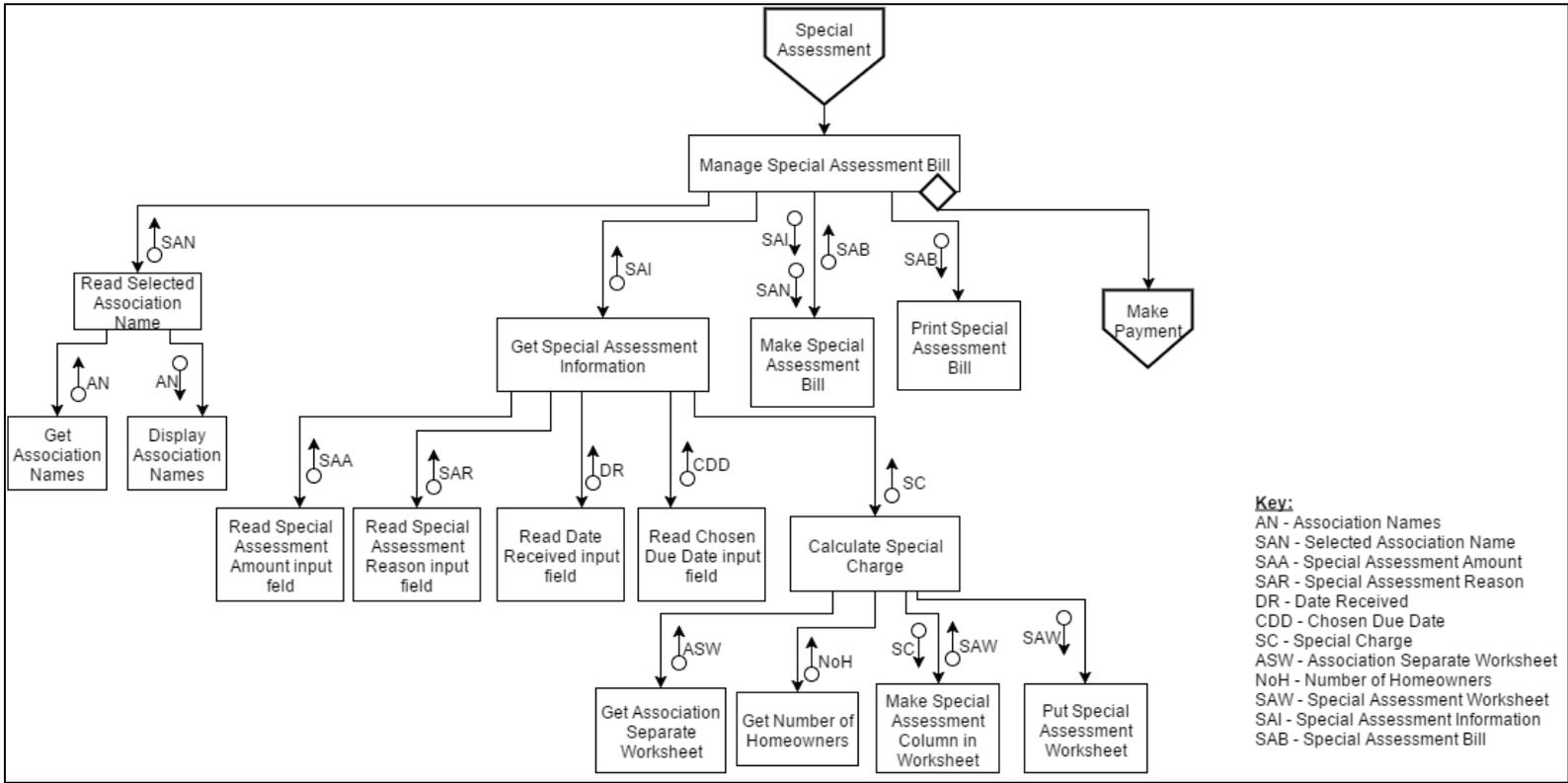


Figure 37: HOA Structure Chart Page 3 - Special Assessment Off-Page Connector Modules

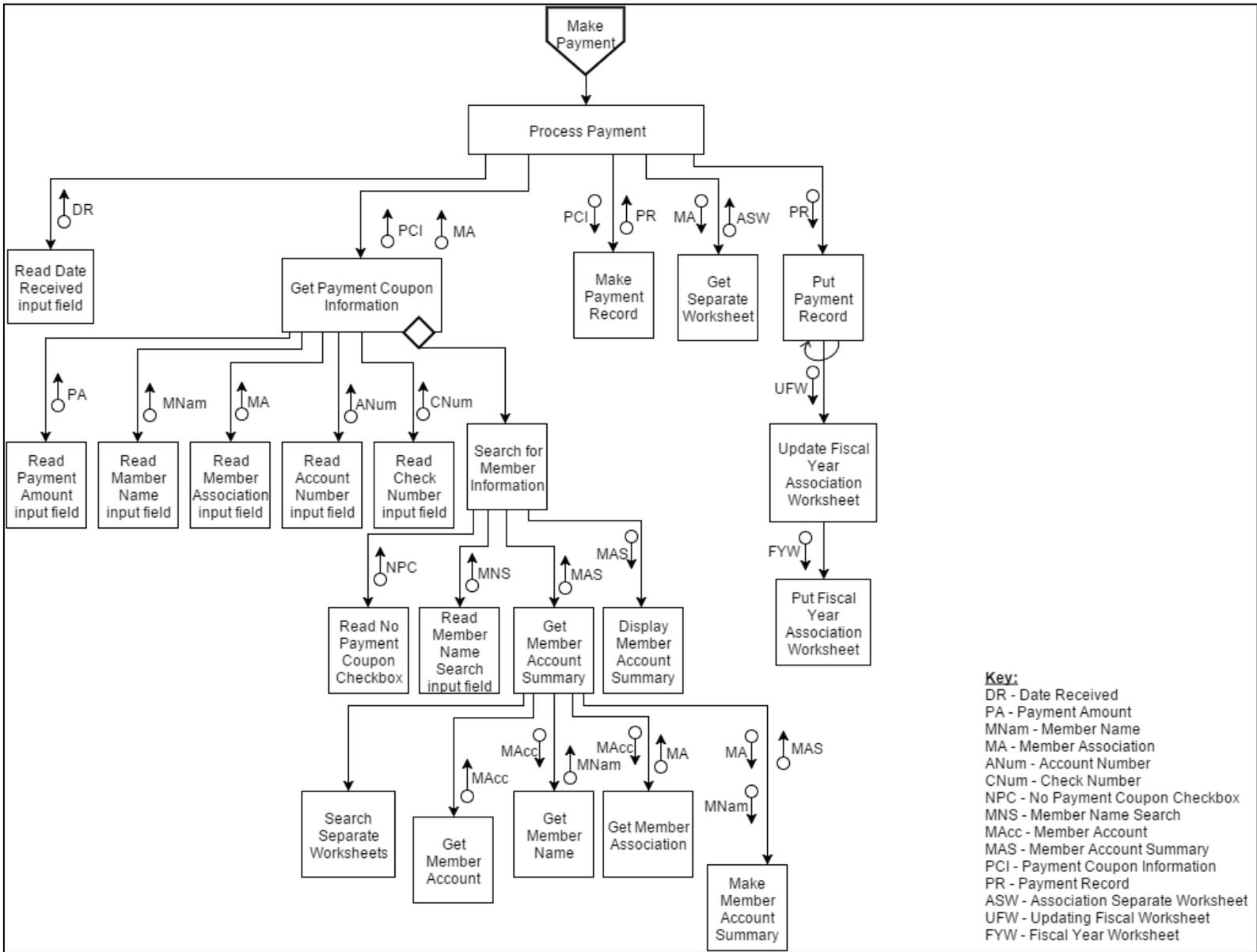


Figure 38: HOA Structure Chart Page 4 - Make Payment Off-Page Connector Modules

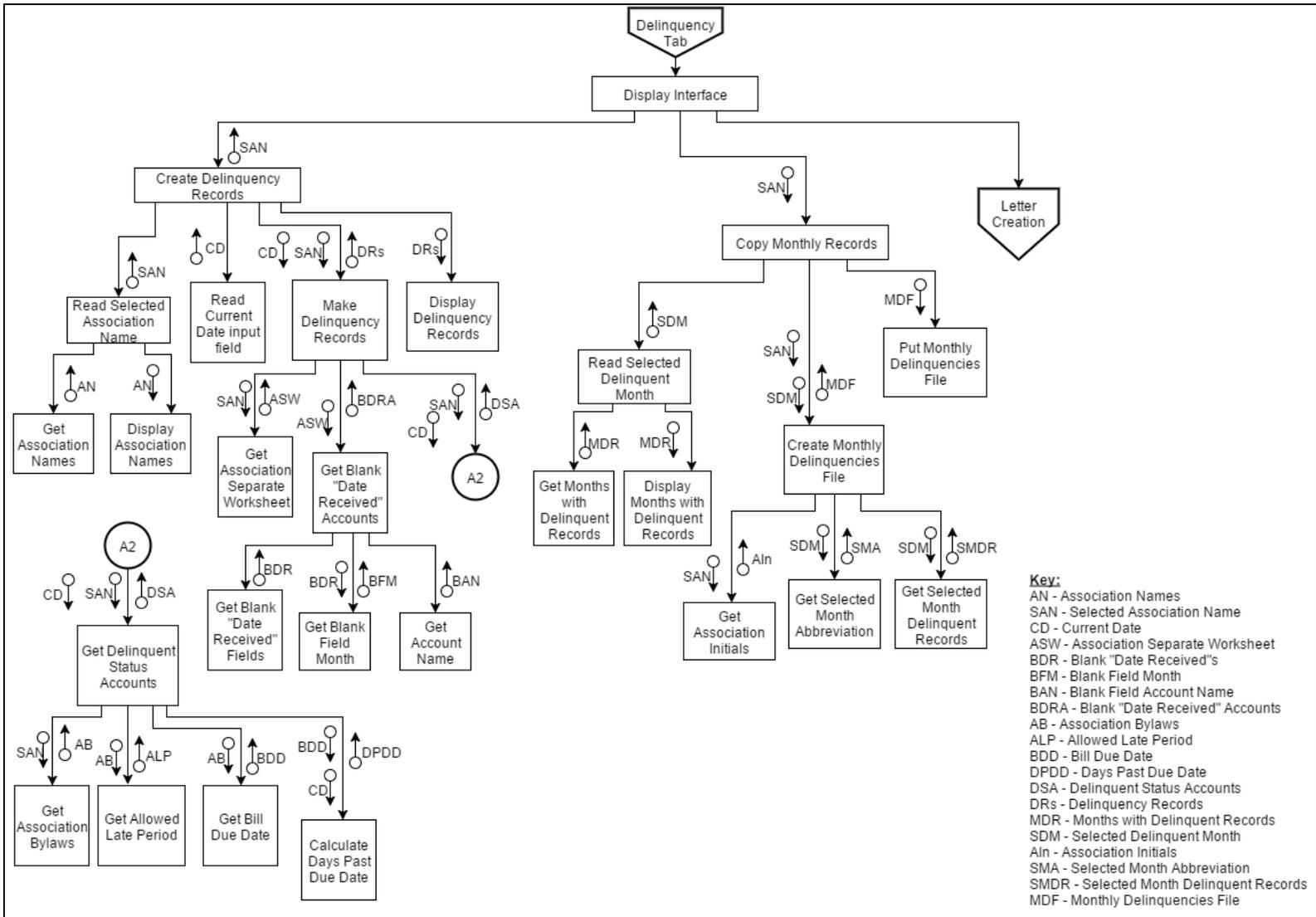


Figure 39: HOA Structure Chart Page 5 - Delinquency Tab Modules

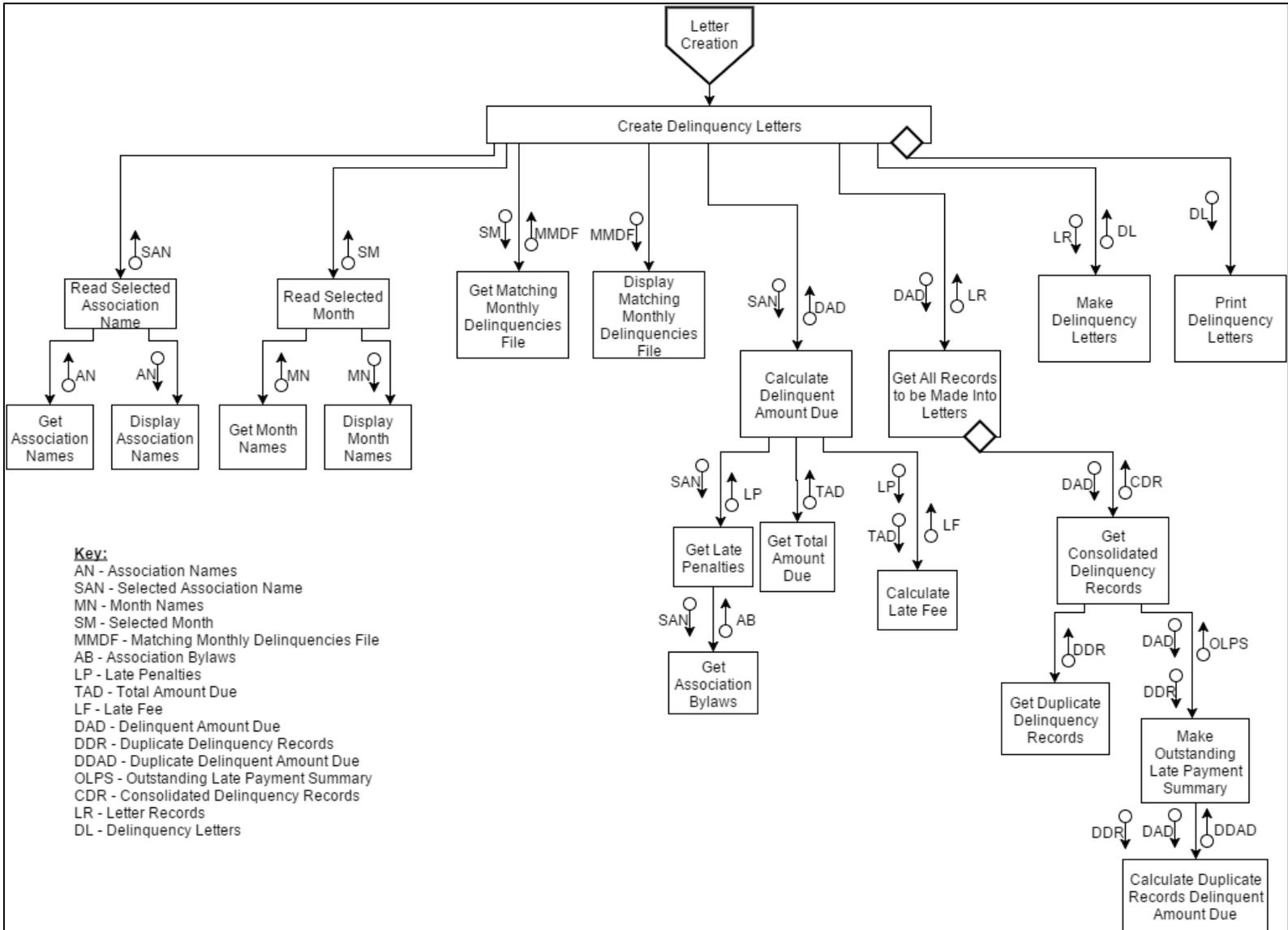


Figure 40: HOA Structure Chart Page 6 – Letter Creation Off-Page Modules

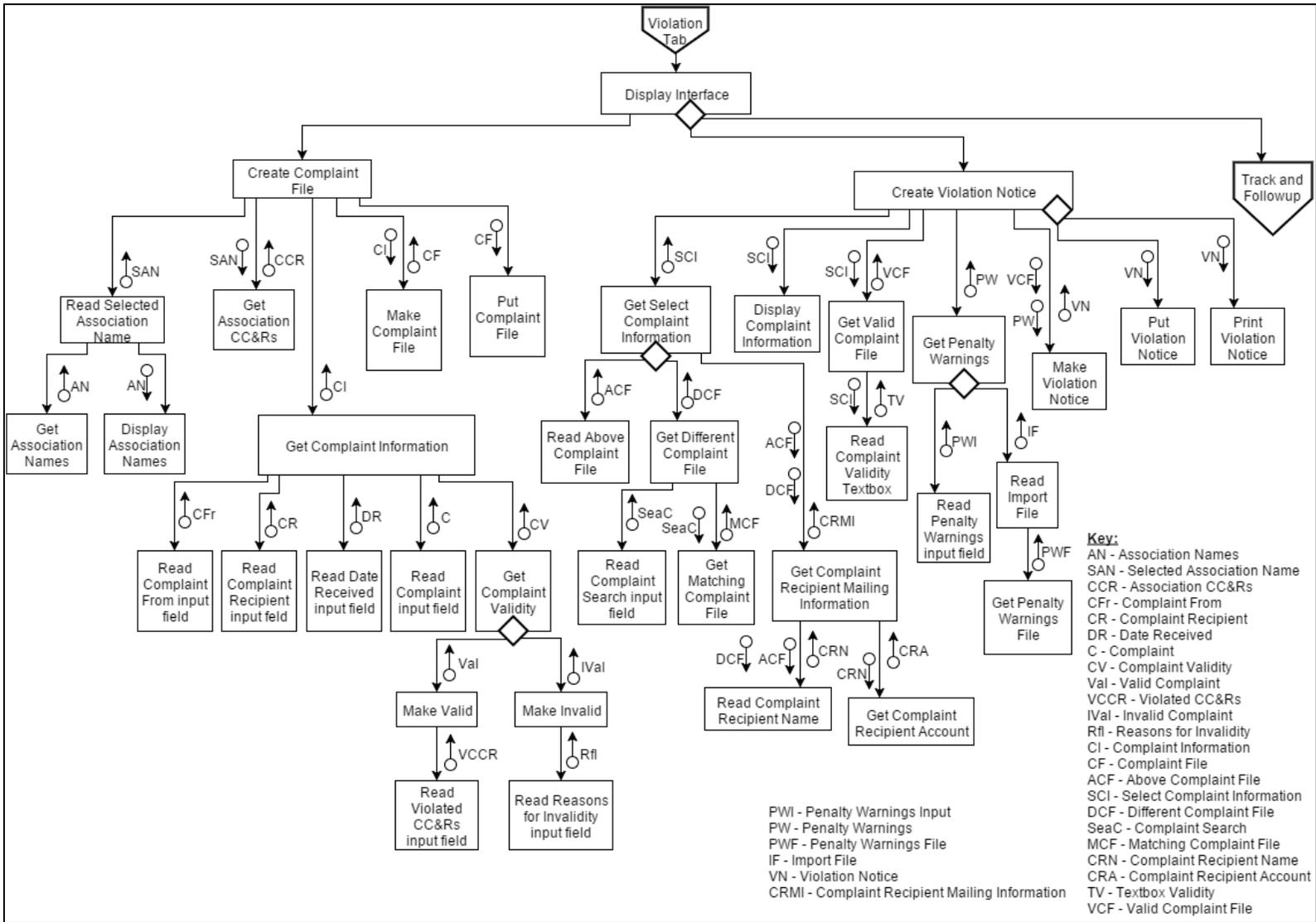


Figure 41: HOA Structure Chart Page 7 – Violation Tab Modules

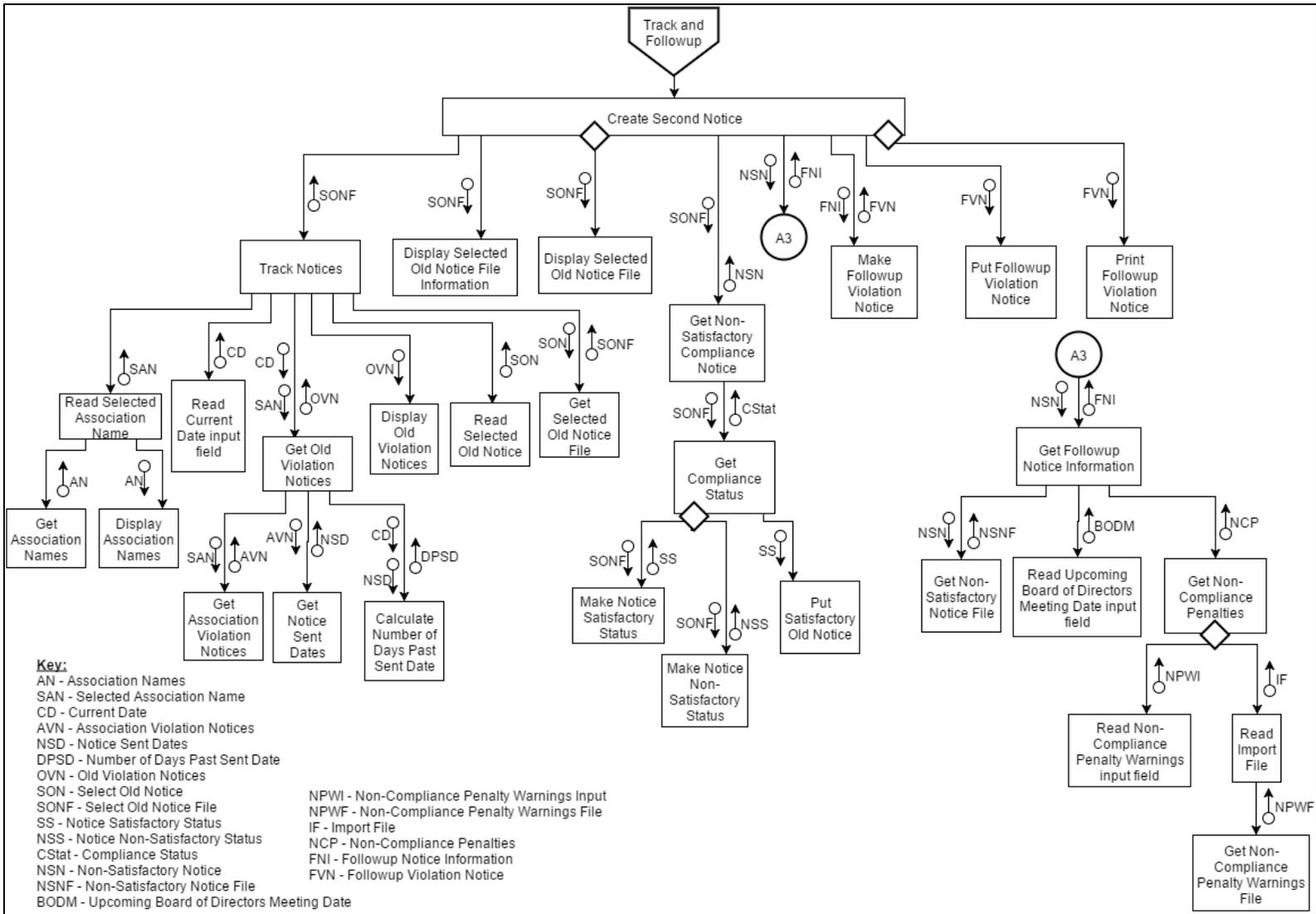


Figure 42: HOA Structure Chart Page 8 - Track and Followup Off-Page Connector Modules



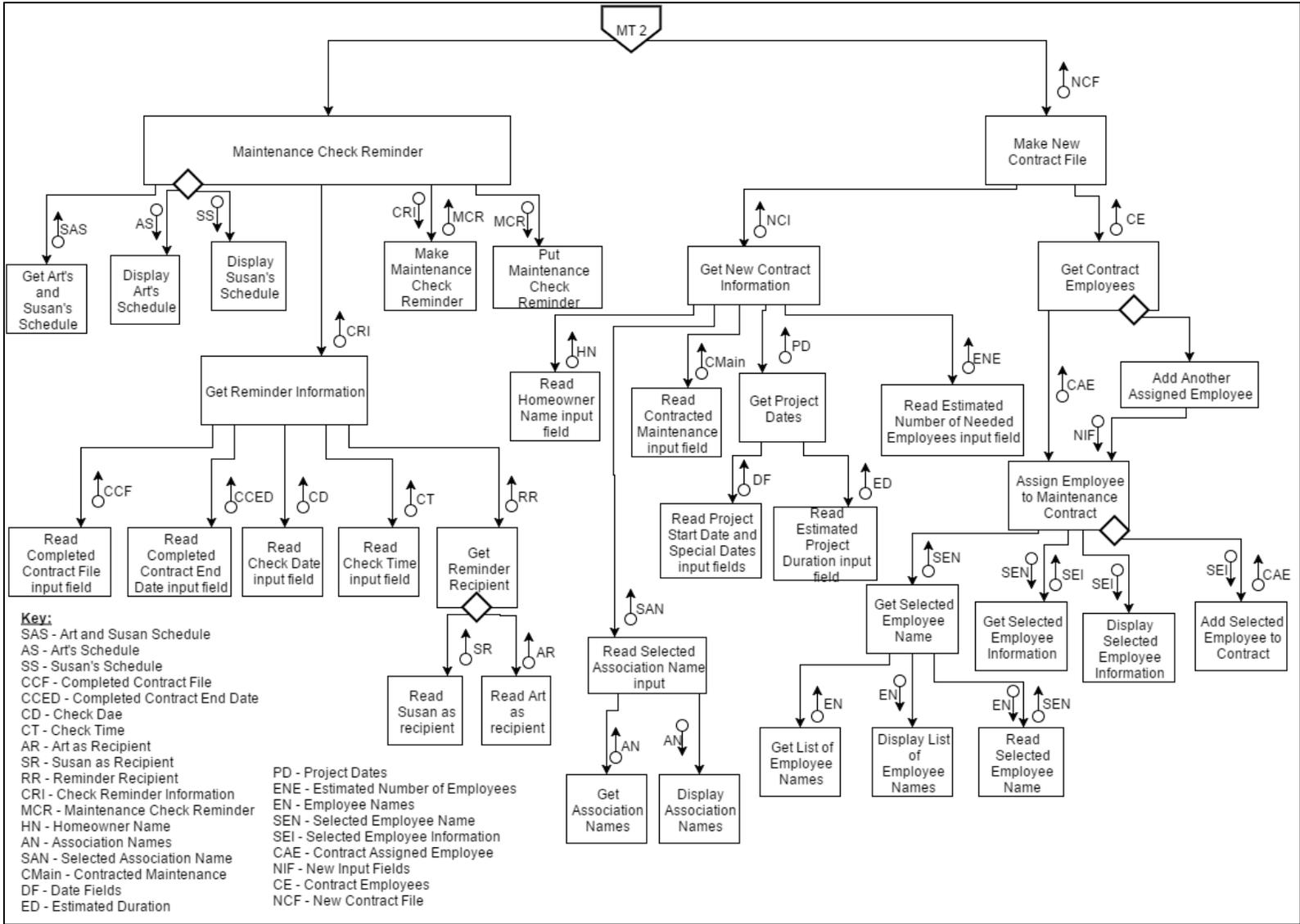


Figure 44: HOA Structure Chart Page 10 - MT2 Off-Page Connector Modules

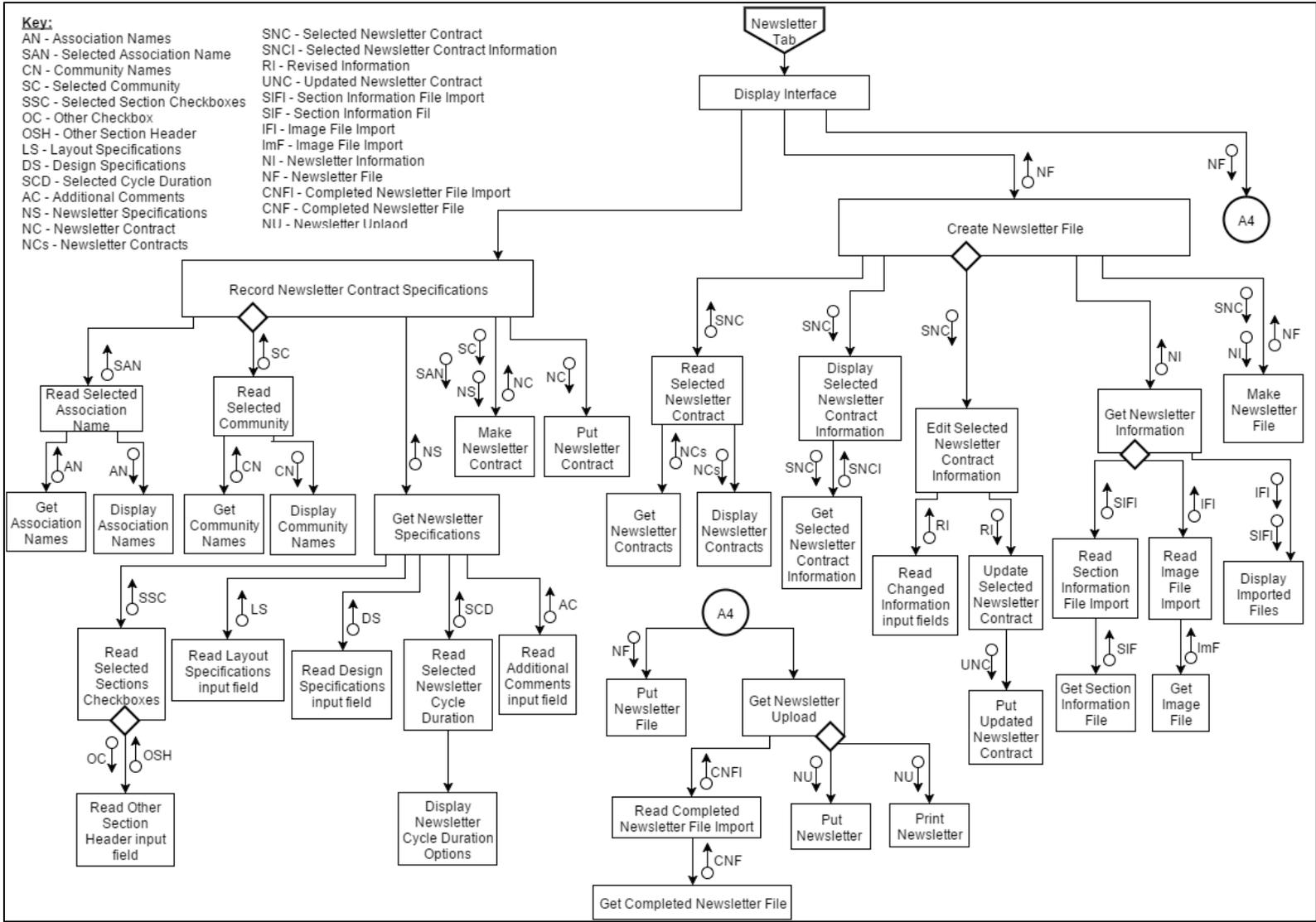


Figure 45: HOA Structure Chart Page 11 - Newsletter Tab Modules

## **Modules Specifications – Pseudocode**

### *HOA Structure Chart Page 1 Module Pseudocode*

#### **Get Ordered Association List**

INPUTS: N/A

OUTPUTS: Ordered Association List

PSEUDOCODE:

RETRIEVE Ordered Association List

#### **Display Ordered Association List**

INPUTS: Ordered Association List

OUTPUTS: Selected Association Name

PSEUDOCODE:

RETRIEVE Ordered Association List

RETRIEVE Selected Association Name FROM CALL Read Selected Association Name()

SEND Ordered Association List to Print Ordered Association List

#### **Read Selected Association Name**

INPUTS: N/A

OUTPUTS: Selected Association Name

PSEUDOCODE:

RETRIEVE Selected Association Name

#### **Print Ordered Association list**

INPUTS: Ordered Association List

OUTPUTS: N/A

PSEUDOCODE:

PRINT Ordered Association List

**Get Selected Association File**

INPUTS: Selected Association Name

OUTPUT: Selected Association File

PSEUDOCODE:

RETRIEVE Selected Association File

**Display Selected Association File**

INPUTS: Selected Association File

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Selected Association File

SEND Selected Association File to Edit Selected Association file and Maintain  
Association Committee List

**Edits Selected Association File**

INPUTS: Selected Association File, Committee Information

OUTPUTS: Committee Information

PSEUDOCODE:

RETRIEVE Selected Association File

RETRIEVE Committee Information FROM CALL ReadChangedInputFields()

SEND Committee Information to Update Association File

**Read Changed Information Input Fields**

INPUTS: N/A

OUTPUTS: Committee Information

PSEUDOCODE:

RETRIEVE Committee Information

**Update Association File**

INPUTS: Committee Information

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Committee Information

CREATE Updated Association File

Send Updated Association File to Put Updated Association File

**Put Updated Association File**

INPUTS: Updated Association File

OUTPUTS:N/A

PSEUDOCODE:

RETRIEVE Updated Association File

STORE Updated Association File

**Maintain Association Committee List**

INPUTS: Selected Association File

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Committee Name FROM CALL GetCommitteeNames()

SEND Committee Name to Display Committee Name

RETRIEVE Selected Committee Name and Updated Committee List FROM CALL

DisplayCommitteNames()

SEND Selected Committee Name to Get Selected Committee Information

RETRIEVE Selected Committee Information FROM CALL

GetSelectedCommitteeInformation()

SEND Selected Committee Information to Display Selected Committee Information

RETRIEVE Updated Member List FROM Call DisplaySelectedCommitteeInformation()

SEND Updated Committee List and Updated Member List to Make Committee List

### **Get Committee Names**

INPUTS: N/A

OUTPUTS: Committee Name

PSEUDOCODE:

RETRIEVE Committee Name

### **Display Committee Names**

INPUTS: Committee Name

OUTPUTS: Selected Committee Name, Updated Committee List

PSEUDOCODE:

RETRIEVE Committee Name

RETRIEVE Selected Committee Name FROM CALL ReadSelectedCommitteeName()

RETRIEVE Updated Committee List FROM CALL AddNewCommittee()

RETRIEVE Updated Committee List FROM CALL RemoveCommittee(Selected  
Committee Name)

### **Read Selected Committee Name**

INPUTS: N/A

OUTPUTS: Selected Committee Name

PSEUDOCODE:

RETRIEVE Selected Committee Name

**Add New Committee**

INPUTS: N/A

OUTPUTS: Updated Committee List

PSEUDOCODE:

RETRIEVE New Committee Name FROM CALL

ReadNewCommitteeNameInputField()

SEND New Committee Name to Update Committee List

**Remove Committee**

INPUTS: Selected Committee Name

OUTPUTS: Updated Committee List

PSEUDOCODE:

RETRIEVE Selected Committee Name

RETRIEVE Updated Committee List FROM CALL UpdateCommitteeList()

SEND Selected Committee Name to Delete Selected Committee Name

**Read New Committee Input Field**

INPUTS: N/A

OUTPUTS: New Committee Name

PSEUDOCODE:

RETRIEVE New Committee Name

**Update Committee List**

INPUTS: New Committee Name

OUTPUTS: Updated Committee List

PSEUDOCODE:

RETRIEVE New Committee Name

STORE New Committee Name

RETURN Updated Committee List

**Delete Selected Committee Name**

INPUTS: Selected Committee Name

OUTPUTS: N/A

PSEUDOCODE:

REMOVE Selected Committee Name

**Get Selected Committee Information**

INPUTS: Selected Committee Name

OUTPUTS: Selected Committee Information

PSEUDOCODE:

RETRIEVE Selected Committee Information

**Display Selected Committee Information**

INPUTS: Selected Committee Information

OUTPUTS: Updated Member List

PSEUDOCODE:

RETRIEVE Selected Committee Information

RETRIEVE Updated Member List FROM CALL AddNewMember()

RETRIEVE Updated Member List FROM CALL RemoveMember()

RETURN Updated Member List

**Add New Member**

INPUTS: N/A

OUTPUTS: Updated Member List

PSEUDOCODE:

RETRIEVE New Member Name FROM CALL ReadNewMemberNameInputField()  
SEND New Member Name to Update Member List

**Remove Member**

INPUTS: N/A

OUTPUTS: Updated Member List

PSEUDOCODE:

RETRIEVE Updated Member List FROM CALL UpdateMemberList()  
SEND Selected Member name to Delete Selected Member Name

**Read New Member Name Input Field**

INPUTS: N/A

OUTPUTS: New Member Name

PSEUDOCODE:

RETRIEVE New Member Name  
Return New Member Name

**Update Member List**

INPUTS: New Member Name

OUTPUTS: Updated Member list

PSEUDOCODE:

RETRIEVE New Member Name  
Return Updated Member List  
STORE Update Member List

**Deleted Selected Member Name**

INPUTS: Selected Member Name

OUTPUTS: N/A

PSEUDOCODE:

DELETE Selected Member Name

**Make Committee List**

INPUTS: Updated Committee Lists, Updated Member List

OUTPUTS: N/A

PSEUDOCODE:

CREATE Committee List

SEND Committee List to Print Committee List

**Print Committee List**

INPUTS: Committee List

OUTPUTS: N/A

PSEUDOCODE:

PRINT Committee List

**Add New Association**

INPUTS: Association File Information, New Association File

OUTPUTS: Association File Information, New Association File

PSEUDOCODE:

**Get New Association File Information**

INPUTS: N/A

OUTPUTS: Association File Information

PSEUDOCODE:

RETRIEVE Association Information FROM CALL

ReadAssociationInformationInputFields()

RETRIEVE Required Services FROM CALL  
ReadSelectedRequiredServicesRadioButtons()  
RETURN Association File Information

**Read Association Information Input Fields**

INPUTS:N/A

OUTPUTS: Association Information

PSEUDOCODE:

RETRIEVE Association Information  
RETURN Association Information

**Read Selected Required Services Radio Buttons**

INPUTS:N/A

Outputs: Required Services

PSEUDOCODE:

RETRIEVE Required Services  
RETURN Required Services

**Get Association Bylaws**

INPUTS: Required Services

OUTPUTS: Association Bylaws

PSEUDOCODE:

RETRIEVE Required Services  
RETRIEVE Monthly Fee FROM CALL ReadMonthlyFee(Required Services)  
RETRIEVE Payment Period FROM CALL GetAssociationPaymentPeriod()  
RETRIEVE Bylaw Late Penalties FROM CALL ReadAssociationLatePenalties()  
RETRIEVE Bylaws Allowed Late Period FROM CALL  
ReadAssociationAllowedLatePeriod()

RETURN Association Bylaws

**Read Monthly Fee**

INPUTS: Required Services

OUTPUTS: Monthly Fee

PSEUDOCODE:

RETRIEVE Monthly Fee FROM CALL CalculateMonthlyFee()

RETURN Monthly Fee

**Calculate Monthly Fee**

INPUTS: Required Services

OUTPUTS: Monthly Fee

PSEUDOCODE:

CALCULATE Monthly Fee = Required Services/12

RETURN Monthly Fee

**Get Association Payment Period**

INPUTS: N/A

OUTPUTS: Payment Period

PSEUDOCODE:

RETRIEVE Association Creation Date FROM CALL ReadAssociationCreationDate()

CREATE Payment Period

RETURN Payment Period

**Read Association Creation Date**

INPUTS: N/A

OUTPUTS: Association Creation Date

PSEUDOCODE:

RETRIEVE Association Creation Date

RETURN Association Creation Date

**Read Association Allowed Late Period**

INPUTS: N/A

OUTPUTS: Bylaws allowed late period

PSEUDOCODE:

RETRIEVE Bylaws allowed late period

RETURN Bylaws allowed late period

**Get Homeowners Accounts**

INPUTS: N/A

OUTPUTS: Homeowners Account

PSEUDOCODE:

RETRIEVE Homeowners Accounts FROM CALL ReadHomeownerAccountFileImport()

RETURN Homeowner's Accounts

**Read Homeowner Accounts File Import**

INPUTS: N/A

OUTPUTS: Homeowner Accounts

PSEUDOCODE:

RETRIEVE Homeowner Accounts

RETURN Homeowner Accounts

**Make News Association File**

INPUTS: Association File Information

OUTPUTS: New Association File

PSEUDOCODE:

RETRIEVE Association File Information

CREATE New Association File

RETURN New Association File

**Put New Association File**

INPUTS: New Association File

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE New Association File

STORE New Association File

*HOA Structure Chart Page 2 Module Pseudocode*

**Display Interface**

INPUTS: N/A

OUTPUTS: N/A

PSEUDOCODE:

PRINT Interface

**Manage Monthly Bill**

INPUTS: Selected Association Name, Selected Month

OUTPUTS: Selected Association Name, Selected Month

PSEUDOCODE:

RETRIEVE Selected Association Name FROM CALL ReadSelectedAssociationName

RETRIEVE Selected Month FROM CALL Read Selected Month

SEND Selected Association Name to A

**Read Selected Association Name**

INPUTS: Association Names

OUTPUTS: Selected Association Name, Association Names

PSEUDOCODE:

RETRIEVE Selected Association Name

RETURN Selected Association Name

**Get Association Names**

INPUTS: N/A

OUTPUTS: Association Names

PSEUDOCODE:

RETRIEVE Association Names FROM INPUT FORM

RETURN Association Names

**Display Association Names**

INPUTS: Association Names

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Association Names

PRINT Association Names

**Read Selected Month**

INPUTS: Month Names

OUTPUTS: Selected Month, Month Names

PSEUDOCODE:

RETRIEVE Month Names FROM Input

RETURN Selected Month, Month Names

**Get Month Names**

INPUTS: N/A

OUTPUTS: Month Names

PSEUDOCODE:

RETURN Month Names

**Display Month Names**

INPUTS: Month Names

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Month Names FROM CALL Get Month Names

PRINT Month Names

**Create and Print Bill**

INPUTS: Selected Association Name, Selected Month, Monthly Bill Details,  
Homeowner Accounts, Monthly Bill

OUTPUTS: Selected Association Name, Selected Month, Monthly Bill Details,  
Homeowner Accounts, Monthly Bill

PSEUDOCODE:

RETRIEVE Selected Association Name

RETRIEVE Selected Month

CALL Get Monthly Bill Details

CALL Get Homeowner Account

CALL Make Monthly Bill

CALL Print Monthly Bill

**Get Monthly Bill Details**

INPUTS: Selected Association Name, Selected Month, Association Bylaws, Late Penalties, Due Date, Total Amount Due

OUTPUTS: Selected Association Name, Selected Month, Association Bylaws

PSEUDOCODE:

**Get Association Bylaws**

INPUTS: Selected Association Name

OUTPUTS: Association Bylaws

PSEUDOCODE:

RETRIEVE Selected Association Name

COMPARE Selected Association Name FROM DATASTORE

RETRIEVE Association ByLaws

RETURN Association ByLaws

**Get Late Penalties**

INPUTS: Association Bylaws

OUTPUTS: Late Penalties

PSEUDOCODE:

RETRIEVE Association ByLaws

RETURN Late Penalties

**Get Due Date**

INPUTS: Association Bylaws, Selected Month, Payment Period

OUTPUTS: Due Date

PSEUDOCODE:

RETRIEVE Association ByLaws

CALL Get Payment Period

CALL Put Due Date

RETURN Due Date

**Get Payment Period**

INPUTS: Association Bylaws

OUTPUTS: Payment Period

PSEUDOCODE:

RETRIEVE Association ByLaws

RETURN Payment Period

**Put Due Date**

INPUTS: Due Date

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Due Date

SAVE Due Date

**Get Total Amount Due**

INPUTS: Association Bylaws, Monthly Fee, Association Credit, Total Amount Due

OUTPUTS: Total Amount Due

PSEUDOCODE:

RETRIEVE Association ByLaws

RETRIEVE Monthly Fee

RETRIEVE Association Credit

RETRIEVE Total Amount Due

CALL Get Association Credit

CALL Calculate Total Amount Due

CALL Put Total Amount Due

RETURN Total Amount Due

**Get Monthly Fee**

INPUTS: Association Bylaws

OUTPUTS: Monthly Fee

PSEUDOCODE:

RETRIEVE Association ByLaws

RETURN Monthly Fee

**Get Association Credit**

INPUTS: N/A

OUTPUTS: Association Credit

PSEUDOCODE:

RETURN Association Credit

**Calculate Total Amount Due**

INPUTS: Association Credit, Monthly Fee

OUTPUTS: Total Amount Due

PSEUDOCODE:

RETRIEVE Association Credit

RETRIEVE Monthly Fee

CALCULATE Total Amount Due = Association Credit - Monthly Fee

RETURN Total Amount Due

**Put Total Amount Due**

INPUTS: Total Amount Due

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Total Amount Due

SAVE Total Amount Due

**Get Homeowner Account**

INPUTS: N/A

OUTPUTS: Homeowner Accounts

PSEUDOCODE:

RETURN Homeowner Accounts

**Make Monthly Bill**

INPUTS: Monthly Bill Details, Homeowner Accounts

OUTPUTS: Monthly Bill

PSEUDOCODE:

RETRIEVE Monthly Bill Details

RETRIEVE Homeowner Account

CREATE Monthly Bill = Monthly Bill Details + Homeowner Accounts

RETURN Monthly Bill

**Print Monthly Bill**

INPUTS: Monthly Bill

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Monthly Bill

PRINT Monthly Bill

**Manage Incurred Bill**

INPUTS: Selected Association Name, Association Fees, Calculated Incurred Fee,  
Incurred Bill

OUTPUTS: Selected Association Name, Association Fees, Calculated Incurred Fee,  
Incurred Bill

PSEUDOCODE:

CALL Read Selected Association Name

CALL Get Association Fees

CALL Calculate Incurred Fee

CALL Make Incurred Bill

CALL Put Incurred Bill

CALL Send Incurred Bill

**Read Selected Association Name**

INPUTS: Association Names

OUTPUTS: Selected Association Name, Association Names

PSEUDOCODE:

RETRIEVE Selected Association Name

RETURN Selected Association Name

**Get Association Names**

INPUTS: N/A

OUTPUTS: Association Names

PSEUDOCODE:

RETRIEVE Association Names FROM INPUT FORM

RETURN Association Names

**Display Association Names**

INPUTS: Association Names

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Association Names

PRINT Association Names

**Get Association Fees**

INPUTS: Water Fee Amount, Gas Fee Amount, Electricity Fee Amount, Trash Pickup Fee Amount, Maintenance Fee Amount, HOA Fee Amount

OUTPUTS: Association Fees

PSEUDOCODE:

RETRIEVE Water Fee Amount

RETRIEVE Gas Fee Amount

RETRIEVE Electricity Fee Amount

RETRIEVE Trash Pickup Fee Amount

RETRIEVE Maintenance Fee Amount

RETRIEVE HOA Fee Amount

CREATE Association Fees = Water Fee Amount + Gas Fee Amount + Electricity Fee Amount + Trash Pickup Amount + Maintenance Fee Amount + HOA Fee Amount

RETURN Association Fees

**Read Water Fee Amount input field**

INPUTS: N/A

OUTPUTS: Water Fee Amount

PSEUDOCODE:

RETURN Water Fee Amount

**Read Gas Fee Amount input field**

INPUTS: N/A

OUTPUTS: Gas Fee Amount

PSEUDOCODE:

RETURN Gas Fee Amount

**Read Electricity Fee Amount input field**

INPUTS: N/A

OUTPUTS: Electricity Fee Amount

PSEUDOCODE:

RETURN Electricity Fee Amount

**Read Trash Pickup Fee Amount input field**

INPUTS: N/A

OUTPUTS: Trash Pickup Fee Amount

PSEUDOCODE:

RETURN Trash Pickup Fee Amount

**Read Maintenance Fee Amount input field**

INPUTS: N/A

OUTPUTS: Maintenance Fee Amount

PSEUDOCODE:

RETURN Maintenance Fee Amount

**Read HOA Fee Amount input field**

INPUTS: N/A

OUTPUTS: HOA Fee Amount

PSEUDOCODE:

RETURN HOA Fee Amount

**Calculate Incurred Fee**

INPUTS: Association Fees

OUTPUTS: Calculated Incurred Fee

PSEUDOCODE:

RETRIEVE Association Fees

CREATE Calculated Incurred Fee

Calculated Incurred Fee = Association Fees1 + Association Fees2 + ... + Association FeesN

RETURN Calculated Incurred Fee

**Make Incurred Bill**

INPUTS: Calculated Incurred Fee, Selected Association Name

OUTPUTS: Incurred Bill

PSEUDOCODE:

RETRIEVE Calculated Incurred Fee

RETRIEVE Selected Association Name

CREATE Incurred Bill

RETURN Incurred Bill

**Put Incurred Bill**

INPUTS: Incurred Bill

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Incurred Bill

SAVE Incurred Bill

**Send Incurred Bill**

INPUTS: Incurred Bill

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Incurred Bill

PRINT Incurred Bill

*HOA Structure Chart Page 3 Module Pseudocode*

**Manage Special Assessment Bill**

INPUTS: Selected Association Name, Special Assessment Information, Special Assessment Bill

OUTPUTS: Selected Association Name, Special Assessment Information, Special Assessment Bill

PSEUDOCODE:

RETRIEVE Selected Association Name, Special Assessment Information, Special Assessment Bill

RETURN Selected Association Name, Special Assessment Information, Special Assessment Bill

**Read Selected Association Name**

INPUTS: Association Names

OUTPUTS: Selected Association Name, Association Names

PSEUDOCODE:

RETRIEVE Selected Association Name

RETURN Selected Association Name

**Get Association Names**

INPUTS: N/A

OUTPUTS: Association Names

PSEUDOCODE:

RETRIEVE Association Names FROM INPUT FORM

RETURN Association Names

**Display Association Names**

INPUTS: Association Names

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Association Names

PRINT Association Names

**Get Special Assessment Information**

INPUTS: Special Assessment Amount, Special Assessment Reason, Date Received,  
Chosen Due Date, Special Charge

OUTPUTS: Special Assessment Information

PSEUDOCODE:

RETRIEVE Special Assessment Amount, Special Assessment Reason, Date Received,  
Chosen Due Date, Special Charge

CREATE Special Assessment Information

RETURN Special Assessment Information

**Read Special Assessment Amount input field**

INPUTS: N/A

OUTPUTS: Special Assessment Amount

PSEUDOCODE:

RETURN Special Assessment Amount FROM INPUT

**Read Special Assessment Reason input field**

INPUTS: N/A

OUTPUTS: Special Assessment Reason

PSEUDOCODE:

RETURN Special Assessment Reason FROM INPUT

**Read Date Received input field**

INPUTS: N/A

OUTPUTS: Date Received

PSEUDOCODE:

RETURN Date Received FROM INPUT

**Read Chosen Due Date input field**

INPUTS: N/A

OUTPUTS: Chosen Due Date

PSEUDOCODE:

RETURN Chosen Due Date FROM INPUT

**Make Special Assessment Bill**

INPUTS: Selected Association Name, Special Assessment Information

OUTPUTS: Special Assessment Bill

PSEUDOCODE:

RETRIEVE Selected Association Name, Special Assessment Information

CREATE Special Assessment Bill

RETURN Special Assessment Bill

**Print Special Assessment Bill**

INPUTS: Special Assessment Bill

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Special Assessment Bill

PRINT Special Assessment Bill

***HOA Structure Chart Page 4 Module Pseudocode***

**Process Payment**

INPUTS: Date Received, Payment Coupon Information, Member Association, Payment Record, Association Separate Worksheet

OUTPUTS: Payment Coupon Information, Member Association, Payment Record

PSEUDOCODE:

RETRIEVE Date Recieved FROM CALL ReadDateRecieved()

RETRIEVE Payment Coupon Information, Member Association FROM CALL  
GetPaymentCouponInformation()

RETRIEVE Payment Record FROM CALL MakePaymentRecord()

RETRIEVE Association Separate Worksheet FROM CALL GetSeperateWorksheet()

RETURN Payment Record

**Read Date Received Input Field**

INPUTS: N/A

OUTPUTS: Date Received

PSEUDOCODE:

RETRIEVE Date Received

RETURN Date Received

### **Get Payment Coupon Information**

INPUTS: Payment Amount, Member Name, Member Association, Account Number,  
Check Number

OUTPUTS: Payment Coupon Information, Member Association

#### PSEUDOCODE:

RETRIEVE Payment Amount FROM CALL ReadPaymentAmountInputField()

RETRIEVE Member Name FROM CALL ReadMemberNameInputField()

RETRIEVE Member Association FROM CALL ReadMemberAssociationInputField()

RETRIEVE Account Number FROM CALL ReadAccountNumberInputField()

RETRIEVE Check Number FROM CALL ReadCheckNumberInputField()

### **Read Payment Amount Input Field**

INPUTS: N/A

OUTPUTS: Payment Amount

#### PSEUDOCODE:

RETRIEVE Payment Amount FROM INPUT

### **Read Member Name Input Field**

INPUTS: N/A

OUTPUTS: Member Name

#### PSEUDOCODE:

RETRIEVE Member Name FROM INPUT

### **Read Member Association Input Field**

INPUTS: N/A

OUTPUTS: Member Association

PSEUDOCODE:

RETRIEVE Member Association FROM INPUT

**Read Account Number Input Field**

INPUTS: N/A

OUTPUTS: Account Number

PSEUDOCODE:

RETRIEVE Account Numbers FROM INPUT

**Read Check Number Input Field**

INPUTS: N/A

OUTPUTS: Check Number

PSEUDOCODE:

RETRIEVE Check Numbers FROM INPUT

**Search for Member Information**

INPUTS: No Payment Coupon Checkbox, Member Name Search, Member Account  
Summary

OUTPUTS: Member Account Summary

PSEUDOCODE:

RETRIEVE No Payment Coupon Checkbox FROM CALL

ReadNoPaymentCouponCheckbox()

RETRIEVE Member Name Search FROM CALL ReadMemberNameSearchCheckbox()

RETRIEVE Member Account Summary FROM CALL GetMemberAccountSummary()

RETURN Member Account Summary

**Read Payment Coupon Checkbox**

INPUTS: N/A

OUTPUTS: No Payment Coupon Checkbox

PSEUDOCODE:

RETRIEVE No Payment Coupon Checkbox FROM INPUT

**Read Member Name Search Input Field**

INPUTS: N/A

OUTPUTS: Member Name Search

PSEUDOCODE:

RETRIEVE Check Numbers FROM INPUT

**Get Member Account Summary**

INPUTS: Member Account, Member Name, Member Association, Member Account  
Summary

OUTPUTS: Member Account, Member Name, Member Association, Member Account  
Summary

PSEUDOCODE:

RETRIEVE Member Account FROM CALL GetMemberAccount()

RETRIEVE MemberName FROM CALL GetMemberName()

RETRIEVE Member Association FROM CALL GetMemberAssociation

RETRIEVE Member Account Summary FROM CALL

MakeMemberAccountSummary()

RETURN Member Account Summary

**Search Separate Worksheets**

INPUTS: N/A

OUTPUTS: N/A

PSEUDOCODE:

RETURN Data

**Get Member Account**

INPUTS: N/A

OUTPUTS: Member Account

PSEUDOCODE:

Retrieve Member Account FROM INPUT

**Get Member Name**

INPUTS: Member Account

OUTPUTS: Member Name

PSEUDOCODE:

RETRIEVE Member Name FROM INPUT

RETURN Member Account

**Get Member Association**

INPUTS: Member Account

OUTPUTS: Member Association

PSEUDOCODE:

RETRIEVE Member Account

RETURN Member Association

**Make Member Account Summary**

INPUTS: Member Account, Member Name

OUTPUTS: Member Account Summary

PSEUDOCODE:

RETRIEVE Member Account, Member Name

CREATE Member Account Summary

RETURN Member Account Summary

**Display Member Account Summary**

INPUTS: Member Account Summary

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Member Account Summary

PRINT Member Account Summary

**Make Payment Record**

INPUTS: Payment Coupon Information

OUTPUTS: Payment Record

PSEUDOCODE:

RETRIEVE Payment Coupon Information

CREATE Payment Record

RETURN Payment Record

**Get Separate Worksheet**

INPUTS: Member Association

OUTPUTS: Association Separate Worksheet

PSEUDOCODE:

RETRIEVE Member Association

CREATE Association Separate Worksheet

RETURN Association Separate Worksheet

### **Put Payment Record**

INPUTS: Payment Record

OUTPUTS: Updates Fiscal Worksheet

PSEUDOCODE:

RETRIEVE Payment Reocrd

CREATE Updates Fiscal Worksheet

STORE Updates Fiscal Worksheet

### **Update Fiscal Year Association Worksheet**

INPUTS: Updated Fiscal Worksheet

OUTPUTS: Fiscal Year Workshet

PSEUDOCODE:

RETRIEVE Updated Fiscal Worksheet

CREATE Fiscal Year Worksheet

RETURN Fiscal Year Worksheet

### **Put Fiscal Year Association Worksheet**

INPUTS: Fiscal Year Worksheet

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Fiscal Year Worksheet

STORE Fiscal Year Worksheet

*HOA Structure Chart Page 5 Module Pseudocode*

**Create Delinquency Records**

INPUTS: Selected Association Name, Current Date, Delinquency Records

OUTPUTS: Selected Association Name, Current Date, Delinquency Records

PSEUDOCODE:

CALL Read Selected Association Name

CALL Read Current Date Input Field

CALL Make Delinquency Records

CALL Display Delinquency Records

**Read Selected Association Name**

INPUTS: Association Names

OUTPUTS: Selected Association Name, Association Names

PSEUDOCODE:

RETRIEVE Selected Association Name

RETURN Selected Association Name

**Get Association Names**

INPUTS: N/A

OUTPUTS: Association Names

PSEUDOCODE:

RETRIEVE Association Names FROM INPUT FORM

RETURN Association Names

**Display Association Names**

INPUTS: Association Names

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Association Names

PRINT Association Names

**Read Current Date Input field**

INPUTS: N/A

OUTPUTS: Current Date

PSEUDOCODE:

RETURN Current Date FROM CURRENT DATE INPUT FIELD

**Make Delinquency Records**

INPUTS: Selected Association Name, Current Date, Association Separate Worksheet, Blank "Date Received" Accounts, Delinquent Status Accounts

OUTPUTS: Delinquency Records, Selected Association Name, Association Separate Worksheet, Current Date

PSEUDOCODE:

RETRIEVE Selected Association Name

RETRIEVE Current Date

CALL Get Association Separate Worksheet

CALL Get Blank Date Received Accounts

CALL Get Delinquent Status Accounts

CREATE Delinquency Records

RETURN Delinquency Records

**Get Association Separate Worksheet**

INPUTS: Selected Association Name

OUTPUTS: Association Separate Worksheet

PSEUDOCODE:

RETRIEVE Selected Association Name

RETRIEVE Association Separate Worksheet FROM DATASTORE

RETURN Association Separate Worksheet

**Get Blank “Date Received” Accounts**

INPUTS: Association Separate Worksheet, Blank “Date Received”s, Blank Field Month,  
Blank Field Account Name

OUTPUTS: Blank “Date Received”s, Blank “Date Received” Accounts

PSEUDOCODE:

**Get Blank “Date Received” Fields**

INPUTS: N/A

OUTPUTS: Blank “Date Received”s

PSEUDOCODE:

RETRIEVE Blank “Date Received”s FROM input form

RETURN Blank “Date Received”s

**Get Blank Field Month**

INPUTS: Blank “Date Received”s

OUTPUTS: Blank Field Month

PSEUDOCODE:

RETRIEVE Blank “Date Received”s

RETRIEVE Blank Field Month FROM input form

RETURN Blank Field Month

**Get Account Name**

INPUTS: N/A

OUTPUTS: Blank Field Account Name

PSEUDOCODE:

RETRIEVE Blank Field Account Name FROM input form

RETURN Blank Field Account Name

**Get Delinquent Status Accounts**

INPUTS: Current Date, Selected Association Name, Association Bylaws, Allowed Late Period, Bill Due Date, Days Past Due Date

OUTPUTS: Association Bylaws, Current Date, Bill Due Date, Delinquent Status Accounts

PSEUDOCODE:

RETRIEVE Current Date

RETRIEVE Selected Association Name

CALL Calculate Days Past Due Date

RETURN Delinquent Status Accounts

**Get Association ByLaws**

INPUTS: Selected Association Name

OUTPUTS: Association ByLaws

PSEUDOCODE:

RETRIEVE Selected Association Name

COMPARE Selected Association Name

RETRIEVE Association ByLaws

RETURN Association ByLaws

**Get Allowed Late Period**

INPUTS: Association ByLaws

OUTPUTS: Allowed Late Period

PSEUDOCODE:

RETRIEVE Association ByLaws

RETURN Allowed Late Period

**Get Bill Due Date**

INPUTS: Association ByLaws

OUTPUTS: Bill Due Date

PSEUDOCODE:

RETRIEVE Association ByLaws

RETURN Allowed Late Period

**Calculate Days Past Due Date**

INPUTS: Current Date, Bill Due Date

OUTPUTS: Days Past Due Date

PSEUDOCODE:

RETRIEVE Current Date

RETRIEVE Bill Due Date

IF Current Date > Bill Due Date

RETURN Days Past Due Date

**Display Delinquency Records**

INPUTS: Delinquency Records

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Delinquency Records

PRINT Delinquency Records

**Copy Monthly Records**

INPUTS: Selected Association Name, Selected Delinquent Month, Monthly Delinquencies File

OUTPUTS: Selected Association Name, Selected Delinquent Month, Monthly Delinquencies File

PSEUDOCODE:

RETRIEVE Selected Association Name

CALL Read Selected Delinquent Month

CALL Create Monthly Delinquencies File

CALL Put Monthly Delinquencies File

**Read Selected Delinquent Month**

INPUTS: Months with Delinquent Records

OUTPUTS: Months with Delinquent Records, Selected Delinquent Month

PSEUDOCODE:

CALL Get Months with Delinquent Records

CALL Display Months with Delinquent Records

**Get Months with Delinquent Records**

INPUTS: N/A

OUTPUTS: Month Delinquent Records

PSEUDOCODE:

RETRIEVE Month Delinquent Records FROM DATASTORE

RETURN Month Delinquent Records

**Display Display Months with Delinquent Records**

INPUTS: Month Delinquent Records

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Month Delinquent Records

PRINT Month Delinquent Records

**Create Monthly Delinquencies File**

INPUTS: Selected Association Name, Selected Delinquent Month, Association Initials,  
Selected Month Abbreviation, Selected Month Delinquent Records

OUTPUTS: Selected Association Name, Selected Delinquent Month, Monthly  
Delinquencies File

PSEUDOCODE:

RETRIEVE Selected Association Name

RETRIEVE Selected Delinquent Month

CALL Get Association Initials

CALL Get Selected Month Abbreviation

CALL Get Selected Month Delinquent Records

RETURN Monthly Delinquencies File

**Get Association Initials**

INPUTS: Selected Association Name

OUTPUTS: Associations Initials

PSEUDOCODE:

RETRIEVE Selected Association Name

RETURN Association Initials

**Get Selected Month Abbreviation**

INPUTS: Selected Delinquencies Month

OUTPUTS: Selected Month Abbreviation

PSEUDOCODE:

RETRIEVE Selected Delinquencies Month

RETURN Selected Month Abbreviation

**Get Selected Month Delinquent Records**

INPUTS: Selected Delinquencies Month

OUTPUTS: Selected Month Delinquent Records

PSEUDOCODE:

RETRIEVE Selected Delinquencies Month

LOOKUP Selected Month Delinquent Records FROM DATASTORE USING Selected Delinquencies Month

RETURN Selected Month Delinquent Records

**Put Monthly Delinquencies File**

INPUTS: Monthly Delinquencies File

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Monthly Delinquencies File

SAVE Monthly Delinquencies File TO DATASTORE

***HOA Structure Chart Page 6 Module Pseudocode***

**Read Selected Association Name**

INPUTS: Association Names

OUTPUTS: Selected Association Name, Association Names

PSEUDOCODE:

RETRIEVE Selected Association Name

RETURN Selected Association Name

**Get Association Names**

INPUTS: N/A

OUTPUTS: Association Names

PSEUDOCODE:

RETRIEVE Association Names FROM INPUT FORM

RETURN Association Names

**Display Association Names**

INPUTS: Association Names

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Association Names

PRINT Association Names

**Read Selected Month**

INPUTS: Month Names

OUTPUTS: Month Names, Selected Month

PSEUDOCODE:

RETRIEVE Selected Month FROM INPUT FORM

RETURN Selected Month

**Get Month Names**

INPUTS: N/A

OUTPUTS: Month Names

PSEUDOCODE:

RETRIEVE Month Names FROM INPUT FORM

RETURN Month Names

**Display Month Names**

INPUTS: Month Names

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Month Names

PRINT Month Names

**Get Matching Monthly Delinquencies File**

INPUTS: Selected Month

OUTPUTS: Matching Monthly Delinquencies File

PSEUDOCODE:

RETRIEVE Selected Month

COMPARE Selected Month WITH DATASTORE

RETURN Matching Monthly Delinquencies File

**Display Matching Monthly Delinquencies File**

INPUTS: Matching Monthly Delinquencies File

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Matching Monthly Delinquencies File

PRINT Matching Monthly Delinquencies File

**Calculate Delinquent Amount Due**

INPUTS: Selected Association Name, Late Penalties, Total Amount Due, Late Fee

OUTPUTS: Selected Association Name, Late Penalties, Total Amount Due, Delinquent Amount Due

PSEUDOCODE:

RETRIEVE Selected Association Name

CALL Get Late Penalties

CALL Get Total Amount Due

CALL Calculate Late Fee

RETURN Delinquent Amount Due

**Get Late Penalties**

INPUTS: Selected Association Name, Association Bylaws

OUTPUTS: Selected Association Name, Late Penalties

PSEUDOCODE:

RETRIEVE Selected Association name

CALL Get Association ByLaws

COMPARE Association ByLaws

RETURN Late Penalties

**Get Association ByLaws**

INPUTS: Selected Association Name

OUTPUTS: Association ByLaws

PSEUDOCODE:

RETRIEVE Selected Association Name

COMPARE Selected Association Name FROM DATASTORE

RETURN Association ByLaws

**Get Total Amount Due**

INPUTS:.,N/A

OUTPUTS: Total Amount Due

PSEUDOCODE:

**Calculate Late Fee**

INPUTS: Total Amount Due, Late Penalties

OUTPUTS: Late Fee

PSEUDOCODE:

RETRIEVE Total Amount Due

RETRIEVE Late Penalties

CALCULATE Total Amount Due + Late Penalties

RETURN Late Fee

**Get All Records to be Made into Letters**

INPUTS: Delinquent Amount Due, Consolidated Delinquency Records

OUTPUTS: Delinquent Amount Due, Letter Records

PSEUDOCODE:

RETRIEVE Delinquent Amount Due

CALL Get Consolidated Delinquency Records

RETURN Letter Records

**Get Consolidated Delinquency Records**

INPUTS: Delinquent Amount Due, Duplicate Delinquency Records, Outstanding Late Payment Summary

OUTPUTS: Delinquent Amount Due, Duplicate Delinquency Records, Consolidated Delinquency Records

PSEUDOCODE:

RETRIEVE Delinquent Amount Due  
CALL Get Duplicate Delinquency Records  
CALL Make Outstanding Late Payment Summary  
RETURN Consolidated Delinquency Records

**Get Duplicate Delinquency Records**

INPUTS: N/A

OUTPUTS: Duplicate Delinquency Records

PSEUDOCODE:

**Make Outstanding Late Payment Summary**

INPUTS: Delinquent Amount Due, Duplicate Delinquent Records, Duplicate Delinquent Amount Due

OUTPUTS: Delinquent Amount Due, Duplicate Delinquent Records, Outstanding Late Payment Summary

PSEUDOCODE:

**Calculate Duplicate Records Delinquent Amount Due**

INPUTS: Delinquent Amount Due, Duplicate Delinquent Records

OUTPUTS: Duplicate Delinquent Amount Due

PSEUDOCODE:

**Make Delinquency Letters**

INPUTS: Letter Records

OUTPUTS: Delinquency Letter

PSEUDOCODE:

RETRIEVE Letter Records

RETURN Delinquency Letter

**Print Delinquency Letters**

INPUTS: Delinquency Letter

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Delinquency Letter

PRINT Delinquency Letter

***HOA Structure Chart Page 7 Module Pseudocode***

**Display Interface**

INPUTS: N/A

OUTPUTS: N/A

PSEUDOCODE:

PRINT Violation Interface

**Create Complaint File**

INPUTS: Selected Association Name, Association CC&Rs, Complaint Information,  
Complaint File

OUTPUTS: Selected Association Name, Complaint Information, Complaint File

PSEUDOCODE:

RETRIEVE Selected Association Name, Association CC&Rs, Complaint Information,  
Complaint File

CREATE Complaint File

RETURN Selected Association Name, Complaint Information, Complaint File

**Read Selected Association Name**

INPUTS: Association Names

OUTPUTS: Selected Association Name, Association Names

PSEUDOCODE:

RETRIEVE Selected Association Name

RETURN Selected Association Name

### **Get Association Names**

INPUTS: N/A

OUTPUTS: Association Names

PSEUDOCODE:

RETRIEVE Association Names FROM INPUT FORM

RETURN Association Names

### **Display Association Names**

INPUTS: Association Names

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Association Names

PRINT Association Names

### **Get Association CC&Rs**

INPUTS: Selected Association Name

OUTPUTS: Association CC&Rs

PSEUDOCODE: (*\*\*The CC&Rs come from the association bylaws, so you must first get those*)

RETRIEVE Selected Association Name

CREATE Association CC&R's

COMPARE Selected Association Name FROM DATASTORE Association ByLaws  
RETURN Association CC&R's

**Get Complaint Information**

INPUTS: Complaint From, Complaint Recipient, Date Received, Complaint, Complaint  
Validity

OUTPUTS: Complaint Information

PSEUDOCODE:

CREATE Complaint Information

CALL Read Complaint From Input Field

CALL Read Complaint Recipient input field

CALL Read Date Received input field

CALL Read Complaint input Field

CALL Get Complaint Validity

RETURN Complaint iInformation

**Read Complaint From input field**

INPUTS: N/A

OUTPUTS: Complaint From

PSEUDOCODE:

RETURN Complaint From

**Read Complaint Recipient input field**

INPUTS: N/A

OUTPUTS: Complaint Recipient

PSEUDOCODE:

RETURN Complaint Recipient

**Read Date Received input field**

INPUTS: N/A

OUTPUTS: Date Received

PSEUDOCODE:

READ Date Received

**Read Complaint input field**

INPUTS: N/A

OUTPUTS: Complaint

PSEUDOCODE:

RETURN Complaint

**Get Complaint Validity**

INPUTS: Valid Complaint, Invalid Complaint

OUTPUTS: Complaint Validity

PSEUDOCODE:

RETRIEVE Valid Complaint

RETRIEVE Invalid Complaint

CHECK Validity Radio Button

RETURN Complaint Validity

**Make Valid**

INPUTS: Violated CC&Rs

OUTPUTS: Valid Complaint

PSEUDOCODE:

RETRIEVE Violated CC&R's

CALL Read Violated CC&R's Input Field

RETURN Valid Complaint

**Read Violated CC&Rs input field**

INPUTS: N/A

OUTPUTS: Violated CC&Rs

PSEUDOCODE:

RETURN Violated CC&R's

**Make Invalid**

INPUTS: Reasons for Invalidity

OUTPUTS: Invalid Complaint

PSEUDOCODE:

RETRIEVE Reason for Invalidity

CALL Read Reasons for Invalidity Input Field

RETURN Invalid Complaint

**Read Reasons for Invalidity input field**

INPUTS: N/A

OUTPUTS: Reasons for Invalidity

PSEUDOCODE:

RETURN Reasons for Invalidity

**Make Complaint File**

INPUTS: Complaint Information

OUTPUTS: Complaint File

PSEUDOCODE:

RETRIEVE Complaint Information

CREATE Complaint File  
RETURN Complaint File

**Put Complaint File**

INPUTS: Complaint File

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Complaint File  
SAVE Complaint File

**Create Violation Notice**

INPUTS: Select Complaint Information, Valid Complaint File, Penalty Warnings,  
Violation Notice

OUTPUTS: Select Complaint Information, Valid Complaint File, Penalty Warnings,  
Violation Notice

PSEUDOCODE:

**Get Select Complaint Information**

INPUTS: Above Complaint File, Different Complaint File, Complaint Recipient Mailing  
Information

OUTPUTS: Above Complaint File, Different Complaint File, Select Complaint  
Information

PSEUDOCODE:

**Read Above Complaint File**

INPUTS: N/A

OUTPUTS: Above Complaint File

PSEUDOCODE:

CALL Get Matching Complaint File

CREATE Above Complaint File  
RETURN Above Complaint File

**Get Different Complaint File**

INPUTS: Complaint Search, Matching Complaint File

OUTPUTS: Complaint Search, Matching Complaint File, Different Complaint File

PSEUDOCODE:

**Read Complaint Search input field**

INPUTS: N/A

OUTPUTS: Complaint Search

PSEUDOCODE:

RETURN Complaint Search

**Get Matching Complaint File**

INPUTS: Complaint Search

OUTPUTS: Matching Complaint File

PSEUDOCODE:

RETRIEVE Complaint Search

COMPARE Complaint Search FROM DATASTORE

CREATE Matching Complaint File

RETURN Matching Complaint File

**Get Complaint Recipient Mailing Information**

INPUTS: Above Complaint File, Different Complaint File, Complaint Recipient Name,  
Complaint Recipient Account

OUTPUTS: Complaint Recipient Name, Complaint Recipient Mailing Information

PSEUDOCODE:

**Read Complaint Recipient Name**

INPUTS: Above Complaint File, Different Complaint File

OUTPUTS: Complaint Recipient Name

PSEUDOCODE:

**Get Complaint Recipient Account**

INPUTS: Complaint Recipient Name

OUTPUTS: Complaint Recipient Account

PSEUDOCODE:

**Display Complaint Information**

INPUTS: Select Complaint Information

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Select Complaint Information

PRINT Select Complaint Information

**Get Valid Complaint File**

INPUTS: Select Complaint Information, Textbox Validity

OUTPUTS: Select Complaint Information, Valid Complaint File

PSEUDOCODE:

**Read Complaint Validity Textbox**

INPUTS: Select Complaint Information

OUTPUTS: Textbox Validity

PSEUDOCODE:

RETRIEVE Select Complaint Information

RETURN Textbox Validity

**Get Penalty Warnings**

INPUTS: Penalty Warnings Input, Import File

OUTPUTS: Penalty Warnings

PSEUDOCODE:

**Read Penalty Warnings input field**

INPUTS: N/A

OUTPUTS: Penalty Warnings Input

PSEUDOCODE:

RETURN Penalty Warnings

**Read Import File**

INPUTS: Penalty Warnings File

OUTPUTS: Import File

PSEUDOCODE:

**Get Penalty Warnings File**

INPUTS: N/A

OUTPUTS: Penalty Warnings File

PSEUDOCODE:

**Make Violation Notice**

INPUTS: Valid Complaint File, Penalty Warnings

OUTPUTS: Violation Notice

PSEUDOCODE:

RETRIEVE Valid Complaint File

RETRIEVE Penalty Warnings

RETURN Violation Notice

**Put Violation Notice**

INPUTS: Violation Notice

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Violation notice

SAVE Violation Notice

**Print Violation Notice**

INPUTS: Violation Notice

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Violation Notice

PRINT Violation Notice

***HOA Structure Chart Page 8 Module Pseudocode***

**Create Second Notice**

INPUTS: Selected Old Notice File, Non-Satisfactory Notice, Followup Notice  
Information, Followup Violation Notice

OUTPUTS: Selected Old Notice File, Non-Satisfactory Notice, Followup Notice  
Information, Followup Violation Notice

PSEUDOCODE:

RETRIEVE Selected Old Notice File, Non-Satisfactory Notice, Followup Notice  
Information, Followup Violation Notice

CREATE Second Notice

RETURN Selected Old Notice File, Non-Satisfactory Notice, Followup Notice  
Information, Followup Violation Notice

### **Track Notices**

INPUTS: Selected Association Name, Current Date, Old Violation Notices, Selected Old Notice, Selected Old Notice File

OUTPUTS: Selected Association Name, Current Date, Old Violation Notices, Selected Old Notice, Selected Old Notice File

#### PSEUDOCODE:

RETRIEVE Selected Association Name, Current Date, Old Violation Notices, Selected Old Notice, Selected Old Notice File

IF Current Date > Old Violation Notices

THEN

RETURN Second Notice of Violation

### **Read Selected Association Name**

INPUTS: Association Names

OUTPUTS: Selected Association Name, Association Names

#### PSEUDOCODE:

RETRIEVE Selected Association Name

RETURN Selected Association Name

### **Get Association Names**

INPUTS: N/A

OUTPUTS: Association Names

#### PSEUDOCODE:

RETRIEVE Association Names FROM INPUT FORM

RETURN Association Names

### **Display Association Names**

INPUTS: Association Names

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Association Names

PRINT Association Names

**Read Current Date input field**

INPUTS: N/A

OUTPUTS: Current Date

PSEUDOCODE:

RETURN Current Date

**Get Old Violation Notices**

INPUTS: Selected Association Name, Current Date, Association Violation Notices,  
Notice Send Dates, Number of Days Past Sent Date

OUTPUTS: Selected Association Name, Current Date, Association Violation Notices,  
Notice Send Dates, Old Violation Notices

PSEUDOCODE:

RETRIEVE Selected Association Name, Current Date, Association Violation Notices,  
Notice Send Dates, Number of Days Past Sent Date

RETURN Selected Association Name, Current Date, Association Violation Notices,  
Notice Send Dates, Old Violation Notices

**Get Association Violation Notices**

INPUTS: Selected Association Name

OUTPUTS: Association Violation Notices

PSEUDOCODE:

RETRIEVE Selected Association Name  
CREATE Association Violation Notices  
RETURN Association Violation Notices

**Get Notice Sent Dates**

INPUTS: Association Violation Notices

OUTPUTS: Notice Sent Dates

PSEUDOCODE:

RETRIEVE Association Violation Notices  
CREATE Notice Sent Dates  
RETURN Notice Sent Dates

**Calculate Number of Days Past Sent Date**

INPUTS: Notice Sent Dates, Current Date

OUTPUTS: Number of Days Past Sent Date

PSEUDOCODE:

RETRIEVE Notice Sent Dates, Current Date  
 $\text{Current Date} - \text{Notice Sent Date} = \text{Number of Days Past Sent Date}$   
RETURN Number of Days Past Sent Date

**Display Old Violation Notices**

INPUTS: Old Violation Notices

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Old Violation Notices  
PRINT Old Violation Notices

**Read Selected Old Notice**

INPUTS: N/A

OUTPUTS: Selected Old Notice

PSEUDOCODE:

RETURN Selected Old Notice

**Get Selected Old Notice File**

INPUTS: Selected Old Notice

OUTPUTS: Selected Old Notice File

PSEUDOCODE:

RETRIEVE Selected Old Notice

CREATE Selected Old Notice File

RETURN Selected Old Notice File

**Display Selected Old Notice File Information**

INPUTS: Selected Old Notice File

OUTPUTS: N/A

PSEUDOCODE: (*\*\*Make sure to only show the relevant information*)

RETRIEVE Selected Old Notice File

PRINT Selected Old Notice File Information

**Display Selected Old Notice File**

INPUTS: Selected Old Notice File

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Selected Old Notice File

PRINT Selected Old Notice File

**Get Non-Satisfactory Compliance Notice**

INPUTS: Selected Old Notice File, Compliance Status

OUTPUTS: Selected Old Notice File, Non-Satisfactory Notice

PSEUDOCODE:

RETRIEVE Selected Old Notice File, Compliance Status

IF Compliance Status = FALSE

RETURN Selected Old Notice File, Non-Satisfactory Notice

**Get Compliance Status**

INPUTS: Selected Old Notice File, Satisfactory Status, Non-Satisfactory Status

OUTPUTS: Selected Old Notice File, Satisfactory Status, Compliance Status

PSEUDOCODE:

RETRIEVE Selected Old Notice File, Satisfactory Status, Non-Satisfactory Status

CREATE Compliance Status

RETURN Selected Old Notice File, Satisfactory Status, Compliance Status

**Make Notice Satisfactory Status**

INPUTS: Selected Old Notice File

OUTPUTS: Satisfactory Status

PSEUDOCODE:

RETRIEVE Selected Old Notice File

CREATE Satisfactory Status

**Make Notice Non-Satisfactory Status**

INPUTS: Selected Old Notice File

OUTPUTS: Non-Satisfactory Status

PSEUDOCODE:

RETRIEVE Selected Old Notice File  
CREATE Non-Satisfactory Status  
RETURN Non-Satisfactory Status

**Put Satisfactory Old Notice**

INPUTS: Satisfactory Status

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Satisfactory Status  
STORE Satisfactory Status

**Get Followup Notice Information**

INPUTS: Non-Satisfactory Status, Non-Satisfactory Status File, Upcoming Board of Directors Meeting Date, Non-Compliance Penalties

OUTPUTS: Non-Satisfactory Status, Followup Notice Information

PSEUDOCODE:

RETRIEVE Non-Satisfactory Status, Non-Satisfactory Status File, Upcoming Board of Directors Meeting Date, Non-Compliance Penalties  
CREATE Non-Satisfactory Status, Followup Notice Information  
RETURN Non-Satisfactory Status, Followup Notice Information

**Get Non-Satisfactory Notice File**

INPUTS: Non-Satisfactory Status

OUTPUTS: Non-Satisfactory Status File

PSEUDOCODE:

RETRIEVE Non-Satisfactory Status  
CREATE Non-Satisfactory Status File

RETURN Non-Satisfactory Status File

**Read Upcoming Board of Directors Meeting Date input field**

INPUTS: N/A

OUTPUTS: Upcoming Board of Directors Meeting Date

PSEUDOCODE:

RETURN Upcoming Board of Directors Meeting Data

**Get Non-Compliance Penalties**

INPUTS: Non-Compliance Penalty Warnings Input, Import File

OUTPUTS: Non-Compliance Penalties

PSEUDOCODE:

RETRIEVE Non-Compliance Penalty Warnings Input, Import File

CREATE Non-Compliance Penalties

RETURN Non-Compliance Penalties

**Read Non-Compliance Penalty Warnings input field**

INPUTS: N/A

OUTPUTS: Non-Compliance Penalty Warnings Input

PSEUDOCODE:

RETURN Non-Compliance Penalty Warnings Input

**Read Import File**

INPUTS: Non-Compliance Penalty Warnings File

OUTPUTS: Import File

PSEUDOCODE:

RETRIEVE Non-Compliance Penalty Warnings File

CREATE Import File

RETURN Import file

**Read Non-Compliance Penalty Warnings input field**

INPUTS: N/A

OUTPUTS: Non-Compliance Penalty Warnings File

PSEUDOCODE:

RETURN Non-Compliance Penalty Warnings File

**Make Followup Violation Notice**

INPUTS: Followup Notice Information

OUTPUTS: Followup Violation Notice

PSEUDOCODE:

RETRIEVE Followup Notice Information

CREATE Followup Violation Notice

RETURN Followup Violation Notice

**Put Followup Violation Notice**

INPUTS: Followup Violation Notice

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Followup Violation Notice

STORE Followup Violation Notice

**Print Followup Violation Notice**

INPUTS: Followup Violation Notice

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Followup Violation Notice

PRINT Followup Violation Notice

*HOA Structure Chart Page 9 Module Pseudocode*

**Display Interface**

INPUTS: Current Month

OUTPUTS: Current Month

PSEUDOCODE:

RETRIEVE Current Month FROM CALL DisplayCurrentMonthMaintenanceCalendar()

CALL DisplayCurrentMonthContractList()

CALL MaintainExistingContract()

CALL AddNewMaintenanceContract()

RETURN Current Month

**Display Current Month Maintenance Calendar**

INPUTS: Maintenance Calendar, Current Month

OUTPUTS: Current Month

PSEUDOCODE:

RETRIEVE Current Month

CALL MakeCurrentMonthContractSummariesList(CurrentMonth)

RETURN Contract Summary List

**Get Maintenance Calendar**

INPUTS: N/A

OUTPUTS: Maintenance Calendar

PSEUDOCODE:

RETRIEVE Maintenance Calendar FROM INPUT FORM  
RETURN Maintenance Calendar

**Get Current Month**

INPUTS: N/A

OUTPUTS: Current Month

PSEUDOCODE:

RETRIEVE Current Month FROM SYSTEM CLOCK  
RETURN Current Month

**Display Current Month Contract Summaries List**

INPUTS: Current Month, Contract Summaries List

OUTPUTS: Current Month, Contract Summaries List

PSEUDOCODE:

RETRIEVE Current Month  
CALL Make Current Month Contract Summaries List  
RETURN Contract Summaries List

**Make Current Month Contract Summaries List**

INPUTS: Current Month, Month Contracts, Contract Information, Contract Summary

OUTPUTS: Current Month, Month Contracts, Contract Information, Contract Summaries  
List

PSEUDOCODE:

RETRIEVE Current Month  
CALL Get Month Contracts  
CALL Get Contract Information

CALL Make Contract Summary  
RETURN Maintenance Calendar

**Get Month Contracts**

INPUTS: Current Month

OUTPUTS: Month Contracts

PSEUDOCODE:

RETRIEVE Current Month

COMPARE Current Month WITH DATASTORE

RETURN Month Contracts

**Get Contract Information**

INPUTS: Month Contracts

OUTPUTS: Contract information

PSEUDOCODE:

RETRIEVE Month Contracts

COMPARE Current Month WITH DATASTORE

RETURN Contract Information

**Make Contract Summary**

INPUTS: Contract Information, Summary Details

OUTPUTS: Contract Information, Contract Summary

PSEUDOCODE:

RETRIEVE Contract Information

CALL Get Summary Details

RETURN Contract Summary

**Get Summary Details**

INPUTS: Contract Information

OUTPUTS: Summary Details

PSEUDOCODE:

RETRIEVE Contract information

RETURN Contract Summary

**Maintain Existing Contract**

INPUTS: Contract Summaries List, Selected Contract File

OUTPUTS: Contract Summaries List, Selected Contract File

PSEUDOCODE:

**Get Existing Contract**

INPUTS: Contract Summaries List, Selected Contract, Selected Contract File

OUTPUTS: Contract Summaries List, Selected Contract, Selected Contract File

PSEUDOCODE:

RETRIEVE Contract Summaries List

CALL Read Selected Contract

CALL Get Selected Contract File

RETURN Selected Contract File

**Read Selected Contract**

INPUTS: Contract Summaries List

OUTPUTS: Selected Contract

PSEUDOCODE:

RETRIEVE Contract Summaries List

RETURN Selected Contract

**Get Selected Contract File**

INPUTS: Selected Contract

OUTPUTS: Selected Contract File

PSEUDOCODE:

RETRIEVE Selected Contract

RETURN Selected Contract File

**Display Selected Contract File**

INPUTS: Selected Contract File

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Selected Contract File

PRINT Selected Contract File

**Edit Selected Contract File**

INPUTS: Selected Contract File, Revised Information

OUTPUTS: Revised Information

PSEUDOCODE:

RETRIEVE Selected Contract File

CALL Read Changed Information Input Fields

CALL Update Selected Contract File

**Read Changed Information Input Fields**

INPUTS: N/A

OUTPUTS: Revised Information

PSEUDOCODE:

RETURN Revised Information FROM INPUT FORM

**Update Selected Contract File**

INPUTS: Revised Information

OUTPUTS: Updated Contract File

PSEUDOCODE:

RETRIEVE Revised Information

CALL Put Updated Contract File

**Put updated Contract File**

INPUTS: Updated Contract File

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Updated Contract File

SAVE Updated Contract File IN DATASTORE

**Add New Maintenance Contract**

INPUTS: New Contract File

OUTPUTS: New Contract File

PSEUDOCODE:

RETRIEVE New Contract File FROM CALL MT2

CALL AddMaintenanceBidRequest()

RETURN New Contract File

**Add Maintenance Bid Request**

INPUTS: Maintenance Bid Request File

OUTPUTS: Maintenance Bid Request File

PSEUDOCODE:

**Make Maintenance Bid Request File**

INPUTS: Selected Association Name, Requesting Homeowner, Requested Maintenance

OUTPUTS: Maintenance Bid Request File

PSEUDOCODE:

RETRIEVE Selected Association Name FROM CALL GetRequestAssociation()

RETRIEVE Requested Homeowner FROM CALL ReadRequestingHomeownerField()

RETRIEVE Requested Maintenance FROM CALL

ReadRequestedMaintenanceInputField()

RETURN Maintenance Bid Request File

**Get Request Association**

INPUTS: Association Names, Selected Association Name

OUTPUTS: Association Names, Selected Association Name

PSEUDOCODE:

RETRIEVE Association Names FROM CALL GetAssociationNames()

RETRIEVE Selected Association Names FROM CALL ReadSelectedAssocaitonNames()

RETURN Association Names, Selected Association Names

**Get Association Names**

INPUTS: N/A

OUTPUTS: Association Names

PSEUDOCODE:

RETRIEVE Association Names FROM Association DATASTORE

RETURN Association Names

**Display Association Names**

INPUTS: Association Names

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Association Names

PRINT Association Names

**Read Selected Association Names**

INPUTS: Association Names

OUTPUTS: Selected Association Names

PSEUDOCODE:

RETRIEVE Association Names

RETURN Selected Association Names

**Read Requesting Homeowner Input Field**

INPUTS: N/A

OUTPUTS: Requesting Homeowner

PSEUDOCODE:

RETRIEVE Requesting Homeowner FROM INPUT

RETURN Requesting Homeowner

**Read Requested Maintenance Input Field**

INPUTS: N/A

OUTPUTS: Requested Maintenance

PSEUDOCODE:

RETRIEVE Requested Maintenance FROM INPUT

RETURN Requested Maintenance

**Put Requested Maintenance Input Field**

INPUTS: Bid Request File

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Bid Request File

STORE Bid Request File IN DATASTORE Maintenance

**Put New Contract File**

INPUTS: New Contract file

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE New Contract File

STORE New Contract File IN DATASTORE Maintenance

*HOA Structure Chart Page 10 Module Pseudocode*

**Get Art's and Susan's Schedule**

INPUTS: N/A

OUTPUTS: Art and Susan Schedule

PSEUDOCODE:

RETRIEVE Art and Susan Schedule

RETURN Art and Susan Schedule

**Display Art's Schedule**

INPUTS: Art's Schedule

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Art Schedule

PRINT Art Schedule

**Display Susan's Schedule**

INPUTS: Susan 's Schedule

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Susan Schedule

RETURN Susan Schedule

**Get Reminder Information**

INPUTS: Completed Contract File, Completed Contract End Date, Check Date, Check Time, Reminder Recipient

OUTPUTS: Check Reminder Information

PSEUDOCODE:

**Read Completed Contract File input field**

INPUTS: N/A

OUTPUTS: Completed Contract File

PSEUDOCODE:

RETURN Completed Contract File

**Read Completed Contract File End Date input field**

INPUTS: N/A

OUTPUTS: Completed Contract End Date

PSEUDOCODE:

RETURN Completed Contract End Date

**Read Check Date input field**

INPUTS: N/A

OUTPUTS: Check Date

PSEUDOCODE:

RETURN Check Date

**Read Check Time input field**

INPUTS: N/A

OUTPUTS: Check Time

PSEUDOCODE:

RETURN Check Time

**Get Reminder Recipient**

INPUTS: Susan as Recipient, Art as Recipient

OUTPUTS: Reminder Recipient

PSEUDOCODE:

RETRIEVE Susan As Recipient, Art as Recipient

CREATE Reminder Recipient

PRINT Reminder Recipient

**Read Susan as Recipient**

INPUTS: N/A

OUTPUTS: Susan as Recipient

PSEUDOCODE:

RETURN Susan as Recipient

**Read Art as Recipient**

INPUTS: N/A

OUTPUTS: Art as Recipient

PSEUDOCODE:

RETURN Art as Recipient

**Make Maintenance Check Reminder**

INPUTS: Check Reminder Information

OUTPUTS: Maintenance Check Reminder

PSEUDOCODE:

RETRIEVE Chem Reminder Information

CREATE Maintenance Check Reminder

RETURN Maintenance Check Reminder

**Put Maintenance Check Reminder**

INPUTS: Maintenance Check Reminder

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Maintenance Check Reminder

STORE Maintenance Check reminder

**Make New Contract File**

INPUTS: New Contract Information, Contract Employees

OUTPUTS: New Contract File

PSEUDOCODE:

RETRIEVE New Contact Information, Contract Employees

CREATE New Contract File

RETURN New Contract File

**Get New Contract Information**

INPUTS: Homeowner Name, Selected Association Name, Contracted Maintenance,  
Project Dates, Estimated Number of Employees

OUTPUTS: New Contract Information

PSEUDOCODE:

RETRIEVE Homeowner Name, Selected Association Name, Contract Maintenance,  
Project Dates, Estimated Number of employees

CREATE New Contract Information

RETURN New Contract Information

**Read Homeowner Name input field**

INPUTS: N/A

OUTPUTS: Homeowner Name

PSEUDOCODE:

RETURN Homeowner Name

**Read Contracted Maintenance input field**

INPUTS: N/A

OUTPUTS: Contracted Maintenance

PSEUDOCODE:

RETURN Contract Maintenance

**Get Project Dates**

INPUTS: Date Fields, Estimated Duration

OUTPUTS: Project Dates

PSEUDOCODE:

RETRIEVE Date Fields, Estimated Duration

CREATE Project Dates

RETURN Project Dates

**Read Project Start Date and Special Dates input fields**

INPUTS: N/A

OUTPUTS: Date Fields

PSEUDOCODE:

RETURN Date Fields FROM INPUT

**Read Estimated Project Duration input field**

INPUTS: N/A

OUTPUTS: Estimated Duration

PSEUDOCODE:

RETURN Estimated Duration FROM INPUT

**Read Estimated Number of Needed Employees input field**

INPUTS: N/A

OUTPUTS: Estimated Number of Employees

PSEUDOCODE:

RETURN Estimated Number of Employees FROM INPUT

**Get Contract Employees**

INPUTS: Contract Assigned Employee

OUTPUTS: Contract Employees

PSEUDOCODE:

RETRIEVE Contract Assigned Employee

CREATE Contract Employees

RETURN Contract Employees

**Assign Employee to Maintenance Contract**

INPUTS: New Input Fields, Selected Employee Name, Selected Employee Information,  
Contract Assigned Employee

OUTPUTS: Contract Assigned Employee

PSEUDOCODE:

RETRIEVE New Input Fields, Selected Employee Name, Selected Employee  
Information, Contract Assigned Employee

CREATE Contract Assigned Employee

RETURN Contract Assigned Employee

**Get Selected Employee Name**

INPUTS: Employee Names, Selected Employee Name

OUTPUTS: Employee Names, Selected Employee Name

PSEUDOCODE:

RETRIEVE Employee Names, Selected Employee Name

RETURN Employee Names, Selected Employee Name

**Get List of Employee Names**

INPUTS: N/A

OUTPUTS: Employee Names

PSEUDOCODE:

RETURN Employee Names

**Display List of Employee Names**

INPUTS: Employee Names

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Employee Names

PRINT Employee Names

**Read Selected Employee Name**

INPUTS: Employee Names

OUTPUTS: Selected Employee Name

PSEUDOCODE:

RETRIEVE Employee Names

RETURN Selected Employee Name

**Get Selected Employee Information**

INPUTS: Selected Employee Name

OUTPUTS: Selected Employee Information

PSEUDOCODE:

RETRIEVE Selected Employee Name

CREATE Selected Employee Information

RETURN Selected Employee Information

**Display Selected Employee Information**

INPUTS: Selected Employee Information

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Selected Employee Information

PRINT Selected Employee Information

**Add Selected Employee to Contract**

INPUTS: Selected Employee Information

OUTPUTS: Contract Assigned Employee

PSEUDOCODE:

RETRIEVE Selected Employee Information

CREATE Contract Assigned Employee

RETURN Contract Assigned Employee

**Add Another Assigned Employee**

INPUTS: N/A

OUTPUTS: New Input Fields

PSEUDOCODE:

RETURN New Input Fields

***HOA Structure Chart Page 11 Module Pseudocode***

**Display Interface**

INPUTS: Newsletter File

OUTPUTS: Newsletter File

PSEUDOCODE:

RETRIEVE Newsletter File

PRINT Newsletter File

RETURN Newsletter File

**Record Newsletter Contract Specifications**

INPUTS: Selected Association Name, Selected Community, Newsletter Specifications,  
Newsletter Contract

OUTPUTS: Selected Association Name, Selected Community, Newsletter Specifications,  
Newsletter Contract

PSEUDOCODE:

RETRIEVE Selected Association Name, Selected Community, Newsletter Specifications,  
Newsletter Contract

STORE Selected Association Name, Selected Community, Newsletter Specifications,  
Newsletter Contract

RETURN Selected Association Name, Selected Community, Newsletter Specifications,  
Newsletter Contract

### **Read Selected Association Name**

INPUTS: Association Names

OUTPUTS: Selected Association Name, Association Names

PSEUDOCODE:

RETRIEVE Selected Association Name

RETURN Selected Association Name

### **Get Association Names**

INPUTS: N/A

OUTPUTS: Association Names

PSEUDOCODE:

RETRIEVE Association Names FROM INPUT FORM

RETURN Association Names

### **Display Association Names**

INPUTS: Association Names

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Association Names

PRINT Association Names

**Read Selected Community**

INPUTS: Community Names

OUTPUTS: Selected Community, Community Names

PSEUDOCODE:

RETRIEVE Community Names

RETURN Selected Community, Community Names

**Get Community Names**

INPUTS: N/A

OUTPUTS: Community Names

PSEUDOCODE:

RETURN Community Names

**Display Community Names**

INPUTS: Community Names

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Community Names

PRINT Community Names

**Get Newsletter Specification**

INPUTS: Selected Sections Checkboxes, Layout Specifications, Design Specifications,  
Selected Cycle Duration, Additional Comments

OUTPUTS: Newsletter Specifications

PSEUDOCODE:

RETRIEVE Selected Sections Checkboxes, Layout Specifications, Design Specifications,  
Selected Cycle Duration, Additional Comments

CREATE Newsletter Specifications

RETURN Newsletter Specifications

**Read Selected Sections Checkboxes**

INPUTS: Other Section Header

OUTPUTS: Other Checkbox, Selected Sections Checkboxes

PSEUDOCODE:

RETRIEVE Other Section Header

RETURN Other Checkbox, Selected Section Checkboxes

**Read Other Section Header input field**

INPUTS: Other Checkbox

OUTPUTS: Other Section Header

PSEUDOCODE:

RETRIEVE Other Checkbox

RETURN Other Section Header

**Read Layout Specifications input field**

INPUTS: N/A

OUTPUTS: Layout Specifications

PSEUDOCODE:

RETURN Layout Specifications

**Read Design Specifications input field**

INPUTS: N/A

OUTPUTS: Design Specifications

PSEUDOCODE:

RETURN Design Specifications

**Read Selected Newsletter Cycle Duration**

INPUTS: N/A

OUTPUTS: Selected Cycle Duration

PSEUDOCODE:

RETURN Selected Cycle Duration

**Display Newsletter Cycle Duration**

INPUTS: N/A

OUTPUTS: N/A

PSEUDOCODE: (\*\**Make sure to include getting the cycle durations*)

RETRIEVE cycle Duration FROM CALL ReadSelectedNewsletterCycleDuration

PRINT Cycle duration

**Read Additional Comments input field**

INPUTS: N/A

OUTPUTS: Additional Comments

PSEUDOCODE:

RETURN Additional Comments

**Make Newsletter Contract**

INPUTS: Selected Association Name, Selected Community, Newsletter Specifications

OUTPUTS: Newsletter Contract

PSEUDOCODE:

RETRIEVE Selected Association Name, Selected Community, Newsletter Specifications

CREATE Newsletter Contract

### **Put Newsletter Contract**

INPUTS: Newsletter Contract

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Newsletter Contract

STORE Newsletter Contract

### **Create Newsletter File**

INPUTS: Selected Newsletter Contract, Selected Newsletter Contract Information,  
Newsletter Information, Newsletter File

OUTPUTS: Selected Newsletter Contract, Selected Newsletter Contract Information,  
Newsletter Information, Newsletter File

PSEUDOCODE:

RETRIEVE Selected Newsletter Contract, Selected Newsletter Contract Information,  
Newsletter Information, Newsletter File

CREATE Newsletter File

RETURN Selected Newsletter Contract, Selected Newsletter Contract Information,  
Newsletter Information, Newsletter File

### **Read Selected Newsletter Contract**

INPUTS: Newsletter Contracts

OUTPUTS: Newsletter Contracts, Selected Newsletter Contract

PSEUDOCODE:

RETRIEVE Newsletter Contracts

RETURN Newsletter Contracts, Selected Newsletter Contract

**Get Newsletter Contracts**

INPUTS: N/A

OUTPUTS: Newsletter Contracts

PSEUDOCODE:

RETURN Newsletter Contracts

**Display Newsletter Contracts**

INPUTS: Newsletter Contracts

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Newsletter Contracts

**Display Selected Newsletter Contract Information**

INPUTS: Selected Newsletter Contract, Selected Newsletter Contract Information

OUTPUTS: Selected Newsletter Contract

PSEUDOCODE:

RETRIEVE Selected Newsletter Contract, Selected Newsletter Contract Information

CREATE Selected Newsletter Contract

RETURN Selected Newsletter Contract

**Get Selected Newsletter Contract Information**

INPUTS: Selected Newsletter Contract

OUTPUTS: Selected Newsletter Contract Information

PSEUDOCODE:

RETRIEVE Selected Newsletter Contract  
CREATE Selected Newsletter Contract Information  
RETURN Selected Newsletter Contract Information

**Edit Selected Newsletter Contract Information**

INPUTS: Selected Newsletter Contract, Revised Information

OUTPUTS: Revised Information

PSEUDOCODE:

RETRIEVE Selected Newsletter Contract, Revised Information  
CREATE Revised Information  
RETURN Revised Information

**Read Changed Information input fields**

INPUTS: N/A

OUTPUTS: Revised Information

PSEUDOCODE:

RETURN Revised Information

**Update Selected Newsletter Contract**

INPUTS: Revised Information

OUTPUTS: Updated Newsletter Contract

PSEUDOCODE:

RETRIEVE Revised Information  
CREATE Updated Newsletter Contract  
RETURN Updated Newsletter Contract

**Put Updated Newsletter Contract**

INPUTS: Updated Newsletter Contract

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Updated Newsletter Contract

**Get Newsletter Information**

INPUTS: Section Information File Import, Image File Import

OUTPUTS: Section Information File Import, Image File Import, Newsletter Information

PSEUDOCODE:

RETRIEVE Section Information File Import, Image File Import

CREATE Newsletter Information

Section Information File Import, Image File Import, Newsletter Information

**Read Section Information File Import**

INPUTS: Image File

OUTPUTS: Image File Import

PSEUDOCODE:

RETRIEVE Image File

CREATE Image File Import

RETURN Image File Import

**Read Image File Import**

INPUTS: Section Information File Import

OUTPUTS: Contract Assigned Employee

PSEUDOCODE:

RETRIEVE Section Information File Import

CREATE Contract Assigned Employee

RETURN Contract Assigned Employee

**Display Imported Files**

INPUTS: Section Information File Import, Image File Import

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Section Information File Import, Image File Import

PRINT Section Information File Import, Image File Import

**Make Newsletter File**

INPUTS: Selected Newsletter Contract, Newsletter Information

OUTPUTS: Newsletter File

PSEUDOCODE:

RETRIEVE Selected Newsletter Contract, Newsletter Information

CREATE Newsletter File

RETURN Newsletter File

**Put Newsletter File**

INPUTS: Newsletter File

OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Newsletter File

STORE Newsletter File

**Get Newsletter Upload**

INPUTS: Completed Newsletter File Import

OUTPUTS: Newsletter Upload

PSEUDOCODE:

RETRIEVE Completed Newsletter File Import  
CREATE Newsletter Upload  
RETURN Newsletter Upload

**Read Completed Newsletter File Import**

INPUTS: Completed Newsletter File  
OUTPUTS: Completed Newsletter File Import  
PSEUDOCODE:

RETRIEVE Completed Newsletter File  
CREATE Completed Newsletter File Import  
RETURN Completed Newsletter File Import

**Read Completed Newsletter File**

INPUTS: N/A  
OUTPUTS: Completed Newsletter File  
PSEUDOCODE:

RETURN Completed Newsletter File

**Put Newsletter**

INPUTS: Newsletter Upload  
OUTPUTS: N/A  
PSEUDOCODE:

RETRIEVE Newsletter Upload  
STORE Newsletter Upload

**Print Newsletter**

INPUTS: Newsletter Upload  
OUTPUTS: N/A

PSEUDOCODE:

RETRIEVE Newsletter Upload

PRINT Newsletter Upload

## **Appendix A**

### **Interface Specifications**

## **Interface Specifications**

The following interface specifications provide well-documented explanations for all of the interfaces created for Homeowners of America. These specifications correspond to the numerical order that they are presented in on the interface structure diagram. Each interface specification file consist of a detailed narrative, an interface prototype reference, and a basic test plan.

## Associations List Tab Interface Specifications

<p>Interface Name: Associations List Tab</p> <p>Type of Interface: <i>Form</i></p> <p>Interface Mode: <i>Screen</i></p>	<p>Interface Number: 0</p> <p>Date Created: 12/2/15</p>
<p><b>Description:</b></p> <p>Susan, Art, and Marget will use this interface.</p> <p>The interface will allow for the various associations details to be recorded.</p> <p>The interface will be used whenever a new association needs to be added, whenever and associations committee list needs to be added or edited, or current details about an association need to be accessed.</p> <p>This interface is required because storing and accessing the data contained in this interface is a key requirement of the system.</p> <p>Relevant sections of the form will be used to either input or view data whenever necessary. Certain parts of the form can be used for both input and output.</p> <p>The user characteristics are defined by managers who are familiar with the system.</p> <p>The task is inputting information into the system and viewing data already in the system..</p> <p>The system is a on a standalone local network PC</p> <p>The environment is employee desks at the HOA office.</p> <p>You leave the interface be either selecting another tab or closing the program with the X in the top right corner.</p> <p>Data flows included in this interface are:</p> <ol style="list-style-type: none"> <li>1. Ordered Association List</li> </ol>	

2. Updated Association List
3. Association Bylaws template
4. Late Penalty Options
5. Creation date
6. Association Information
7. Association Worksheet
8. Association File
9. Committee Member Names
10. Updated Committee Information
11. Association Name
12. Committee Names
13. Committee list printout
14. Committee List

This interface is derived from Process 1 (Organize and Update Ordered Association List); and Process 12 (Maintain Association Committee List), as well as all of their sub processes.

Fields on this form that are filled in by the system are:

1. Ordered Associations
  - a. Ordered Association List (List of Strings)
2. Selected Association Information
  - a. Association Name (String)
  - b. Association Size (int)
  - c. Required Services (String)
  - d. Association Bylaws (String)
  - e. Homeowner Accounts (Excel file)
3. Committee
  - a. Committee Name (Combo box option; String)
  - b. Committee Members (Combo box option)
4. Association Bylaws
  - a. Monthly Fee (int)

Fields on this form to be filled in by the user are:

1. Committee
  - a. New Committee Name (String)
  - b. New Member Name (String)
2. Add Associations
  - a. Association Name (String)
  - b. Association Size (Combo box option)
  - c. Required Services/ HOA Services (Check boxes)
  - d. Association Creation Date (int)
  - e. Payment Due By Date (int)
  - f. Late Payment Penalties (String)
  - g. Allowed Late Period (String)
  - h. Homeowner List Import File

Buttons used on this interface are:

1. Print Association List: Prints the selected Association List
2. Edit: Enable editing of the grayed out fields.
3. Select Committee: User indicates which association committee they would like to view from a list of all association committees
4. Select Member: User indicates which committee member they would like to view
5. Remove Committee: Deletes the selected committee from the datastore
6. Add Committee: Adds a new committee with the new committee name to the datastore
7. Remove Member: Delete the selected member from the selected committee
8. Add member: Add a member to the selected committee with the new member name
9. Print Committee List: Sends the committee list to the printer.
10. Save committee List: Saves the recent changes to the committee list

11. All buttons under Required services: Check boxes which mark the service next to it as required.
12. Calculate Monthly Fee: use the association bylaws to determine the monthly fee.
13. Browse: Browse the local PC for a file to import.
14. Import: Import the selected file
15. Reset: Clear all above fields
16. Save Association: Commit changes to the datastore

**Prototype Layout:**

Prototype can be viewed in Appendix \_\_\_.

**Test Plan:**

First, select an association from the list and click Print Association List. Then, select a committee from the drop down. Add a member, then delete the same member. Click print committee list, then save committee list. Next, enter an association name, choose the size as 10000, and select the checkbox for Manage financial Information. Click Calculate Monthly fee. Enter the current date as the creation date. Enter a date for payment due by date, enter test in the late payment penalties box, and the allowed late period box. Click browse and select a homeowner account file to import, then click import. Finally, click save association.

**Ordered Associations List Printout Interface Specifications**

<p>Interface Name: Ordered Association List Printout</p> <p>Type of Interface: <i>Report</i></p> <p>Interface Mode: <i>Paper</i></p>	<p>Interface Number: <i>0.1</i></p> <p>Date Created: 12/2/2015</p>
<p><b>Description:</b></p> <p>This report will be used by Art and Susan at HOA.</p> <p>The purpose of this report is to display an ordered list based on size of all the associations HOA manages.</p> <p>This report will be used whenever a new association is added and Art and Susan must figure out who will manage it. It may also be used whenever someone wants a list of all the associations.</p> <p>This report will be used in the Art and Susan's offices.</p> <p>This report is needed so Art and Susan are able to decide who is in charge of managing which associations. It also provides a quick overlook of the associations managed by HOA.</p> <p>This report will be completed by being printed out.</p> <p>The user characteristics are Art and Susan who understand how to read the list.</p> <p>The task is to provide Art and Susan with an ordered list of all the associations managed by HOA.</p> <p>The system is a standalone local network PC.</p> <p>The environment is Art and Susan's offices.</p> <p>You can leave this interface by throwing away the piece of paper.</p> <p>Data flows involved in this interface:</p>	

1. Ordered Association List
2. Updated Association List

This interface is derived from Process 1.2 (Update Ordered Association List).

The fields filled in by the system are:

1. Association Information
  - a. Association Name (String)
  - b. Association Size (String)
  - c. Required Services (List of Strings)

There are no buttons involved with this interface.

**Prototype Layout:**

Prototype can be viewed in Appendix \_\_.

**Test Plan:**

This report will be tested with a walkthrough evaluation. So, first the report must be printed out and then we must make sure that all associations are listed with the correct association size and required services.

## Association Committee Printout Interface Specifications

<p>Interface Name: Association Committee List Printout</p> <p>Type of Interface: <i>Report</i></p> <p>Interface Mode: <i>Paper</i></p>	<p>Interface Number: <i>0.2</i></p> <p>Date Created: 12/2/2015</p>
<p><b>Description:</b></p> <p>This report will be used by HOA office workers</p> <p>The purpose of this report is to display a list of association committees and their members.</p> <p>This report will be used daily because all HOA office workers will have the report on hand.</p> <p>This report will be used in the HOA offices.</p> <p>This report is needed so HOA office workers are able to answer any question a customer has when they call.</p> <p>This report will be completed by being printed out.</p> <p>The user characteristics are HOA office workers who understand the report layout.</p> <p>The task is to provide HOA office workers with a convenient reference to the association committees and members.</p> <p>The system is a standalone local network PC.</p> <p>The environment is the employee desks at the HOA offices.</p> <p>You can leave this interface by throwing away the piece of paper.</p> <p>Data flows involved in this interface:</p> <ol style="list-style-type: none"> <li>1. Committee List</li> <li>2. Committee List Printout</li> </ol> <p>This interface is derived from Process 12.4 (Print Out Committee List).</p>	

The fields filled in by the system are:

1. Association Name (String)
2. Committee Name(s) (String)
3. Member Names (List of Strings)

There are no buttons involved with this interface.

**Prototype Layout:**

Prototype can be viewed in Appendix \_\_\_.

**Test Plan:**

This report will be tested with a walkthrough evaluation. So, first the report must be printed out and then we must make sure that all the correct fields appear on the report as they do on the computer.

## Payment Tab Interface Specifications

<p>Interface Name: Payment Tab</p> <p>Type of Interface: <i>Form</i></p> <p>Interface Mode: <i>Screen</i></p>	<p>Interface Number: 1</p> <p>Date Created: 12/8/2015</p>
<p><b>Description:</b></p> <p>This interface will be used by all HOA office workers but mainly Rhonda and Margaret. The purpose of this interface is to manage all finances of HOA and the associations it manages by creating and printing different bills, and enabling workers to process payments as they come in.</p> <p>This interface will be used when the creation of a certain bill is needed and/or a payment needs to be processed.</p> <p>This interface will be used in the HOA office.</p> <p>This interface is needed because it makes managing all financial information more efficient and easier to access and edit. It also makes the creation of various types of bills easier and faster.</p> <p>This interface is completed with inputs from the mouse and keyboard.</p> <p>The user characteristics are HOA office workers who are familiar with the system.</p> <p>The task is to manage all financial information by creating bills and processing payments.</p> <p>The system is a standalone local network PC.</p> <p>The environment is employee desks at the HOA office.</p> <p>You can leave the interface by clicking the “X” in the upper right-hand corner of the screen.</p> <p>Data flows included in this interface are:</p> <ol style="list-style-type: none"> <li>1. Current Month</li> <li>2. Bill Due Date</li> </ol>	

3. Association Bylaws
4. Association Credit
5. Total Amount Due
6. Homeowner Accounts
7. Printed Bill
8. Due Payment
9. Payment Coupon
10. Date Received
11. Fiscal Year Worksheet
12. Association Worksheet
13. Association Bills
14. Incurred Bill
15. Unexpected Bill
16. Special Assessment Due Date
17. Special Assessment Bill

This interface is derived from Process 2 (Create Monthly Bill); Process 3 (Process Monthly Due Payment); Process 9 (Process Special Assessment); and Process 10 (Calculate Incurred Association Bill), as well as their sub processes.

Fields on this form that are filled in by the system are:

1. Process Bill Payment/ Process Special Assessment Payment
  - a. Account Found
    - i. Member Name (String)
    - ii. Member Association (String)
    - iii. Account Number (int)
2. Incurred Fee
  - a. Total Fee Amount (int)

Fields on this form to be filled in by the user are:

1. Create Monthly Bill

- a. Association Name (Combo box option)
- b. Month (Combo box option)
2. Process Bill Payment/ Process Special Assessment Payment
  - a. Date Received (int)
  - b. Payment Amount (int)
  - c. Member name (String)
  - d. Member Association (String)
  - e. Account Number (int)
  - f. Check Number (int)
  - g. No Payment Coupon (Check box)
  - h. Search Member Name (String)
3. Incurred Bill
  - a. Association Name (Combo box option)
  - b. Water Fee Amount (int)
  - c. Gas Fee Amount (int)
  - d. Electricity Fee Amount (int)
  - e. Trash Pickup Fee Amount (int)
  - f. Maintenance Fee Amount (int)
  - g. HOA Fee Amount (int)
4. Create Special Assessment Bill
  - a. Association nme (Combo box option)
  - b. Special Assessment Bill Amount (int)
  - c. Special Assessment Reason (String)
  - d. Date Bill Received (int)
  - e. Chosen Due Date (int)

Buttons used on this interface are:

1. Association Name: User indicates which association they would like to create and print a monthly bill for
2. Select Month: User indicates which month they would like the bill to portray

3. Create and Print Bill: Creates the monthly bill, saves it, and sends it to the printer to be printed off
4. Search: Searches all of the separate worksheets for the member name in the search field
5. Save Payment: Records the payment amount entered and saves the payment record
6. Reset: Clears all fields within the preceding section
7. Association Name: User indicates which association they would like to enter the fees for and get the incurred amount
8. Calculate Incurred Fee: Calculates the incurred fee based on the amounts entered in the input fields above
9. Send Bill to HOA: Creates the incurred bill and sends the total fee amount to the HOA office
10. Create and Save Bill: Creates the incurred bill and saves it to the datastore
11. Association Name: User indicates which association they would like to make a special assessment for
12. Calculate Special Charge: Calculates the special charge per homeowner which the specified association
13. Create and Print Special Bill: Creates the special assessment bill, saves it, and sends it to the printer to be printed out
- 14.

**Prototype Layout:**

Prototype can be viewed in Appendix \_\_.

**Test Plan:**

First the screen will go through a walkthrough evaluation to ensure that all fields are relevant to managing the financial information of HOA and the various homeowner associations it manages. It will then go through an interactive evaluation to make sure is understandable by users and to check the ease of use of the form.

## Monthly Bill Printout Interface Specifications

<p>Interface Name: Monthly Assessment Printout</p> <p>Type of Interface: <i>Report</i></p> <p>Interface Mode: <i>Paper</i></p>	<p>Interface Number: <i>1.1</i></p> <p>Date Created: 12/8/2015</p>
<p><b>Description:</b></p> <p>This report will be used by the homeowners in each association.</p> <p>The purpose of this report is to inform homeowners of the amount that is due at the end of the month.</p> <p>This report will be used once a month, the exact day depends on the association payment period.</p> <p>This report will be used in the homes of all homeowners.</p> <p>This report is needed so homeowners know how much they have to pay when their payment period is over and when the payment is due by.</p> <p>This report will be completed by being printed out.</p> <p>The user characteristics are homeowners who are able to understand the bill.</p> <p>The task is to inform homeowners of how much they must pay by a certain date.</p> <p>The system is a standalone local network PC.</p> <p>The environment is all homeowner houses.</p> <p>You can leave this interface by throwing away the piece of paper.</p> <p>Data flows involved in this interface:</p> <ol style="list-style-type: none"> <li>1. Printed Bill</li> <li>2. Homeowner Accounts</li> </ol> <p>This interface is derived from Process 2.4 (Create the Bill), as well as its sub processes.</p>	

The fields filled in by the system are:

1. Association Name (String)
2. Homeowner Name (String)
3. Homeowner Address (String)
4. Account Number (int)
5. Current Date (String)
6. Month (String)
7. Total Amount Due (int)
8. Next Meeting Date (int)
9. Next Meeting Time (int)
10. Next Meeting Location (String)
11. Late Penalty (int)
12. Date Due By (int)

There are no buttons involved with this interface.

**Prototype Layout:**

Prototype can be viewed in Appendix \_\_\_.

**Test Plan:**

This report will be tested with a walkthrough evaluation. So, first the report must be printed out and then we must make sure that all the correct fields appear on the report as they do on the computer.

## Incurred Bill Printout Interface Specifications

<p>Interface Name: Incurred Bill Printout</p> <p>Type of Interface: <i>Report</i></p> <p>Interface Mode: <i>Paper</i></p>	<p>Interface Number: <i>1.2</i></p> <p>Date Created: 12/8/2015</p>
<p><b>Description:</b></p> <p>This report will be used by the HOA office.</p> <p>The purpose of this report is to inform HOA of the incurred amount they must pay for a particular association and the individual fee amounts.</p> <p>This report will be used whenever an association sends in fees that HOA is responsible for paying.</p> <p>This report will be used in the HOA office.</p> <p>This report is needed so HOA is able to know how much they must pay out for every association, and to see what each associations' fees are.</p> <p>This report will be completed by being printed out.</p> <p>The user characteristics are HOA office workers who understand the report layout.</p> <p>The task is to inform HOA of the incurred amount that must be paid.</p> <p>The system is a standalone local network PC.</p> <p>The environment is the HOA offices.</p> <p>You can leave this interface by throwing away the piece of paper.</p> <p>Data flows involved in this interface:</p> <ol style="list-style-type: none"> <li>1. Association Bills</li> <li>2. Association Name</li> <li>3. Incurred Bill</li> </ol>	

This interface is derived from Process 10.2 (Create Incurred Bill), as well as its sub processes.

The fields filled in by the system are:

1. Association Name (String)
2. Water Fee Amount (int)
3. Gas Fee Amount (int)
4. Electricity Fee Amount (int)
5. Trash Pickup Fee Amount (int)
6. Maintenance Fee Amount (int)
7. HOA Fee Amount (int)
8. Total Incurred Amount (int)

There are no buttons involved with this interface.

**Prototype Layout:**

Prototype can be viewed in Appendix \_\_\_.

**Test Plan:**

This report will be tested with a walkthrough evaluation. So, first the report must be printed out and then we must make sure that all the correct fields appear on the report as they do on the computer.

## Special Assessment Printout Interface Specifications

<p>Interface Name: Special Assessment Printout</p> <p>Type of Interface: <i>Report</i></p> <p>Interface Mode: <i>Paper</i></p>	<p>Interface Number: <i>1.3</i></p> <p>Date Created: 12/8/2015</p>
<p><b>Description:</b></p> <p>This report will be used by the homeowners in an association with a special assessment. The purpose of this report is to inform homeowners of the special charge amount that is due at a date chosen by the association board of directors.</p> <p>This report will be used whenever an association submits a large unexpected bill to HOA. This report will be used in the homes of all homeowners in that particular association. This report is needed so homeowners are aware of the extra cost they are being charged, how much that cost is, and when it is due.</p> <p>This report will be completed by being printed out.</p> <p>The user characteristics are homeowners who are able to understand the bill. The task is to inform homeowners of how much they must pay by a certain date. The system is a standalone local network PC. The environment is homeowner houses within the particular association.</p> <p>You can leave this interface by throwing away the piece of paper.</p> <p>Data flows involved in this interface:</p> <ol style="list-style-type: none"> <li>1. Special Assessment Bill</li> <li>2. Special Assessment Due Date</li> </ol> <p>This interface is derived from Process 9.4 (Create Special Assessment Bill).</p> <p>The fields filled in by the system are:</p>	

1. Association Name (String)
2. Homeowner Name (String)
3. Homeowner Address (String)
4. Account Number (int)
5. Current Date (String)
6. Special Assessment Charge (int)
7. Amount Due (int)
8. Next Meeting Date (int)
9. Next Meeting Time (int)
10. Next Meeting Location (String)
11. Late Penalty (int)
12. Special Assessment Date Due By (int)

There are no buttons involved with this interface.

**Prototype Layout:**

Prototype can be viewed in Appendix \_\_\_.

**Test Plan:**

This report will be tested with a walkthrough evaluation. So, first the report must be printed out and then we must make sure that all the correct fields appear on the report as they do on the computer.

## Delinquency Tab Interface Specifications

<p>Interface Name: Delinquency Tab</p> <p>Type of Interface: <i>Form</i></p> <p>Interface Mode: <i>Screen</i></p>	<p>Interface Number: 2</p> <p>Date Created: 12/2/15</p>
<p><b>Description:</b></p> <p>This interface will be accessed by Margret, Rhonda, and Julie.</p> <p>This interface allows information about bill delinquencies to be entered and displayed.</p> <p>This interface will be used twice per week when Margaret processes the delinquencies, when Rhonda searches for and identifies duplicate delinquency letters, and when Julie submits the combined delinquency letters.</p> <p>The interface is required because these employees must have a way to access the delinquency letters.</p> <p>The interface is used to submit delinquency records, view who is delinquent, and submit the completed delinquency letters after duplicates have been combined, Depending on the task at hand, the relevant section of the form is used.</p> <p>The user characteristics are defined by office employees who are familiar with the system.</p> <p>The task is inputting information into the system and viewing records.</p> <p>The system is a on a standalone local network PC</p> <p>The environment is employee desks at the HOA office.</p> <p>To leave the interface, choose another tab, or close the program by clicking the X in the top right corner.</p> <p>Data flows included in this interface are:</p> <ol style="list-style-type: none"> <li>1. Delinquency letters       <ol style="list-style-type: none"> <li>a. Total Amount Due</li> </ol> </li> </ol>	

- b. Late Penalties
- c. Bill Due Date
- d. Allowed Late Period
2. Finalized Delinquency letters
3. List Reference Information
4. Delinquency records list
5. Association worksheet

This Interface is derived from Process 4 (Create and File Monthly Delinquency Records); Process 5 (Create Delinquency Letters); and Process 6 (Identify Duplicate Delinquency Letters), as well as all of the sub processes of these processes.

Fields on this form that are filled in by the system are:

1. Delinquent Homeowner Records
  - a. Month (String)
  - b. Homeowner Accounts (String)

Fields on this form to be filled in by the user are:

1. Association Name (Combo box option)
2. Current Date (int)
3. Month (Combo box option)

Buttons used on this interface are:

1. Association Name: User indicates which association they want to find delinquent records for
2. Find Delinquent Records: Displays all delinquent records matching the search criteria
3. Select Month: User indicates which month with delinquent records they would like to copy
4. Copy and Save Records: Exports a copy of the records

5. Association Name (Create Delinquency Letters): User indicates which association they want to create delinquency letters for
6. Select Month (Create Delinquency Letters): User indicates which month they would like to view delinquent records from
7. Find Delinquent Records List File: Browse for the file to be included
8. Consolidate Duplicate Records: Runs a script to combine any duplicate letters.
9. Print Delinquent Letters: Print out all of the finalized letters once the duplicates have been consolidated

**Prototype Layout:**

Prototype can be viewed in Appendix \_\_\_.

**Test Plan:**

Enter the name of the association you would like to retrieve letters for, and the current date. Then click find delinquency records. Then select January, and click copy and save records. Next, choose another association in the other drop down box. Enter January for the month. Click find Delinquency records list file. Then click consolidate duplicate records, then click print delinquent letters.

## Delinquency Letter Printout Interface Specifications

<p>Interface Name: Delinquency Letter Printout</p> <p>Type of Interface: <i>Report</i></p> <p>Interface Mode: <i>Paper</i></p>	<p>Interface Number: <i>2.1</i></p> <p>Date Created: 12/8/2015</p>
<p><b>Description:</b></p> <p>This report will be used by the delinquent homeowners.</p> <p>The purpose of this report is to inform homeowners that they are late on payments and tell them what they owe.</p> <p>This report will be assumed to be used daily by the homeowners, but will be printed off from HOA every 15, 30, 60, and 90 days.</p> <p>This report will be used in the homes of the delinquent homeowners.</p> <p>This report is needed so HOA is able to inform homeowners that they are late on payment, and so homeowners know how much they owe for being delinquent.</p> <p>This report will be completed by being printed out.</p> <p>The user characteristics are homeowners who are able to understand the letter.</p> <p>The task is to make homeowners aware of their delinquency status and inform them of the delinquent amount they owe.</p> <p>The system is a standalone local network PC.</p> <p>The environment is the delinquent homeowners' houses.</p> <p>You can leave this interface by throwing away the piece of paper.</p> <p>Data flows involved in this interface:</p> <ol style="list-style-type: none"> <li>1. Finalized Delinquency Letters</li> </ol> <p>This interface is derived from Process 6.2 (Create Consolidated Delinquency Letters); and Process 6.3 (Print Finalized Delinquency Letters), as well as their sub processes.</p>	

The fields filled in by the system are:

1. Current Date (String)
2. Homeowner Name (String)
3. Homeowner Address (String)
4. Association Name (String)
5. Number of Days Late (int)
6. Delinquent Month (String)
7. Delinquent Amount Due (int)
8. Total Amount Due (int)
9. Late Penalty (int)
10. Association Manager Name (String)

There are no buttons involved with this interface.

**Prototype Layout:**

Prototype can be viewed in Appendix \_\_\_.

**Test Plan:**

This report will be tested with a walkthrough evaluation. So, first the report must be printed out and then we must make sure that all the correct fields appear on the report as they do on the computer.

## Violation Tab Interface Specifications

<p>Interface Name: Violation Tab</p> <p>Type of Interface: <i>Form</i></p> <p>Interface Mode: <i>Screen</i></p>	<p>Interface Number: 3</p> <p>Date Created: 12/8/2015</p>
<p><b>Description:</b></p> <p>This interface can be used by all HOA office workers but it mainly used by Julie and Rhonda.</p> <p>The purpose of this interface is to record all complaints received. It enables users to easily view and edit complaint files, create violation notices, and followup on notices that have been sent out.</p> <p>This interface will be used essentially everyday to followup on notices sent out, but also whenever a complaint is received or an existing complaint file needs to be changed.</p> <p>This interface will be used in the HOA office.</p> <p>This interface is needed because it allows HOA office workers to efficiently manage all aspects of complaints and violation notices. It also allows for an easy way of viewing file information and creating the notices themselves.</p> <p>This interface is completed with inputs from the mouse and keyboard.</p> <p>The user characteristics are HOA office workers who are familiar with the system.</p> <p>The task is to efficiently manage all aspects of violation notices and complaints.</p> <p>The system is a standalone local network PC.</p> <p>The environment is employee desks at the HOA office.</p> <p>You can leave the interface by clicking the “X” in the upper right-hand corner of the screen.</p> <p>Data Flows included in this interface are:</p> <ol style="list-style-type: none"> <li>1. Complaint</li> </ol>	

2. CC&Rs
3. Penalty Warnings
4. Homeowner Accounts
5. Notice of Violation
6. Current Date File Update
7. Notice Compliance Status
8. Nest Board Meeting Date
9. Violation Penalties
10. Second Violation Notice

This interface is derived from Process 7 (Send Notice of Violation); and Process 8 (Track and Follow-up on Notices), as well as their sub processes.

Fields on this form that are filled in by the system are:

1. Complaint Information
  - a. Association Name (String)
  - b. Date Complaint Received (int)
  - c. Complaint Validity (String)
  - d. Complaint From (String)
  - e. Complaint Recipient (String)
  - f. Violated CC&Rs (String)
  - g. Complaint (String)
  - h. Complaint Recipient Information
    - i. Name (String)
    - ii. Mailing Address (String)
2. Notice Tracking
  - a. Old Violation Notices List
    - i. Notice Recipient (String)
    - ii. Number of Days Past (int)
    - iii. Notice File (String)
  - b. Selected Notice Information

- i. Notice Recipient (String)
- ii. Notice Sent Date (int)
- iii. Number of Days Past (int)
- iv. Notice File (String)
- v. Complaint in Notice (String)
- c. Follow-up Notice
  - i. Non-Satisfactory Notice File (String)

Fields on this form to be filled in by the user are:

1. Complaint
  - a. Association Name (Combo box option)
  - b. Complaint From (String)
  - c. Complaint Recipient (String)
  - d. Date Complaint Received (int)
  - e. Complaint (String)
  - f. Complaint Validity
    - i. Valid (Radio button option)
    - ii. Violated CC&Rs (String)
    - iii. Invalid (Radio button option)
    - iv. Reason(s) Complaint is Invalid (String)
2. Violation Notice
  - a. Use Above Complaint Information (Radio button option)
  - b. Search for a different reported complaint (Radio button option)
    - i. Search Complaints (String)
  - c. Penalty Warnings/ File Import (String)
3. Notice Tracking
  - a. Association Name (Combo box option)
  - b. Current Date (int)
4. Follow-up Notice
  - a. Upcoming Board of Directors Meeting Date (int)
  - b. Non-Compliance Penalties/ File Import (String)

Buttons used on this interface are:

1. Association Name: User indicates which association the reported complaint is from
2. Get CC&Rs: Retrieves CC&Rs for the selected association and holds them in the background
3. Save Complaint: Creates complaint file and saves it in the datastore
4. Reset: Clears all fields in the preceding section
5. Search: Searches the datastore for a complaint matching the complaint written in the search bar
6. Browse: Allows the user to browse their computer of the needed file
7. Import: Imports the file provided by the user into the system
8. Create and Save Violation Notice: Creates the final violation notice and saves it in the datastore
9. Create and Print Violation Notice: Creates the final violation notice and sends it to the printer to be printed out
10. Association Name (Notice Tracking): User indicates which association they would like to find old notices for
11. Find Old Notices: Searches the selected association for notices that are 15 or more days old
12. Follow-up on Selected Notice: Reads the selected notices and displays information in following sections
13. Open Copy of Notice: Gets a copy of the selected notice and displays the notice on the screen
14. Remove Notice File: Removes notices with satisfactory compliance status from the datastore
15. Create Second Notice: Sends non-satisfactory notice file to Follow-up Notice section text field
16. Create and Print Second Violation Notice: Creates the final violation notice, saves it, and sends it to the printer to be printed out

**Prototype Layout:**

Prototype can be viewed in Appendix \_\_\_.

**Test Plan:**

First the screen will go through a walkthrough evaluation to ensure that all fields are relevant to dealing with incoming complaints, creating initial notices for valid complaints, and following-up on notices that were sent out. It will then go through an interactive evaluation to make sure is understandable by users and to check the ease of use of the form.

## Violation Letter Printout Interface Specifications

<p>Interface Name: Violation Letter Printout</p> <p>Type of Interface: <i>Report</i></p> <p>Interface Mode: <i>Paper</i></p>	<p>Interface Number: <i>3.1</i></p> <p>Date Created: 12/2/2015</p>
<p><b>Description:</b></p> <p>This report will be used by homeowners who have a valid complaint filed against them. The purpose of this report is to inform violating homeowner about what CC&amp;Rs are exactly being violated and prompt them into addressing the issue.</p> <p>This report will be used whenever there is a complaint filed against a homeowner, or when the homeowner in question is not compliant within 15 days.</p> <p>This report will be used in the homes of the violating homeowners.</p> <p>This report is needed so homeowners become aware of the fact that someone in their community is not pleased with them, and it gives them a chance to fix whatever problems others may have. It can also act as a follow-up letter if the homeowner ignores the initial violation notice.</p> <p>This report will be completed by being printed out.</p> <p>The user characteristics are homeowners who are able to understand the letter.</p> <p>The task is to make homeowners aware of their violations and prompt them to address the violated within a certain time frame.</p> <p>The system is a standalone local network PC.</p> <p>The environment is the violating homeowners' houses.</p> <p>You can leave this interface by throwing away the piece of paper.</p> <p>Data flows involved in this interface:</p> <ol style="list-style-type: none"> <li>1. Penalty Warnings</li> <li>2. Homeowner Accounts</li> </ol>	

3. Notice of Violation
4. Notice Compliance Status
5. Next Board Meeting Date
6. Violation Penalties
7. Second Violation Notice

This interface is derived from Process 7.3 (Create Notice of Violation); and Process 7.4 (Save and Send Violation Notice); and Process 8.3 (Create Second Notice), as well as their sub processes.

The fields filled in by the system are:

1. Current Date (String)
2. Homeowner Name (String)
3. Homeowner Address (String)
4. Association Name (String)
5. Violated CC&Rs (String)
6. Violated CC&Rs Specifics (String)
7. Complaint the was reported (String)
8. Association Manager Name (String)

There are no buttons involved with this interface.

**Prototype Layout:**

Prototype can be viewed in Appendix \_\_\_.

**Test Plan:**

This report will be tested with a walkthrough evaluation. So, first the report must be printed out and then we must make sure that all the correct fields appear on the report as they do on the computer.

## Maintenance Tab Interface Specifications

Interface Name: Maintenance Tab	Interface Number: 4
Type of Interface: <i>Form</i>	Date Created:
Interface Mode: <i>Screen</i>	12/2/2015
<p><b>Description:</b></p> <p>This interface will only be accessed by Rhonda the receptionist.</p> <p>This interface will allow Rhonda to schedule maintenance contracts, and to send reminders to check in maintenance has been completed.</p> <p>The interface will be used whenever a new maintenance contract needs to be added, or a reminder needs to be created.</p> <p>This interface is required because these maintenance requests need to be stored digitally, and this interface allows the relevant data to be input.</p> <p>When a maintenance request is accepted, Rhonda will use the various fields to input the relevant data for that maintenance request. Once that is done, she will create a reminder so that Art or Susan can check and assure the maintenance has been completed.</p> <p>The user characteristics are defined by Rhonda, the receptionist, who is familiar with the system.</p> <p>The task is inputting information into the system.</p> <p>The system is a on a standalone local network PC</p> <p>The environment is employee desks at the HOA office.</p> <p>To leave the interface, choose another tab, or close the program by clicking the X in the top right corner.</p> <p>Data flows included in this interface are:</p> <ol style="list-style-type: none"> <li>1. Maintenance Calendar</li> <li>2. HOA Employee Schedule</li> <li>3. Maintenance Contract</li> <li>4. Maintenance Bid</li> <li>5. Maintenance Check Reminder</li> </ol>	

This interface is derived from Process 11 (Maintain Maintenance Calendar) and all of its sub processes.

Fields on this form that are filled in by the system are:

1. Maintenance Calendar
  - a. Maintenance Calendar (Interactive Calendar)
  - b. Month Contracts
    - i. Contract Name (String)
    - ii. Project Start Date (int)
    - iii. Estimated End Date (int)
    - iv. Homeowner Name (String)
    - v. Homeowner Association (String)
2. Maintenance Contract
  - a. Contract Homeowner Name (String)
  - b. Homeowner Association (String)
  - c. Project Start Date (int)
  - d. Estimated Project Duration (String)
  - e. Contracted Maintenance (String)
  - f. Special Dates (String)
  - g. Assigned Employees
    - i. Employee Name (String)
    - ii. Employee ID (int)
3. Maintenance Check Reminder
  - a. Art/Susan Schedules (Interactive Calendar)

Fields on this form to be filled in by the user are:

1. Maintenance Contract
  - a. Mark as Complete (Check box)
2. Maintenance Check Reminder
  - a. Completed Contract File (String)

- b. Check date (int)
- c. Complete Contract End Date (int)
- d. Maintenance Check time
  - i. hh (int)
  - ii. mm (int)
- e. Reminder recipient choice (Radio button options)
- 3. Maintenance Bid Request
  - a. Association Name (Combo box option)
  - b. Requesting Homeowner (String)
  - c. Requested Maintenance (String)

Buttons used on this interface are:

1. Add New Contract: Opens the Add New Maintenance Contract interface
2. View Selected Contract File: Displays selected contract file in interface
3. Edit: Enable editing of a Maintenance contract
4. Update Contract: Save edits
5. Mark contract as completed: Marks the contract as completed
6. Art Schedule Tab: Displays Art's schedule
7. Susan Schedule Tab: Displays Susan's schedule
8. hh: User chooses the hour they would like the check time to be at from a drop down box
9. mm: User chooses the minute they would like the check time to be at from a drop down box
10. Reset: Clears fields
11. Set Reminder: Submits reminder to database and to the appropriate recipient
12. Save Bid Request: Creates and saves the submitted bid request

**Prototype Layout:**

Prototype can be viewed in Appendix \_\_\_.

**Test Plan:**

First, click the edit button. All fields are filled in with relevant data. Then, click update contract. Next, click set reminder, then save bid report.

## Add New Maintenance Contract Interface Specifications

<p>Interface Name: Add New Maintenance Contract Interface</p> <p>Type of Interface: <i>Form</i></p> <p>Interface Mode: <i>Screen</i></p>	<p>Interface Number: 4.1</p> <p>Date</p> <p>Created:12/2/2015</p>
<p><b>Description:</b></p> <p>This interface will only be accessed by Rhonda the receptionist.</p> <p>This interface will allow Rhonda to schedule maintenance contracts.</p> <p>The interface will be used whenever a new maintenance contract needs to be added.</p> <p>This interface is required because these maintenance requests need to be stored digitally, and this interface allows the relevant data to be input.</p> <p>When a maintenance request is accepted, Rhonda will use the various fields to input the relevant data for that maintenance request.</p> <p>The user characteristics are defined by Rhonda, the receptionist, who is familiar with the system.</p> <p>The task is inputting information into the system.</p> <p>The system is a on a standalone local network PC</p> <p>The environment is employee desks at the HOA office.</p> <p>To leave the interface, choose another tab, or close the program by clicking the X in the top right corner.</p> <p>Data flows included in this interface are:</p> <ol style="list-style-type: none"> <li>1. Maintenance Calendar</li> <li>2. HOA Employee Schedule</li> <li>3. Maintenance Contract</li> <li>4. Maintenance Bid</li> </ol>	

This interface is derived from Process 11.2 (Get Contract Dates and Estimate Duration); Process 11.3 (Select Employee to Complete Contract); and Process 11.4 (Enter Contract Dates and Estimated Duration), as well as their sub processes.

Fields on this form that are filled in by the system are:

1. Selected Employee Information
  - a. HOA Employee Schedule (Interactive Calendar)
  - b. Employee Name (String)
  - c. Employee ID (int)
  - d. Employee Specialities (String)
  - e. Employee Image (jpeg/String)

Fields on this form to be filled in by the user are:

1. Contract Information
  - a. Contract Homeowner Name (String)
  - b. Homeowner Association(Combo box option)
  - c. Project Start Date (int)
  - d. Estimated Project Duration (Combo box option)
  - e. Contracted Maintenance (String)
  - f. Special Dates (String)
  - g. Estimated Number of Employees (Selector option)
  - h. Employee Name (Combo box option)

Buttons used on this interface are:

1. Homeowner Association: User indicates which association they are from
2. Select Duration: User indicates how long they think the contracted project will last
3. Estimated Number of Required Employees: User indicates how many employees they think the project will require

4. Select Employee Name: User indicates which employee information they would like to view from a list of HOA employees
5. Assign Employee: Adds the selected employee to the job
6. Add Another Employee: clears the fields so that another employee can be selected to be added to the job
7. Cancel: Closes the form and does not submit data
8. Reset: Clears all fields
9. Save Maintenance Contract: Save and submit contract to data store, then close the interface.

**Prototype Layout:**

Prototype can be viewed in Appendix \_\_.

**Test Plan:**

First, click the add new contract button on then maintenance tab. Then, fill out the relevant information in the form. Click assign employee to assign the first employee. Then, click add another employee, and enter the details for another employee in the appropriate boxes. Then click assign employee again. Click Save Maintenance Contract.

## Newsletter Tab Interface Specifications

<p>Interface Name: Newsletter Tab</p> <p>Type of Interface: <i>Form</i></p> <p>Interface Mode: <i>Screen</i></p>	<p>Interface Number: 5</p> <p>Date Created: 12/8/2015</p>
<p><b>Description:</b></p> <p>This interface will mainly be used by Julie.</p> <p>The purpose of this interface is to save and edit the newsletter contracts managed by HOA, along with allowing Julie to create new contract files.</p> <p>This interface will be used whenever Julie must create a community newsletter and/or an association wants to start a new newsletter contract.</p> <p>This interface will be at Julie’s desk.</p> <p>This interface is needed so all of the various newsletter contracts are easily retrievable by Julie, and it makes getting all required information needed for the newsletter easier and more efficient.</p> <p>This interface is completed by inputs from the mouse and keyboard.</p> <p>The user characteristics are an HOA office worker (Julie) who is familiar with the system.</p> <p>The task is viewing an existing newsletter contract and possibly editing it, and/or inputting new contract information into the system.</p> <p>The system is a standalone local network PC.</p> <p>The environment is employee desks at the HOA office.</p> <p>You can leave the interface by clicking the “X” in the upper right-hand corner of the screen.</p> <p>Data flows included in this interface are:</p> <ol style="list-style-type: none"> <li>1. Newsletter Contract</li> <li>2. Community Newsletter</li> </ol>	

This interface is derived from Process 13 (Create Community Newsletter), as well as its sub processes.

Fields on this form that are filled in by the system are:

1. Contract Information
  - a. Association Name (String)
  - b. Community Name (String)
  - c. Newsletter Cycle Duration (String)
  - d. Newsletter Sections (String)
  - e. Layout Specifications (String)
  - f. Design Specification (String)
  - g. Additional Comments (String)
2. Newsletter Information
  - a. Imported Files (List of Strings)

Fields on this form to be filled in by the user are:

1. Newsletter Contract
  - a. Association Name (Combo box option)
  - b. Community Name (Combo box option)
  - c. Newsletter Sections
    - i. Community News (Check box)
    - ii. Weekly Weather (Check box)
    - iii. Community Activities (Check box)
    - iv. Community Events (Check box)
    - v. Upcoming Dates (Check box)
    - vi. Birthdays (Check box)
    - vii. Births (Check box)
    - viii. Obituaries (Check box)
    - ix. Other (Check box)
  1. Other Section Header(s) (String)

- d. Layout Specifications (String)
- e. Newsletter Cycle Duration (Combo box option)
- f. Design Specifications (String)
- g. Additional Comments (String)
- 2. Community Newsletter
  - a. Newsletter Contract (Combo box option)
- 3. Newsletter Information
  - a. Section Information File (String)
  - b. Image Upload (jpeg/String)
- 4. Newsletter Upload
  - a. Finished Newsletter Upload (String)

Buttons used on this interface are:

1. Association Name: User indicates which association this new contract is for
2. Community (if specified): User indicates which association community the newsletter is being made for
3. Select Newsletter Cycle Duration: User indicates how often they want newsletters to be printed
4. Save Contract and Specifications: Creates and saves the inputted newsletter contract information.
5. Reset: Clears all fields above the button
6. Select Newsletter Contract: User indicated which newsletter contract they would like to view and edit
7. Edit: Enables editing of the greyed fields
8. Save: Saves any changes made while in edit mode
9. Browse: Allows the user to browse their computer for the needed file
10. Import: Imports the file provided by the user into the system
11. Save Newsletter Information: Saves the imported files to the newsletter contract file
12. Clear: Clears the imported files showing
13. Save Newsletter: Saves the finished uploaded newsletter to the datastore

14. Print Newsletter: Sends the finished uploaded newsletter to the printer to be printed

**Prototype Layout:**

Prototype can be viewed in Appendix \_\_.

**Test Plan:**

First the screen will go through a walkthrough evaluation to ensure that all fields are relevant to creating a community newsletter. It will then go through an interactive evaluation to make sure is understandable by users and to check the ease of use of the form.

## Newsletter Printout Interface Specifications

<p>Interface Name: Newsletter Printout</p> <p>Type of Interface: <i>Report</i></p> <p>Interface Mode: <i>Paper</i></p>	<p>Interface Number: <i>5.1</i></p> <p>Date Created: 12/8/2015</p>
<p><b>Description:</b></p> <p>This report will be used by the homeowners of a particular community in an association. The purpose of this report is to keep community homeowners updated about whatever the association contracts the newsletter to be about.</p> <p>This report will be used for the newsletter cycle duration that was chosen by the association. (There are various cycle lengths so it depends)</p> <p>This report will be used in the homes of all homeowners in the given community.</p> <p>This report is needed so homeowners are able to keep up with community happenings, and so that they can stay involved in the community.</p> <p>This report will be completed by being printed out.</p> <p>The user characteristics are homeowners who are able to understand how to read.</p> <p>The task is to inform homeowners of current information the association would like to share with the members of the particular community.</p> <p>The system is a standalone local network PC.</p> <p>The environment is all community homeowner houses.</p> <p>You can leave this interface by throwing away the piece of paper.</p> <p>Data flows involved in this interface:</p> <ol style="list-style-type: none"> <li>1. Newsletter Contract</li> <li>2. Community Newsletter</li> </ol> <p>This interface is derived from Process 13.3 (Create and Print Newsletter).</p>	

The fields filled in by the system are:

1. Association Name (String)
2. Community Name (String)
3. Newsletter Title (String)
4. Sections Headers (String)
  - a. Section Information Imports (String)
  - b. Image Imports (jpeg)

There are no buttons involved with this interface.

**Prototype Layout:**

Prototype can be viewed in Appendix \_\_\_.

**Test Plan:**

This report will be tested with a walkthrough evaluation. So, first the report must be printed out and then we must make sure that printed newsletter matches or is very similar to the specifications provided by the association.

## **Appendix B**

### **Interface Test Plans**

## **Interface Test Plans**

The following test plans correspond to each of the interfaces displayed in the interface structure diagram. The test plans ensure that each interface works how it is supposed to including reading the input fields, the buttons, etc. Each test plan consist of multiple test cases and are only partially completed due to not having run the actual tests yet. The test cases within each test plan test various problems that might occur in that particular interface using fake testing inputs.

## Associations List Tab Test Plan 1

<b>Test Plan</b>		Page 1 of 3
<b>Program ID: Associations List Tab    Version Number: 1</b>		
<b>Tester: Matthew McJury    Date Designed : 12/2/15    Date Conducted : 12/2/15</b>		
<b>Results:    X Passed</b>		<b>Open Items</b>
<b>Test ID: 1    Requirement Addressed: Managing associations and committees</b>		
<b>Objective:</b> To test the select association section of the Associations List Tab Interface		
<b>Test Cases</b>		
Interface ID	Data Field	Value Entered
1) _____1_____	_____Selected Association _____	_____Association 2_____
<b>Script</b> Select Association 2 from the drop down menu. The other field should auto populate. Click edit. The greyed out fields become white. In the name field, change the name of Association 2 by appending it with an "A". Click Print association list, and the changes should save, and a printed copy should be created.		
<b>Expected Results/Notes</b> A printout with the relevant information for that association is created.		
<b>Actual Results/Notes</b>		

## Associations List Tab Test Plan 2

<b>Test Plan</b>		Page 2 of 3
<b>Program ID: Associations List Tab    Version Number: 2</b>		
<b>Tester: Matthew McJury    Date Designed : 12/2/15    Date Conducted : 12/2/15</b>		
<b>Results:    X Passed</b>		<b>Open Items</b>
<b>Test ID: 1    Requirement Addressed: Managing associations and committees</b>		
<b>Objective:</b> To test the association committee section of the Associations List Tab interface.		
<b>Test Cases</b>		
Interface ID	Data Field	Value Entered
1) _____1_____	_____Selected committee_____	_____Committee 2_____
2) _____2_____	_____Committee Member_____	_____Member 3_____
<b>Script</b>		
Select Association 2 from the drop down menu. The other field should auto populate. Select Committee 2 under the committee drop down. Select member 3 under the member drop down menu. Click the red “Remove Member” Button. Click Save committee List. Click Print Committee List		
<b>Expected Results/Notes</b>		
A printout of Committee 2’s members is printed, and it is missing member 3.		
<b>Actual Results/Notes</b>		



**Ordered Association List Printout Test Plan**

<b>Test Plan</b>		Page
___ of ___		
<b>Program ID: Print Ordered Association List Version Number:</b> _____		
<b>Tester: Tyler Dobbs    Date Designed : 12-3-15    Date Conducted :</b> _____		
-		
<b>Results:    Passed</b>		<b>Open Items</b>
<b>Test ID:</b> _____	<b>Requirement Addressed:</b> _____	
<b>Objective:</b> The print a report showing associations in order		
<b>Test Cases</b> Inputs: “Bob’s Apartments, Franks Motel, Crystal Rock Hotel” Outputs: Print off of Ordered Association List		
<b>Script</b> -Click the button Receive the report		
<b>Expected Results/Notes</b> A report showing an ordered list of associations		
<b>Actual Results/Notes</b>		



**Payment Tab Test Plan**

<b>Test Plan</b>		Page <u>1</u>
of <u>1</u>		
<b>Program ID: Monthly Bill Version Number: 2_</b>		
<b>Tester: Tyler Dobbs Date Designed : 12/2/15 Date Conducted : 12/2/15</b>		
<b>Results: X Passed Open Items</b>		
<b>Test ID: 1 Requirement Addressed: Monthly Bills</b>		
<b>Objective: Create and print Monthly Bill</b>		
<b>Test Cases</b>		
Interface ID	Data Field	Value Entered
1) _____ Complex	_____Association Name_	Franks's Apartment
2) _____	_____Select Month	July
<b>Script</b>		
Select an Association Name		
Select a month		
Click on Create and Print Bill		
<b>Expected Results/Notes</b>		
A printed document of all bills for the selected month		
<b>Actual Results/Notes</b>		

**Delinquency Tab Test Plan**

<b>Test Plan</b>		Page _1
of _1_		
<b>Program ID: Delinquency Tab    Version Number: 2_</b>		
<b>Tester: Tyler Dobbs    Date Designed : 12/2/15    Date Conducted : 12/2/15</b>		
<b>Results:    X Passed</b>		<b>Open Items</b>
<b>Test ID: 1    Requirement Addressed: Finding Delinquent Homeowners</b>		
<b>Objective:</b>		
<b>Test Cases</b>		
Interface ID	Data Field	Value Entered
1) _____	_____ Association Name _____	Joe's
Apartment_____		
2) _____	_____ Current Date	12-5-2012
_____		
<b>Script</b>		
Select Association Name from drop down box.		
Input the current date		
Click on Find Delinquent Records		
Select a month to filter by.		
<b>Expected Results/Notes</b>		
<b>Actual Results/Notes</b>		

**Delinquency Letter Printout Test Plan**

<b>Test Plan</b>		Page _1
of _1_		
<b>Program ID: Delinquency Tab    Version Number: 2_</b>		
<b>Tester: Tyler Dobbs    Date Designed : 12/2/15    Date Conducted : 12/2/15</b>		
<b>Results:    X Passed</b>		<b>Open Items</b>
<b>Test ID: 1    Requirement Addressed: Generating Delinquent Letters</b>		
<b>Objective:</b>		
<b>Test Cases</b>		
Interface ID	Data Field	Value Entered
1) _____	_____ Association Name _____	Joe's
_____ Apartments _____		
2) _____	_____ Current Date	July _____
<b>Script</b>		
Select Association Name from drop down box.		
Input the month to search for		
Click on Find Delinquent Records List File		
Click on Consolidate Duplicate Records		
Click on Print Delinquent Letters		
<b>Expected Results/Notes</b>		
<b>Actual Results/Notes</b>		

**Violation Tab Test Plan 1**

<b>Test Plan</b>		Page <u>1</u>
of <u>1</u>		
<b>Program ID: Complaint Version Number: 1</b>		
<b>Tester: Tyler Dobbs Date Designed : 12/2/15 Date Conducted : 12/2/15</b>		
<b>Results: X Passed Open Items</b>		
<b>Test ID: 1 Requirement Addressed: Submit a complaint</b>		
<b>Objective:</b> To test the complaint section to make sure it works corretly.		
<b>Test Cases</b>		
Interface ID	Data Field	Value Entered
1) _____	_____Association Name_____	Joe's Apartments
2) _____	_____Complaint From_____	Peter Guddles
3) _____	_____Complaint Recipient	Joe
Guddles_____		
4) _____	_____Date Complaint Recieved_	12/24/2012_____
5) _____	_____Valid_____	Checked_____
6) _____	_____Violated CC&R's_____	Too
Loud_____		
<b>Script</b>		
Enter Association name Click on Get CC&R's Enter complaint information Click on Save complaint		
<b>Expected Results/Notes</b>		
That the complaint is recorded.		
<b>Actual Results/Notes</b>		

**Violation Tab Test Plan 2**

<b>Test Plan</b>		Page <u>1</u>
of <u>1</u>		
<b>Program ID: Notice Tracking    Version Number: 2_</b>		
<b>Tester: Tyler Dobbs    Date Designed : 12/2/15    Date Conducted : 12/2/15</b>		
<b>Results:    X Passed</b>		<b>Open Items</b>
<b>Test ID: 1    Requirement Addressed: Find past complaints and send a notice</b>		
<b>Objective:</b>		
<b>Test Cases</b>		
Interface ID	Data Field	Value Entered
1) _____	Association Name	Joes Apartments
2) _____	Current Date	12-4-15 _____
<b>Script</b>		
Selected Association Name from drop down box		
Select the current date		
Click on Find Old Notices		
Click on Relevant notice		
Click on Follow-up on Selected notice		
Click on Create Second Notice if necessary, if not, Remove Notice File		
Select the Meeting date for which homeowner shall attend		
Click on Create and Print Second Violation Notice		
<b>Expected Results/Notes</b>		
<b>Actual Results/Notes</b>		

**Violation Letter Printout Test Plan**

<b>Test Plan</b>		Page <u>  </u> of <u>  </u>
<b>Program ID: Violation notice    Version Number: 2_</b>		
<b>Tester: Tyler Dobbs    Date Designed : 12/2/15    Date Conducted : 12/2/15</b>		
<b>Results:    X Passed</b>		<b>Open Items</b>
<b>Test ID: 1      Requirement Addressed: Send a notice about a violation</b>		
<b>Objective:</b>		
<b>Test Cases</b>		
Interface ID	Data Field	Value Entered
1) _____	Search for a different reported complaint	Checked _____
<b>Script</b>		
Click on Search for a different reported complaint		
Type in search criteria		
Click on Search		
Verify information that appeared		
Click on Create and Print Violation Notice		
Click on Create and Save Violation Notice		
<b>Expected Results/Notes</b>		
<b>Actual Results/Notes</b>		



**Add New Maintenance Contract Test Plan**

**Newsletter Tab Test Plan 1**

<b>Test Plan</b>		Page _1
of _1_		
<b>Program ID: Newsletter Contract    Version Number: 2_</b>		
<b>Tester: Tyler Dobbs    Date Designed : 12/2/15    Date Conducted : 12/2/15</b>		
<b>Results:    X Passed</b>		<b>Open Items</b>
<b>Test ID: 1    Requirement Addressed: Create a newsletter Contract Objective:</b>		
<b>Test Cases</b>		
Interface ID	Data Field	Value Entered
1) _____	___Association Name_____	Joes Apartments
2) _____ News_____	_____Newsletter Section	Community
3) _____	Select Newsletter Cycle duration__	Weekly_____
<b>Script</b>		
Select an Association Name Select multiple(if applicable) details and requirements Select Cycle Duration Click Save Contract and Specification		
<b>Expected Results/Notes</b>		
<b>Actual Results/Notes</b>		

**Newsletter Tab Test Plan 2**

<b>Test Plan</b>		Page <u>1</u>
of <u>1</u>		
<b>Program ID: Community newsletter    Version Number: 2_</b>		
<b>Tester: Tyler Dobbs    Date Designed : 12/2/15    Date Conducted : 12/2/15</b>		
<b>Results:    X Passed</b>	<b>Open Items</b>	
<b>Test ID: 1</b>	<b>Requirement Addressed: Edit a newsletter Contract</b>	
<b>Objective:</b>		
<b>Test Cases</b>		
Interface ID	Data Field	Value Entered
1) _____	___Select Newsletter Contract_____	Joe's Apartments
<b>Script</b>		
Select a Newsletter Contract		
Verify Information		
Click on edit if necessary		
Click on Save		
<b>Expected Results/Notes</b>		
<b>Actual Results/Notes</b>		

**Newsletter Printout Test Plan**

<b>Test Plan</b>		Page _1
of _1_		
<b>Program ID: Newsletter Upload Version Number: 2_</b>		
<b>Tester: Tyler Dobbs Date Designed : 12/2/15 Date Conducted : 12/2/15</b>		
<b>Results: X Passed</b>		<b>Open Items</b>
<b>Test ID: 1</b>	<b>Requirement Addressed: Upload a new Newsletter</b>	
<b>Objective:</b>		
<b>Test Cases</b>		
Interface ID	Data Field	Value Entered
1) _____	___Finished Newsletter Upload_____	newsletter.pdf
<b>Script</b>		
Click on browse		
Select Newsletter File		
Click on Import		
Click on Print Newsletter		
Click on Save Newsletter		
<b>Expected Results/Notes</b>		
<b>Actual Results/Notes</b>		