CS4400 Database Project
Fall Semester 2014
(Version 1.2)

Look at the last page for edits made for each version

Purpose of the Project

Analyze, specify, design, implement, document and demonstrate an information system application to support a new apartment rental system. You are required to use the Classical Methodology for Database Development. The system should be implemented using a relational DBMS that supports standard SQL queries. Class administrators will provide you with information about how to access a college-managed MySQL server in order to implement your database and the application. The professors must approve any other alternative implementations. *In no circumstances can you use a tool that automatically generates SQL or automatically maps programming objects into the database.*

Project Phases

The three phases of the project cover the following work-processes from the Classical Methodology for Database Development (see notes on T-square under resources).

Phase		Due Date
I	Analysis & Specification	Oct 1
II	Logical Database Design	Nov 3
III	Implementation & Testing	Dec 2
	Demonstrations	Dec 2

Groups

Project groups may have 3-4 members. Please avoid having smaller groups. 5 or more members will NOT be acceptable.

A group may remove a member from further participation in the group when Phase I is turned in or when Phase II is turned in. A written notification must be provided to the professor at that time.

Deliverables

Slides on database design methodology will be useful for phases I and II: These slides have been posted on t-square.

Phase I (hard copy)

The deliverables include:

1. A cover page listing all members in the team with their respective sections and email addresses and T-square username.
2. Enhanced Entity Relationship (EER) Diagram
3. Information Flow Diagram
4. A list of logical constraints that will be enforced. Do not include any constraints that can be shown in the ER diagram, but rather semantic, business logic related constraints. **You are required to include at least five constraints, although a fully-specified system will probably have more than that. Constraints that can be specified directly using ER notation will not count toward the five required.**
5. Any assumptions made including explanations.

Notes:

1. The EER must capture the constraints of the system as much as possible whenever applicable, i.e. total participation, super/sub class, weak entities.
2. The design of your system must satisfy all the constraints. You are allowed to make up additional assumptions and constraints as long as they do not conflict with the specified constraints and requirements. If possible, those additional assumptions and constraints should be included in the ER diagram.
   You must turn in a hard copy of your report in class

Phase II (hard copy)

1. Cover Page
2. Copy of the ER Diagram (either from phase I (with any revisions) or from the solution provided)
3. Copy of the Information Flow Diagram from phase I (either from phase I (with any revisions) or from the solution provided)
4. Relational Schema Diagram (with primary and foreign keys identified, referential integrity is shown by arrows)
5. Create Table statements, including domain constraints, integrity constraints, primary keys, and foreign keys
6. SQL statements for each task (***follow the template in the phase II design methodology***)

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

Phase III

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

Implement a working application with all functionality described in this document. Your source code should be mailed to the respective TA who grades your project by the deadline.

Deliverables for Phase 3 are:
1. Bring your laptop for the demo.

Heavyweight option (NO SUBMISSION REQUIRED):
2. Make sure you have a text file (soft copy) with all your SQL queries only (This is just in case your implementation doesn't work)
3. Working functional application with embedded SQL statements that accesses your database (This is your actual application)

Lightweight option(NO SUBMISSION REQUIRED):
2. Make sure you have a text file (soft copy) with all your SQL queries only.

Grading

The project will consist of three phases (deliverables) as well as a final demonstration to the TA. Phase I and Phase II of the project are each worth 10% credit.

Phase III (20% or 5% credit, depending on option):
Heavy Weight Option (20 %): The students would be required to use the embedded SQL feature of MySQL which allows you to embed SQL statements in a Java program or web application. (You can use whatever programming language you are comfortable with)
Light Weight option (5 %): The students would be required to demo the SQL queries on the MySQL console. They would be required to take the Final exam.
Final Exam (15 %) : This would be only taken by students who have opted for the lightweight phase III. Under no circumstances would a heavy weight option student be allowed to take the Final.

Atlanta Apartment Homes Portal (AAHP)

AAHP is a simple online apartment rental system, which leases apartments of a single apartment community called Atlanta Apartment Homes to prospective residents. There are three types of users: **prospective residents**, **residents** and **management (members of the management cannot be residents)**.

The following sections contain a functional description of the AAH application along with some screen mockups. The user interfaces depicted in this project description merely serve as examples to guide your thinking. Your project's interface may look different and that is fine—even encouraged! For example, you might choose to split up some interfaces we have shown on a single screen into multiple screens. You might choose to use popup windows instead of refreshing the page. A complete reorganization of the user interface is acceptable as long as your application supports the same functionality as described below. You may implement the project as a traditional standalone application (e.g., using Java GUIs) or as a web application (e.g., using a web scripting language like PHP). **There is no restriction on the choice of language.**

1. Logging in

Fig 1. shows the AAHP login screen. Residents / prospective residents and the management are uniquely identified by their **Username.** A valid **Username** and **Password** combination is required to log in to the system and proceed. If the user provides invalid log in credentials, an error message should be displayed and the user should be redirected to the log in screen. A prospective resident needs to create an account before he / she can fill in the prospective resident form.

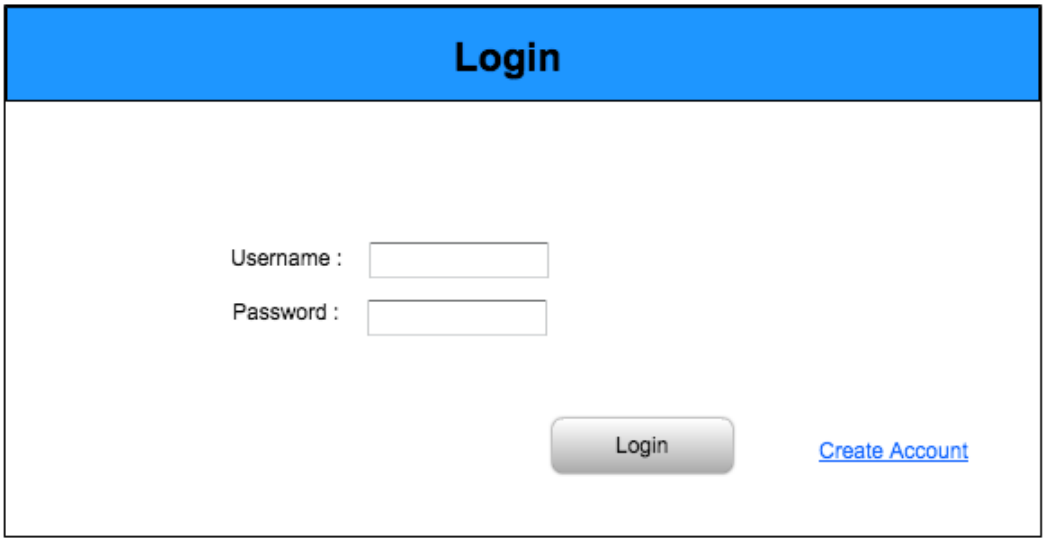


Figure 1. Login

2. New User Registration

A new user needs to register before using the AAH. This applies only to prospective residents. The management personnel already have their credentials created behind the scenes. Clicking the 'Create Account' link on the log in screen displays the new user registration form as shown in Fig. 2.

Figure 2. Create Account

After the user clicks **Register**, the system should verify that all fields are filled in, that the **Username** has not already been registered, and that the **Password** and **Confirm Password** fields are equal. If any of these validations fail , the user should be returned to this screen to make corrections. The user should be provided with meaningful error messages so he (from here onwards we would use 'he' to represent the user without any intended bias) knows what to correct. The new user should be navigated to the prospective resident application form shown in Fig. 3

3. Prospective Resident Applicant Form

The system should not allow a resident to fill out prospective resident application form more than once. Other than the basic information such as name, date of birth and gender they would be required to fill in their monthly income and the category of the apartment they are looking for. Please note that the combination of name and date of birth is unique. The apartments may be of the following categories: 1bdr-1bth, 2 bdr- 1bth and 2 bdr-2bth. Each category may have multiple units of different area(sq. ft). E.g there may be five 2bdr-1 bath

apartments each having a different area. The user would also be required to fill in a range of monthly rent based on his budget. The prospective residents would also need to fill in a move-in date , lease term (3months / 6months / 12 months) and their previous address. Also note that the move-in date cannot be beyond 2 months from the current date (date of the application). The system should prompt the user in that case. After applying the form is processed by the management. The application would be rejected if monthly income is less than 3 times the rent or if the preferred category of the apartment is not available on the preferred move in date. If the above criteria is met, the application is processed behind the scenes and the prospective resident is sent a welcome letter. (Due to the limited scope of the application we assume that once a prospective resident is alloted an apartment by the management, he takes it). Only the prospective resident who fills in the application form and is allotted an apartment by the management can use this system. If a prospective resident tries to log in before being approved by the management, the system should prompt the user with an error message 'Your application is under review'. We assume that his family members would not create an account for themselves. If the prospective resident doesn't submit the form or closes the window by mistake, then his username and password should be deleted from the database. In such a case he would need to create a new account. **(All fields are mandatory).**

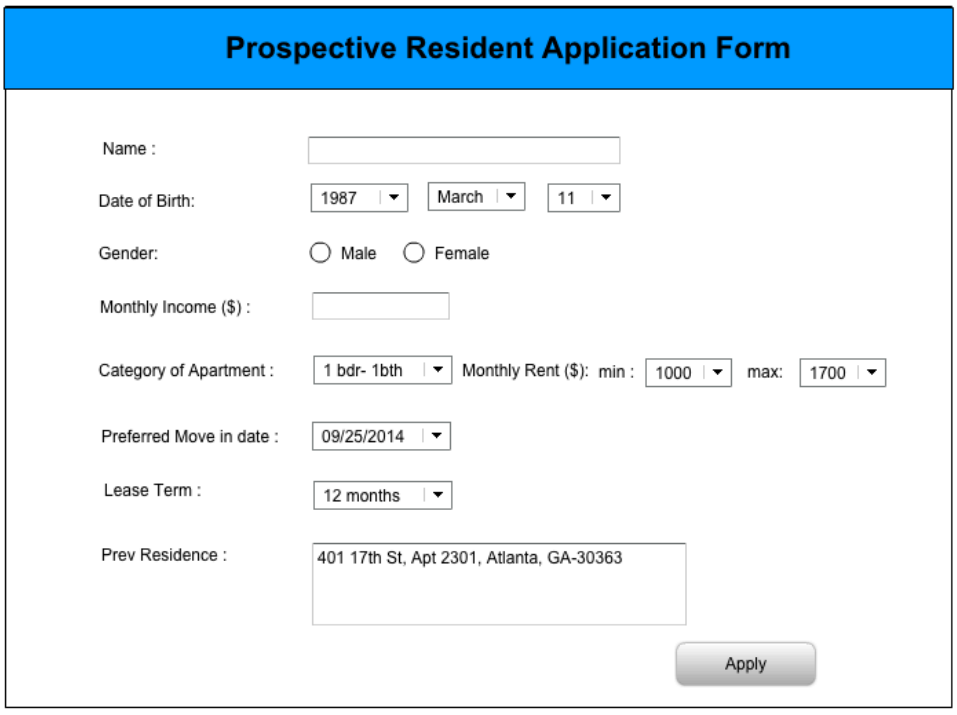


Figure 3. Prospective Resident Application Form

After logging in the user should be directed towards a homepage as shown in Fig. 4

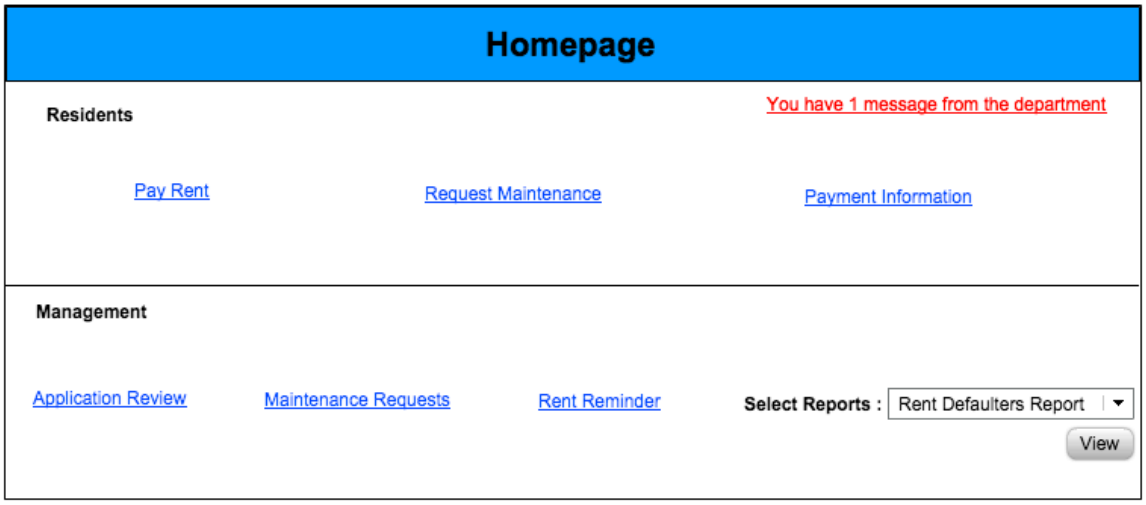


Figure 4. Homepage

All the users are navigated to the homepage after logging in. Only those functionalities which apply to a user should be enabled for him. A resident can do the following :-

1. Pay Rent
2. Request Maintenance
3. Add / Delete payment information

The management can do the following :-

1. Review Applications from prospective residents
2. View Maintenance Requests
3. Rent Reminders
4. Select a report to view.

Resident Functionality

1. Rent Payment

The residents can use this system to pay rent. Rent for a month is due on the 3rd day of the month. There is a 50$ fine per day after the 3rd. If a resident has moved in after the 7th day of the month, the rent is pro-rated (rent per day = Monthly Rent / No. of days in the month). The system would keep a record of the rent paid for every resident. The screen shown in Fig. 5 would be used to perform this operation. (Autopopulate the rent amount and the apartment number). **(All fields are mandatory)**

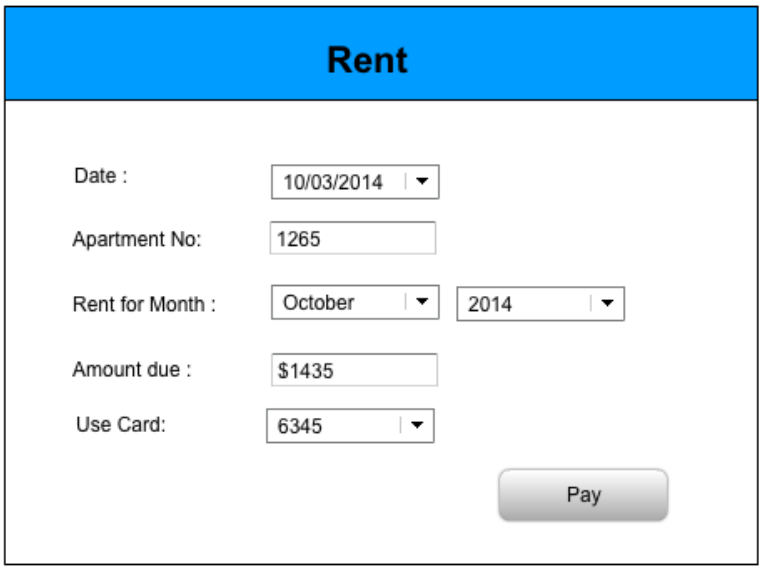


Figure 5. Rent

2 .Request Maintenance

The residents need to put in a service request for any kind of maintenance issues with respect to their apartment. He needs to choose from a list of issues from a drop down list. The apartment number should get autopopulated. The screen shown in Fig. 6 would be used for the same.

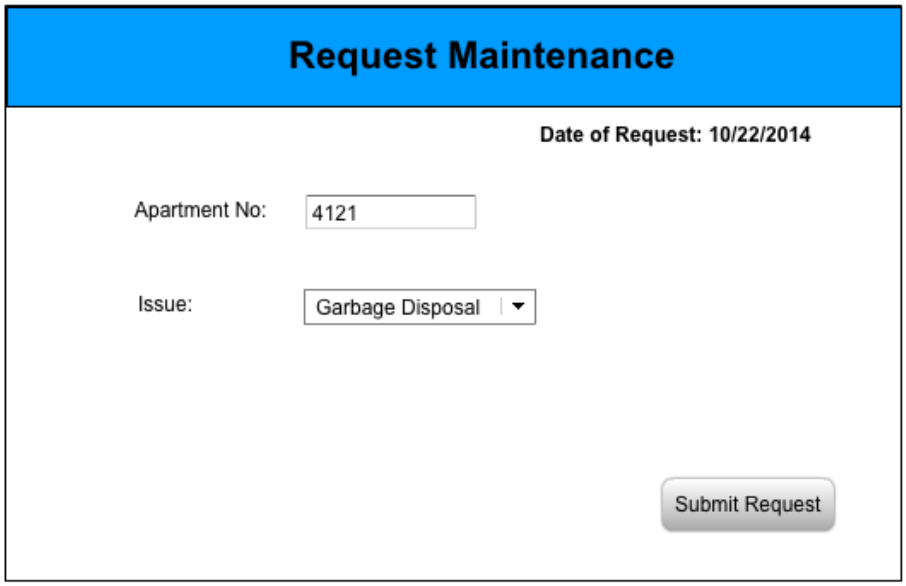


Figure 6. Request Maintenance

2. Payment Information

A resident can use the payment information link to add information about a new credit card or delete information about previous cards. A resident can add multiple cards in the system. Please note that he cannot pay rent using this system unless he has added information about at least one card. The left side of the screen shown in Fig 7 is used to enter information about a new card. The right side of the screen is used to select one of the cards to be deleted. You can either show the last 4 digits of the card to be deleted or use the complete card number.

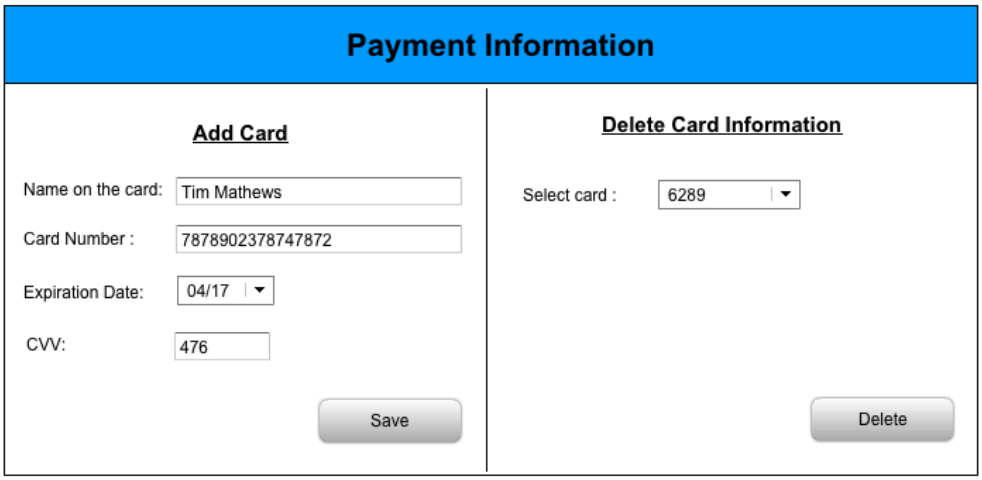


Figure 7. Payment Information

Management Functionality

3. Assigning apartments to prospective residents

The management can view the applications submitted by the prospective residents. The screen shown in Fig 8 shows the applicant information along with information on which ones got accepted / rejected. The official needs to select an accepted applicant and click 'Next' to go to the next screen to allot an apartment.

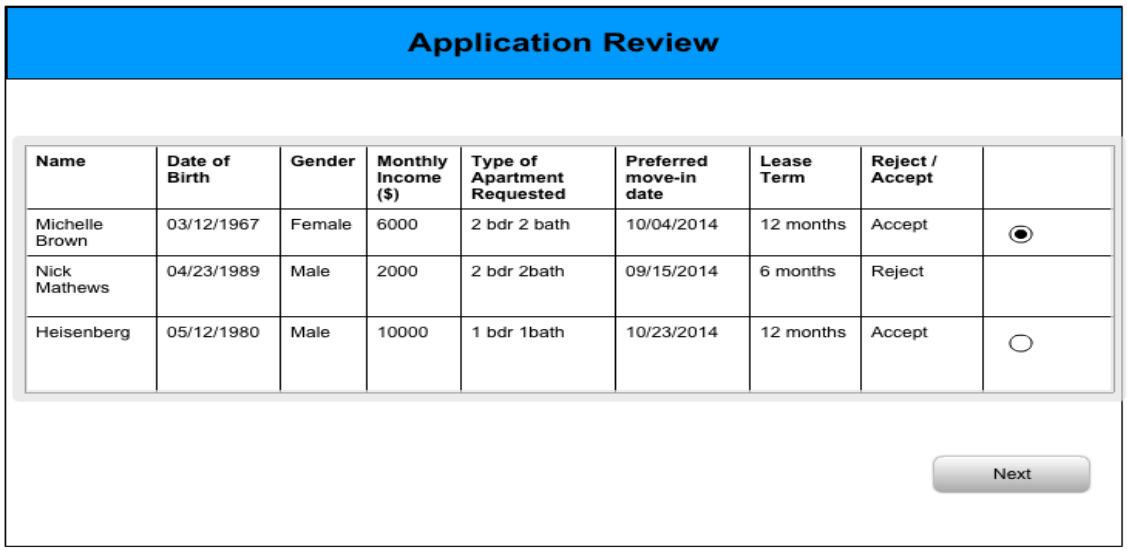


Figure 8. Application Review

Fig. 9 shows the apartment allotment screen with the applicant name on top. It also shows the apartment sorted by rent (descending) matching his criteria along with other information about the apartment such as dates of availability. We assume that if a resident moves out, then his records are deleted from the system.

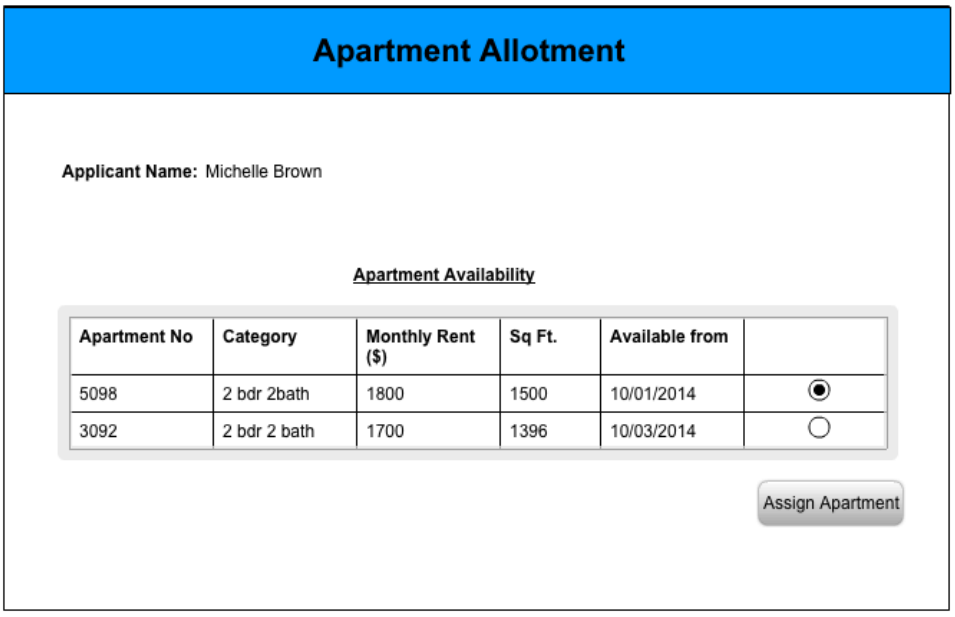


Figure 9. Apartment Allotment

4. View maintenance requests

The management can also view the service requests made by residents for issues in their apartment. The screen in Fig. 10 shows the requests made sorted by the date of request from oldest to the most recent. (See Fig. 8). On checking the appropriate checkboxes and clicking on 'Resolved', the request should be moved down to the resolved section.

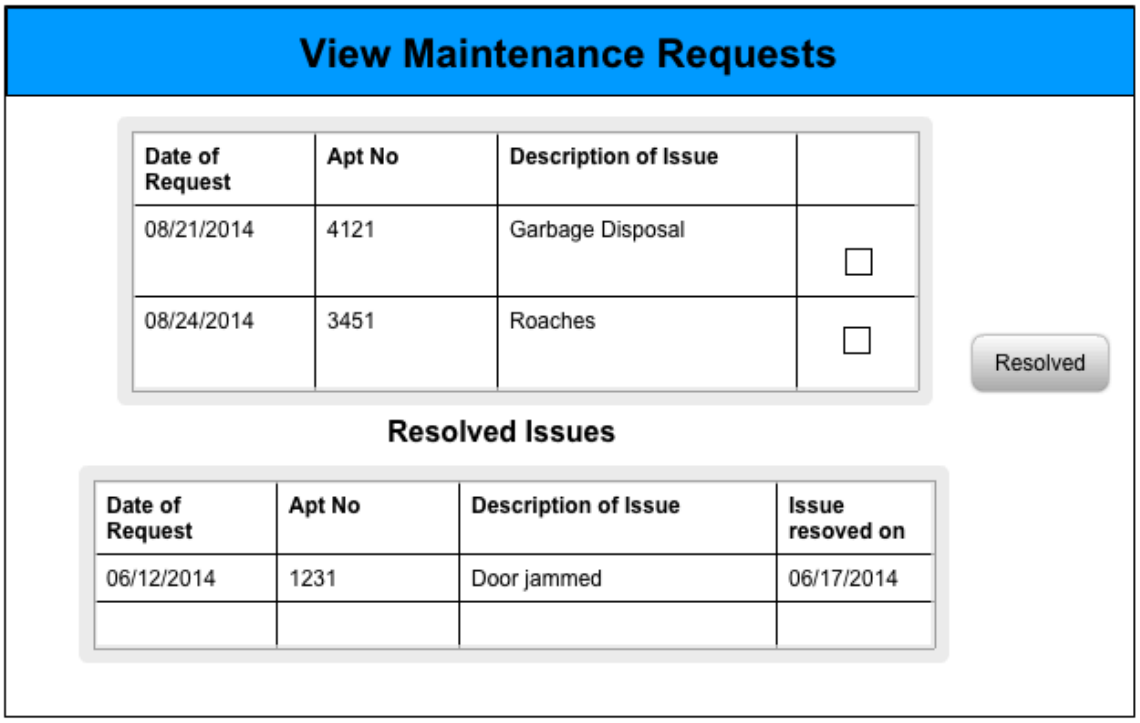


Figure 10. View Maintenance Requests

5. Rent Reminders

The management needs to send a reminder to a resident who has not paid rent by the 3rd of the month. This would be done using a messaging functionality. The screen shown in Fig 11. Would be used to send the messages. Only those apartment numbers should appear in the dropdown menu which have not paid the rent. When a message is sent the resident of the corresponding apartment would get a notification on the homepage (see Fig 4.) when he logs in. On clicking the notification he can view the message sent by the management in a pop up box. (The message should not show up once opened). If a message is unopened it should be a considered as unread and the count should be displayed accordingly.

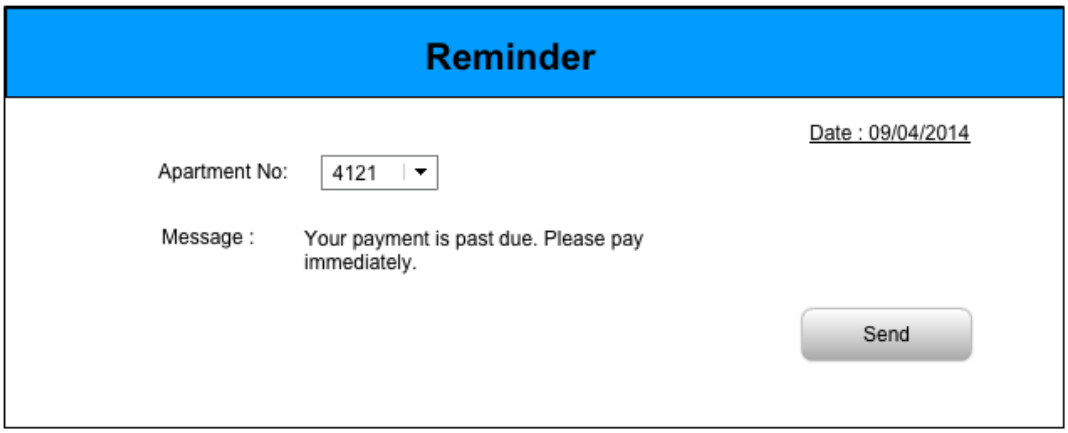


Figure 11. Reminder

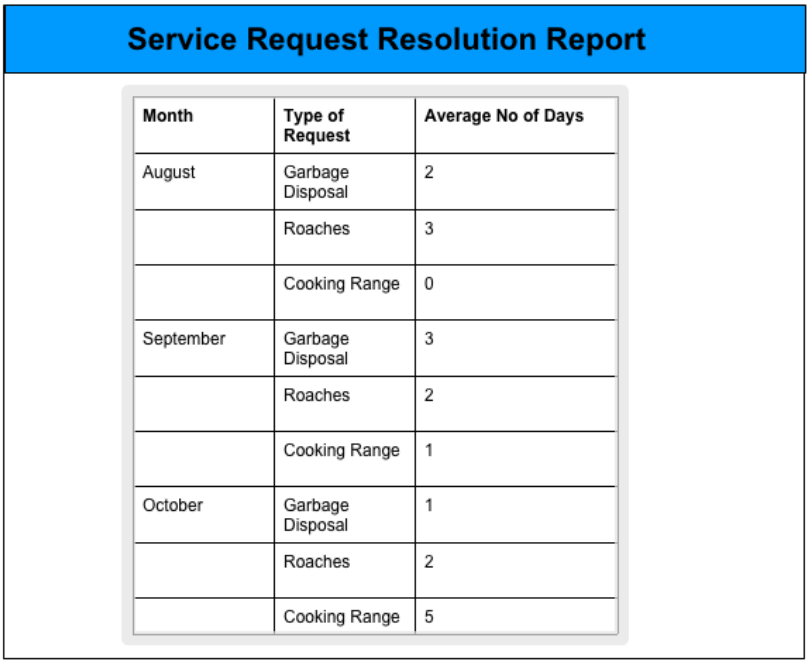
6. REPORTS

1. **Leasing Report:** This report shows the number of apartments leased for each category grouped by month. You need to show the report for the months of August, September, October.

Month	Category	No. of Apartments
August	1 bdr- 1bth	10
	2bdr - 1bth	25
	2bdr - 2bth	23
September	1 bdr- 1bth	23
	2bdr - 1bth	12
	2bdr - 2bth	14
October	1 bdr- 1bth	20
	2bdr - 1bth	12
	2bdr - 2bth	32

Figure 12. Leasing Report

2. **Service Request Resolution Report:** This report shows the average number of days taken to resolve a request grouped by month. This is also a 3 month report for August, September and October. If time taken to resolve an issue is less than a day, consider it as 1 day.

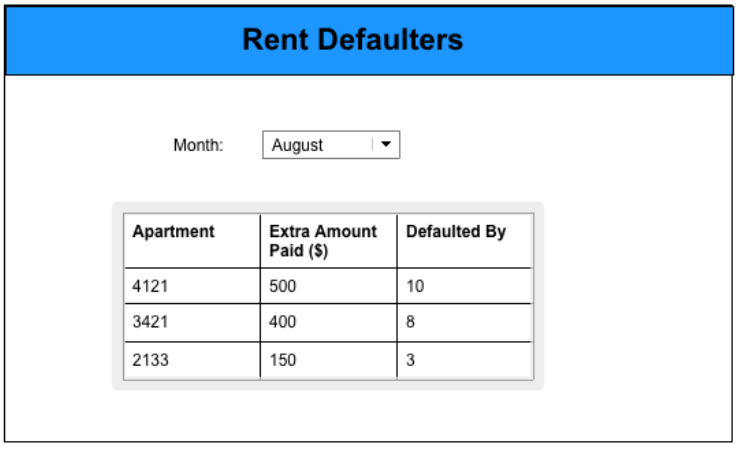


Service Request Resolution Report

Month	Type of Request	Average No of Days
August	Garbage Disposal	2
	Roaches	3
	Cooking Range	0
September	Garbage Disposal	3
	Roaches	2
	Cooking Range	1
October	Garbage Disposal	1
	Roaches	2
	Cooking Range	5

Figure 13. Service Request Report

3. **Rent Defaulter Report:** This report shows the apartments which have defaulted on the rent for a particular month along with the extra amount they paid over and above the rent and the number of days they were late by. The report should be available for any month.



Rent Defaulters

Month: August

Apartment	Extra Amount Paid ($)	Defaulted By
4121	500	10
3421	400	8
2133	150	3

Figure 14. Rent Defaulters

Version information:

Version Number	Comments
1.0	Original Draft
1.1	1. A prospective resident needs to first create an account and register and then fill out the application form. If he tries to log in without submitting the application form, he should be prompted to do so. 2. View Reports button added to the homepage. 3. If a rent reminder is not opened, then the reminder should be considered unread and the count should be displayed accordingly.
1.2	1. Prospective resident link removed from the log in screen Following is the process flow for the system: A) A prospective resident needs to create an account

	before logging in. B) Once the account is created, the prospective resident is navigated to the prospective resident application form. C) If he doesn't submit the application or closes the window by mistake, the account that he created gets deleted from the system. D) Once he submits his application, if he tries to log in before the management allots him an apartment, he should be prompted with an error message 'Your application is under review'.