Project Proposal:

Cowboy Property Management Website

Group 10

Renuka Regulagadda, Cassidy Hoff, Saurabh Dhaigude

Table of Contents

[**Introduction**](#_mjmao6fur3ue) **2**

[**Use Case Requirements and Roles**](#_3fkihbbgtvv8) **3**

[**Sitemap and Wireframe Pages**](#_dpqd66ftg5u4) **7**

[Site Map](#_sdh96tr727w1) 7

[Wire frames](#_y59q0dmble5d) 8

[**Database ERD**](#_qoxy3cjm3ucm) **13**

[**Challenges and Tentative Delivery Schedule**](#_nanronyl9o8) **14**

# 

# 

# Introduction

We will be creating a dynamic website for Cowboy Property Management. The main purpose of this project is to create a dynamic portal for users and employees for an effective rental property experience. The dynamic portal will feature options specific to employees such as the ability to view renter information and add new renters to the system. Similarly, the portal will allow customers to view their lease, rent information, submit payments, maintenance requests, and complaints. These features will help Cowboy Property Management reach their goal of offering superior customer service.

This project interested us much because we were in a similar situation when we moved to a new city and were searching for a rental property. This experience seemed like a more realistic as well as important as this is one of the real-life experiences that we face in our life. That’s why it made more sense to choose and continue with this topic.

The objective of this project is that the website will manage the resident’s information at the Cowboy Property. We are storing the resident’s information in the site’s database and arranging the availability of apartments for rent. This will provide end users better options and clarity regarding the properties that they are looking for. Additionally, the website will offer features for Cowboy Property Management employees as well, which enables a seamless customer experience. Working on this project is a great opportunity for us to solve real life business problems and will help us hone our skills used for this.

To log in to test our website and test different roles, please use:

|  |  |  |
| --- | --- | --- |
| **Username** | **Password** | **Role** |
| bsmith | mysecret | CTO |
| pjones | acrobat | Employee |
| bcustomer | customer101 | Customer |

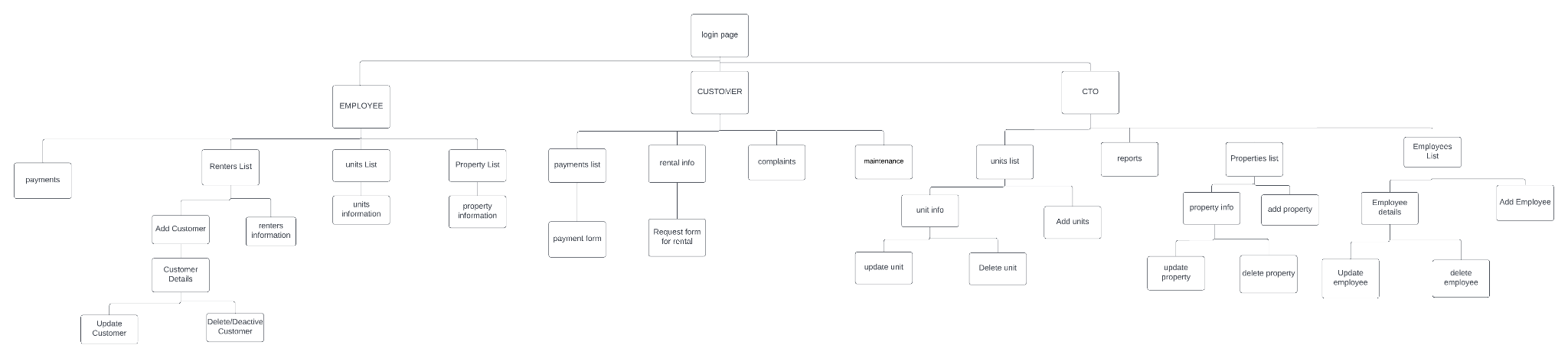
Link to YouTube presentation: <https://youtu.be/mRYldSlxwGs>.

# Use Case Requirements and Roles

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Use Case | Page Name | Role(s) | Description |
| 1 | Login | login | Employee  CTO  Customer | 1. User logs in to the website. 2. User is directed to the landing page. |
| 2 | Landing page | landing | Employee  CTO  Customer | 1. Due to the various uses of the website, the landing page provides a one-stop shop where users can navigate to other desired web-pages on the site, once logged in. 2. Show list of navigation options at top of page, nav will take users to renter-info, rental-app, payment, reports, maintenance, complaint, units, properties, add/deactivate employee or customer. |
| 3 | View renters | renters-list | Employee | 1. User views table of renters, their first name and last name, phone, active renter? status, lease party size, count of complaints against customer. 2. User may click on name of renter to be taken to renter-info page specific to customer. |
| 4 | View renter information | renter-info | Employee | 1. User views first name, last name, phone, email, DOB of renter. 2. User views active renter status, lease party size, count of complaints against customer if user is an employee. |
| 5 | Add customer | add-customer | Employee | 1. User views a form. 2. User may add a customer to the website by inputting first name, last name, phone, email, user status, and a customer ID into the form. |
| 6 | Submit rental application/lease extension requests | rental-app | Customer | 1. User views a form requesting their first and last name, phone, email, DOB, lease party size, unit type, and the date they’d like to begin their lease. 2. User is able to submit form after filling in all fields noted in 1 above. 3. Upon submission, user sees “Thank you” message. |
| 7 | Submit payment | payment | CTO  Customer | 1. User enters amount due, payment type (late fee, rent, etc.), due date, amount paid into form. 2. User submits form. 3. User sees “Thank you” message. |
| 8 | Generate/modify report/dashboard | reports | CTO | 1. User may view maintenance or financial report. 2. Financial report shows wages for employees. 3. Maintenance report shows completed and pending statuses. |
| 9 | Add employee | add-emp | CTO | 1. User sees form requesting userid, first name, last name, and employeeType. 2. User submits form. 3. New user is added to database. 4. User sees “Thank you” message. |
| 11 | Submit maintenance request | maintenance | Customer | 1. User sees form with unitID, date, a text box for the issue description. 2. User may fill out form and hit submit. 3. User sees “thank you” message upon submission. |
| 12 | Submit complaint | complaint | Customer | 1. User sees form with fields for type, description, and date and time of complaint. 2. User may submit form. 3. User sees “thank you” message upon submission. |
| 13 | View units list | unit-list | Employee  CTO | 1. User sees list of all units with unitID, propertyID, beds, baths, price, square footage, and occupancy. 2. User may click on UnitID to go to unit page specific to unit. |
| 14 | View unit information | unit | Employee  CTO | 1. User sees unitID, propertyID, beds, baths, price, square footage, and occupied? status for unit. 2. User may update price and/or occupied status and hit save button and it will update the database. |
| 15 | View property information | properties | Employee  CTO | 1. User sees list of properties with the name and employeeID of the manager of the property. |
| 16 | View payments | payment-list | CTO  Employee  Customer | 1. User sees list of payments with type, amount due, due date, overdue?, amount paid. (For employees they would see their paychecks here, for customers they would see payments they have made and need to make here). |
| 17 | Add unit | add-unit | CTO | 1. User sees form with unitID, propertyID, beds, baths, price, square footage, and occupancy to fill out. 2. User may click on save to add a new unit to the database. |
| 18 | Delete unit | delete-unit | CTO | 1. User sees page with unitID, propertyID, beds, baths, price, square footage, and occupancy and a delete checkbox. 2. User may check delete checkbox to delete unit from database. |
| 19 | Add property | add-property | CTO | 1. User sees form with the name and employeeID of the manager of the property. 2. User may click on save to add a property to the database. |
| 20 | Delete property | delete-property | CTO | 1. User sees page with the name and employeeID of the manager of the property and a delete checkbox.   2. User may check delete checkbox to delete unit from database. |
| 21 | Employee List | list-emp | CTO | 1. User sees list showing first name, last name, user type, user status, and employeeID. |
| 22 | Complaint list | complaint-list | CTO  Employee | 1. User sees list showing type, description, and date and time of complaint. 2. User may click on complaint ID to be taken to view-complaint page specific to the complaint. |
| 23 | View complaint | view-complaint | Employee | 1. User sees page showing type, description, and date and time of complaint. |
| 24 | Logout | logout | CTO  Employee  Customer | 1. User session is ended and they are prompted to re-login. |
| 25 | User | user | N/A | 1. This page finds the roles associated with each website user to enable proper authorization on each webpage. |
| 26 | Unauthorized | unauthorized | CTO  Employee  Customer | 1. This page displays a message to a user when they are not allowed to view a page due to their associated role. |
| 27 | Manage customer users | (Multiple pages: add-customer, edit-customer, renter-info, delete-customer) | Employee | 1. Add-customer adds a customer and user records to the database and generic username/password for that customer. Only employees can add customers. 2. Employees can view renter info. 3. Employees can update customer passwords. 4. Employees can delete customers (and their user records). |

# Sitemap and Wireframe Pages

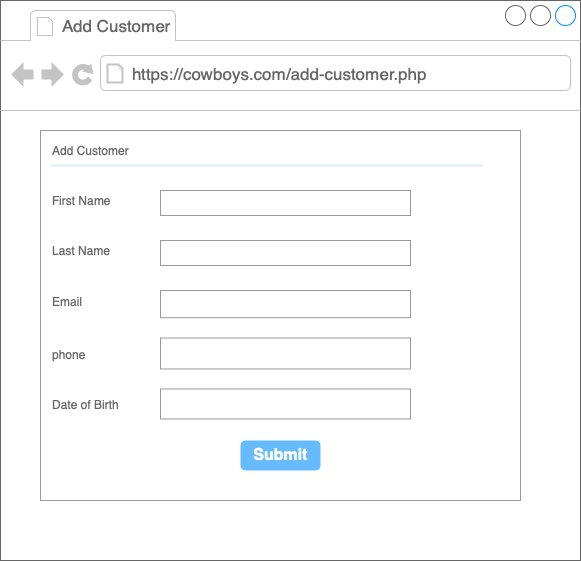
## Site Map

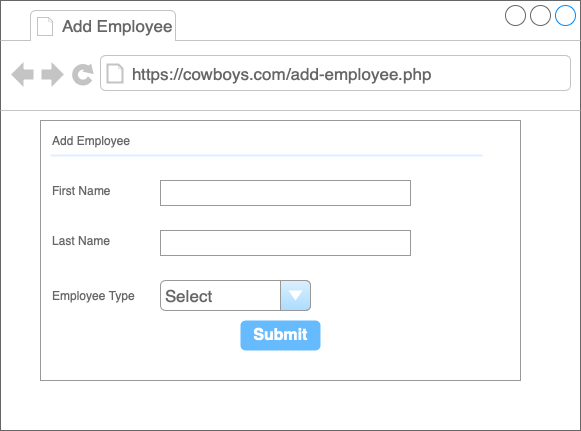


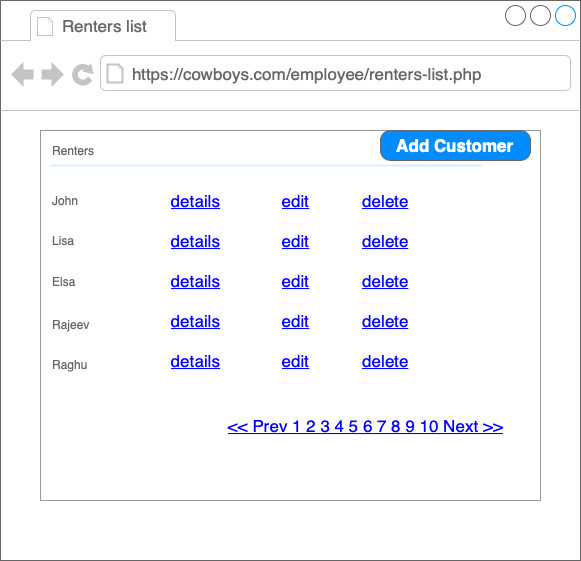
## Wire frames

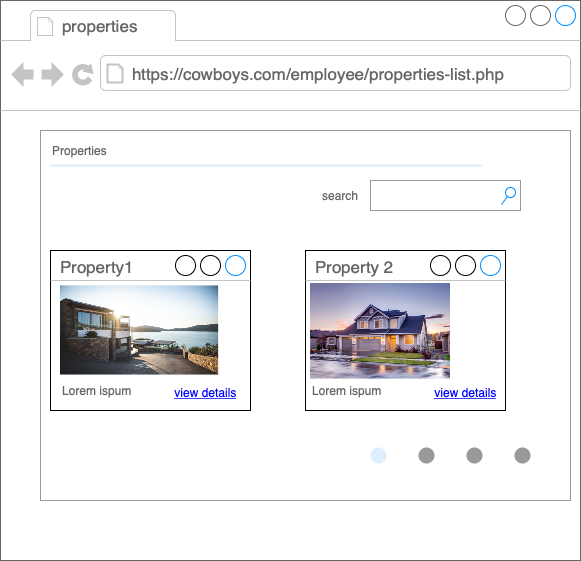
# 

# 

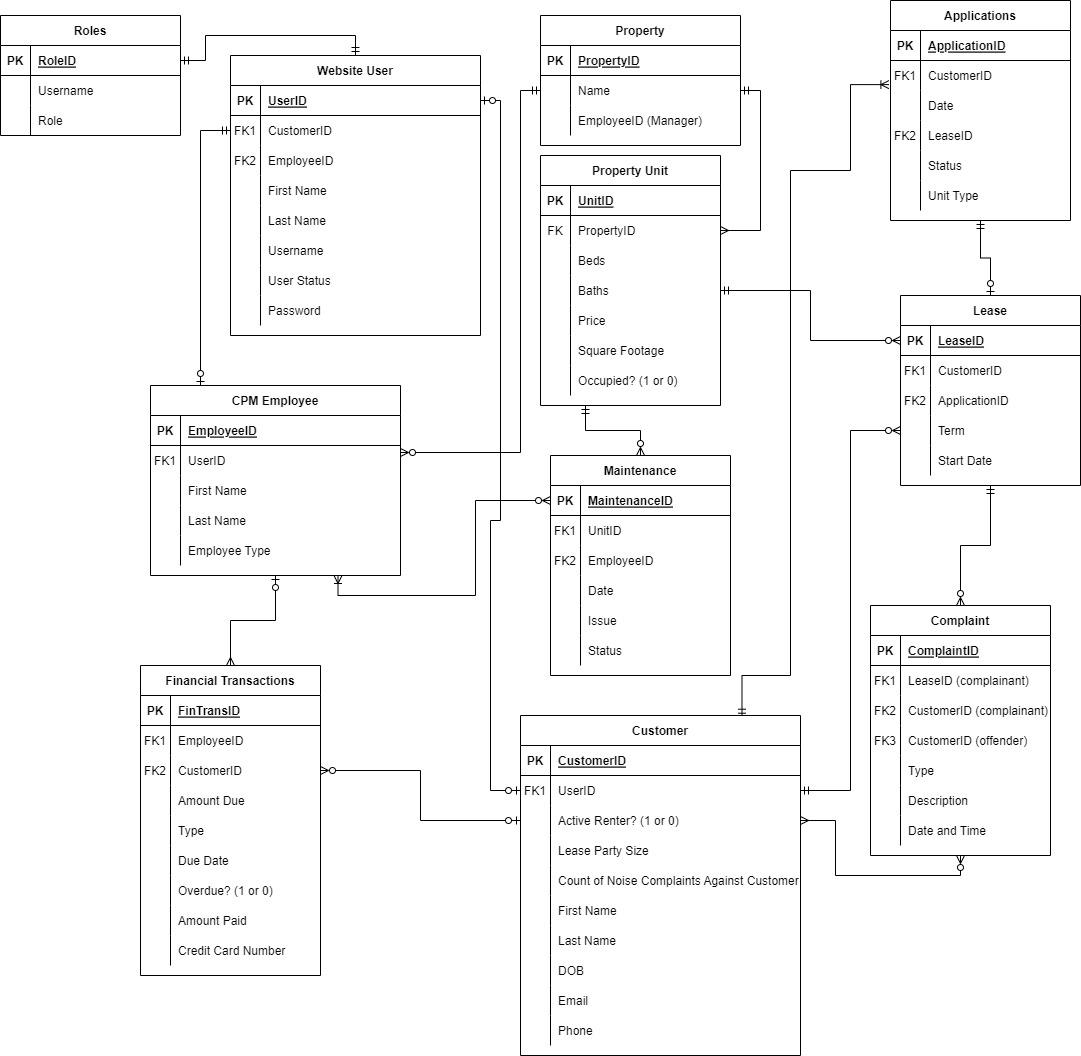








# Database ERD



# 

# Challenges and Tentative Delivery Schedule

Some challenges we are facing are that we are familiar with languages similar to PHP but none of us are experts and we have found finding times to meet difficult with our busy schedules.

Tentative delivery schedule:

|  |  |
| --- | --- |
| **Task** | **Due Date** |
| Coding first half of webpages in HTML/CSS | October 28 |
| Finalizing mid term project submission | November 3 |
| Code four additional use case webpages | November 10 |
| Set up database | November 10 |
| Ensure user management is working properly | November 17 |
| Draft presentation prepared | November 17 |
| Finalize presentation | November 24 |