**Group 12:**

**Bunch of Coders**

**Scrum Master**

Bobby Buxani

**Team Members**

Chandni Amin

Charlie McCollough

Riley Buckles

Emma Turner

Bobby Buxani

**Table of Contents**

[3.1.2. Executive Summary: An Expanded Version of your System Vision Statement](#_s1kjd9eyjm1k)

[3.1.3. Use Case Diagram and Domain Class Diagram](#_fyotn0fqiy5b)

[Use Case Diagram](#_la5vkm15a9l0)

[Domain Class Diagram](#_d7cm3l8sy246)

[3.1.4. System Prototype (Demo)](#_qtx7lsgqy0la)

[3.1.5. User Manual (Video)](#_fnqr4aoeulk3)

[3.1.6. Screenshots](#_4fro4xql30cg)

[3.1.7. Project Evaluation](#_kdvwnzdkpe04)

[3.1.8. Project Retrospective: Lessons Learned](#_y0a4vrdqlfu6)

**DELIVERABLE #3 (FINAL PROJECT DELIVERABLE)**

**3.1. Final Project Details**

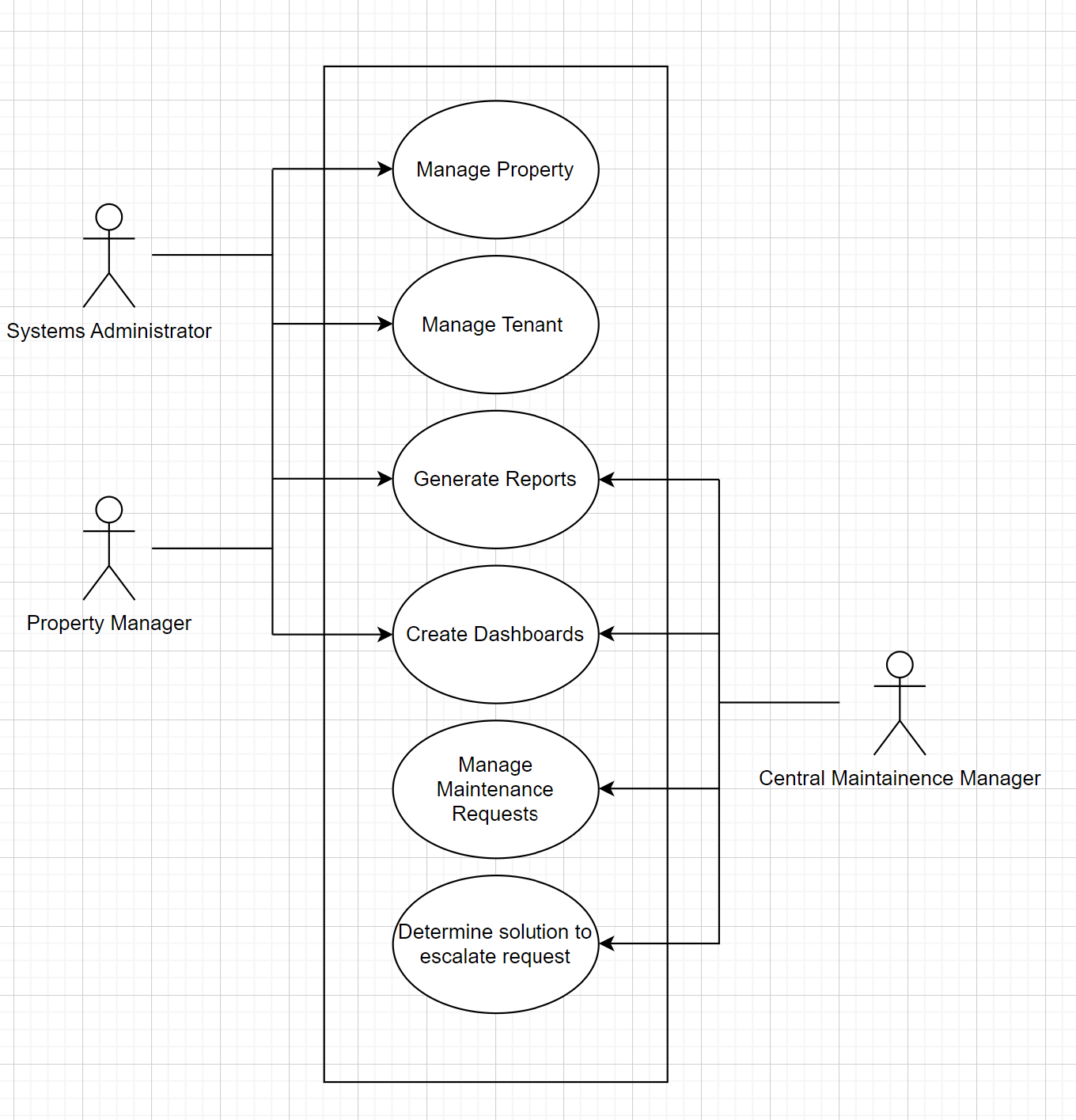
# **3.1.2. Executive Summary: An Expanded Version of your System Vision Statement**

RentoMax is a property management company that is currently using a manual process to track all their tenants and maintenance requests from their different properties throughout Atlanta, Georgia. Attempting to keep track of all of this data manually has lead to inefficiencies and errors that reflect on the maintenance crew and ultimately can harm the tenant. RentoMax has requested our team, Bunch of Coders, to create a centralized system that will track all maintenance requests for electrical and plumbing related issues that are sent in by tenants via phone, web, or email. Tracking the maintenance request information and storing related data on tenants (like unit number, phone, email. etc.) can minimize data errors and data loss by providing one place to enter and store that data, while supplying key information necessary to manage the RentoMax properties. Since all of the essential data a property manager, maintenance manager, and worker needs is always accessible in the system, issues can be serviced faster and with more efficiency. RentoMax also requested the system to track maintenance requests that are pending, in-progress, or resolved along with automatically notifying maintenance workers on their assigned units. These status and notification features will provide more visibility into the maintenance process and improve labor and resource allocation to each maintenance issue that arises. Our system will address all weaknesses in their process by providing a single platform to generate reports, keep track of request details, and take care of any other actions RentoMax may need.

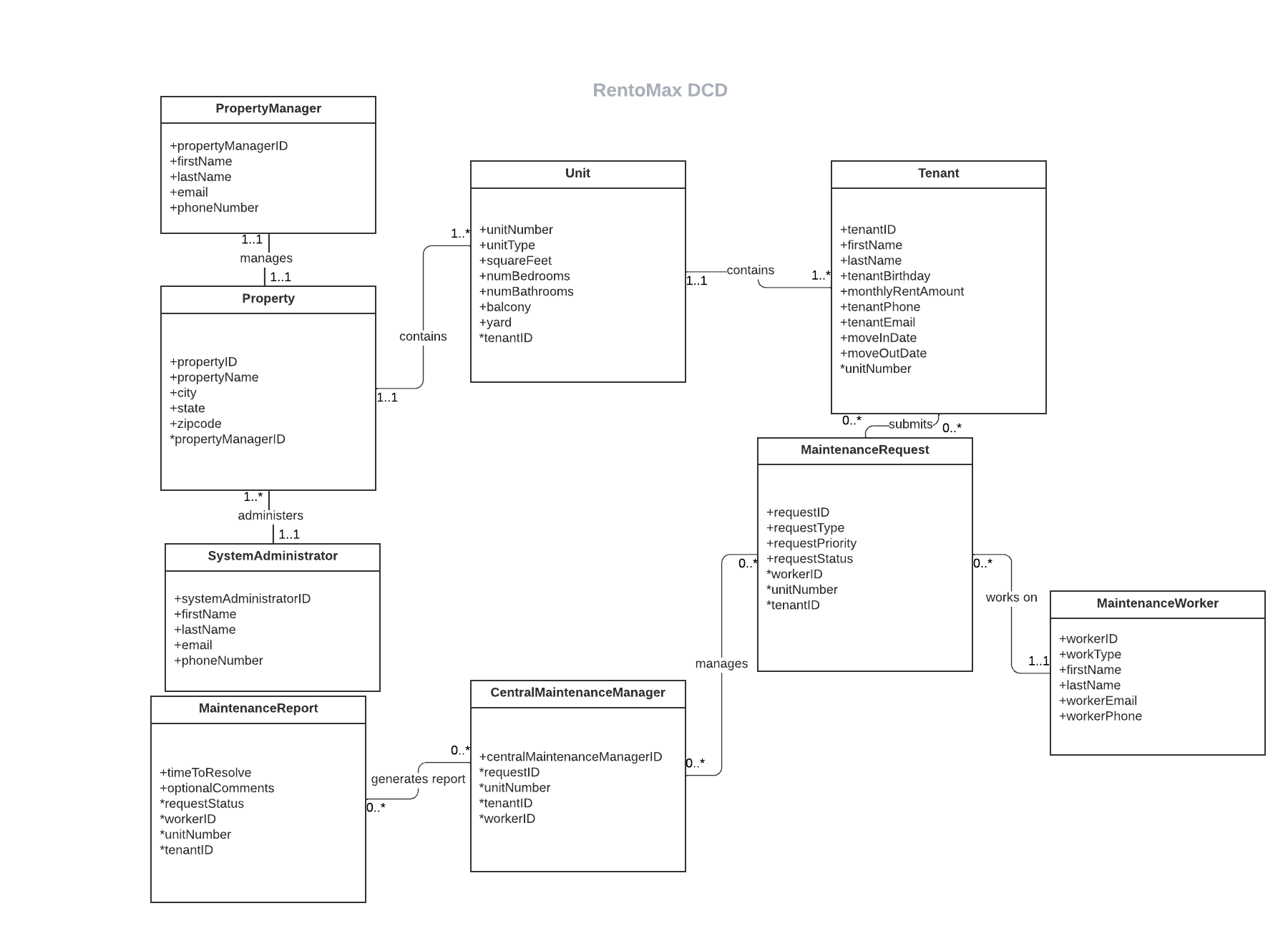
In our Salesforce-based system, we have created a maintenance request page where property managers can create a maintenance request based on information sent in by the tenant. This page captures all relevant data including the type of issue (electrical or plumbing), tenant information, as well as a space to assign a maintenance worker to the request and update its status. If a maintenance request has not been resolved within forty-eight hours, our email alert feature escalates the request to the central maintenance manager. These features aide in more efficient maintenance scheduling, and ensuring no issue is left unresolved. Outside of our maintenance request page, our team has also created a dedicated page for properties, units, and contacts. In each page, system users can view details such as the type, the manager of each property, the units within those properties, and tenants who reside in each unit. Specific details like contact information of tenants and system users, the square footage of a unit, unit occupation status, monthly rent, yard size, and more can be found in the “Details” section of each page. These pages are also dynamic, so users can create, update, and delete properties, units, contacts, and all corresponding details. Accessibility of this information in one place helps every member of the RentoMax team provide faster and better quality service to tenants. In the reports page system, users will find generated tenant reports that keep an accurate record of tenants residing in a specific unit type along with all the tenant’s information grouped by unit type and rental status. In this page there is also a property report that keeps an accurate record of the types of units that exist in each properties, along with pricing and important attributes to current and future residents. System users can also generate specific case reports that include maintenance request information like the property, unit, type, and any additional comments from the tenant or assigned maintenance worker. The dashboards page provides visibility into the maintenance request process as a whole and can be a tool for management to analyze what types of maintenance issues are occurring in certain locations, as well as how long they take to resolve. The reports and dashboards features are key for the team to focus on continually raising the standard of their service by analyzing weaknesses and adjusting their tactics as needed.

# **3.1.3. Use Case Diagram and Domain Class Diagram**

## **Use Case Diagram**



## **Domain Class Diagram**



# **3.1.4. System Prototype (Demo)**

Provide the URL link to your prototype and any information needed to access it (e.g., user name and password). I will deduct points if I cannot access your prototype. Also provide the URL for the web-accessible demonstration of your prototype.

Test the prototype by giving it to a couple of your friends who are not technically inclined and see if they can operate it without problems. Also test whether or not the prototype will run with various web browsers. The prototype should run on any reasonable browser.

Reports should look professional and neat, and should include example data that is sensible to your users. Reports should include appropriate headings for all columns. Your goal for the reports is to make sure that if someone picks up a report, he will be able to read for himself what the report is.

The prototype will be graded on its appearance (30 points), ease of use (40 points), and extent to which it accomplishes the logic of the use cases that you were assigned to prototype (100 points).

In recording the video that demonstrates your prototype, put yourselves in the stakeholders’ shoes and try to tell a story with your prototype. That is, present your use cases in a logical manner in terms of how they help carry out a business process or function (from beginning to end).

* + **Link (5 min vid): https://drive.google.com/file/d/11NLiV\_EaTBS8ZrDqmlTNITfUrbLqpuff/view?usp=share\_link**

# **3.1.5. User Manual (Video)**

Create a user manual (video) where you instruct the user on how to use the system. The user manual should include step-by-step instructions (everything from “go to website …” onward). A great way of doing this is through a video. This can be the same or different from the demonstration of your prototype. Remember: the objective of the demo video is to demonstrate the functionality of your system and it is limited to 5 minutes. A training video needs to provide detailed step by step instructions on how to use your system and may be much longer and much more detailed than your demo. If your demo can serve both purposes, then that’s fine. Provide the link to your demo. If not, provide the link to your training video.

-Training for system admin , property manager, maintenance

* **Link to User Manual Video: https://drive.google.com/file/d/1jREGMtOtPpItt7NFqVCFpKZfpNTm8JJL/view?usp=share\_link**

# **3.1.6. Screenshots**

Provide screenshots of screens in the prototype. Also provide a sample print of each report (if any) (either the printed report or a screenshot of the report). Organize and label these.

# **3.1.7. Project Evaluation**

Provide an honest evaluation of your system. What are you proud of? What can be improved? What would you have done differently in retrospect? Which requirements have you delivered?

Which not? Why? What’s went well? What did not go well as it should?

Include any feedback from potential users of the system on your final prototype.

# **3.1.8. Project Retrospective: Lessons Learned**

One of the things we don’t do enough in systems development is project assessment. Therefore, I would like you to reflect upon “Lessons Learned” from the project and discuss those. These can be in terms of learning various aspects of SA&D (e.g., how requirements can evolve, how to manage scope creep, difficulty in communicating with clients, usefulness of use case diagrams, not properly eliciting requirements or assessing technical feasibility up front, etc) or how to work better in teams, or project management. What do you think worked effectively? What do you think you would change next time and how?

**3.1.9.Group Member Peer Evaluation**

We take peer evaluations very seriously. You need to submit these to the appropriate eLC.