**Group 12:**

**Bunch of Coders**

**Scrum Master**

Bobby Buxani

**Team Members**

Chandni Amin

Charlie McCollough

Riley Buckles

Emma Turner

Bobby Buxani

**Table of Contents**

[2.1.1 Prioritized Product Backlog](#_mjz6qshhv5nz)

[2.1.2 An updated Use Case Diagram, brief Use Case Description table, and Domain Class Diagram](#_jwjsqqchy9gl)

[Revised Use Case Diagram](#_3otvozf3udwg)

[Revised Use Case Description Table](#_xutsgp7qbh80)

[Revised Domain Class Diagram](#_bp0n47e3t42u)

[2.1.3 Starting Sprint Backlog](#_m3torpbkowmz)

[2.1.4 A Brief Description of all User Stories Developed in the Sprint](#_nwdruk19saca)

[2.1.5 Acceptance Criteria for each User Story in the Sprint](#_e2ba06j2029)

[2.1.6 Sprint Review - Tutorial Screencast](#_ye4njwuy1ems)

[2.1.7 Sprint Retrospective](#_jwy5i13ygt72)

**DELIVERABLE #2**

**2.1. Structure of Sprint**

# 

# 2.1.1 Prioritized Product Backlog

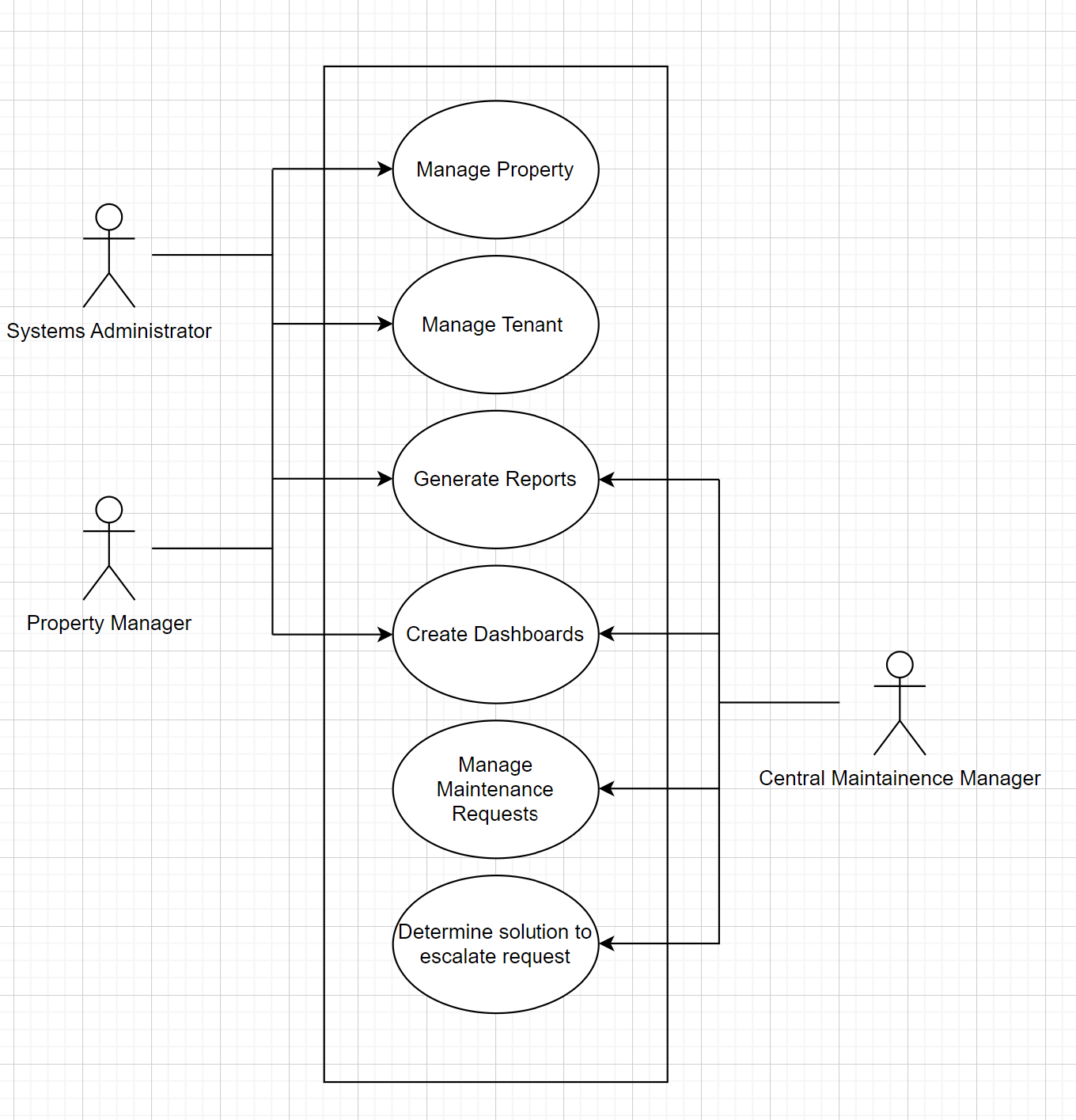
**Product Backlog**

**Sprint 1 Sprint 2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Number | Use Case | As a… | I want to… | So that I can… |
| 1 | Create Data Structures (non-functional) | System User (Property Manager /Administrator/Maintenance Manager) | Set up data models and tables | Store maintenance and tenant information along with associations between entities |
| 2 | Create Logins (non-functional) | System User (Property Manager /Administrator/Maintenance Manager) | Have a secure login and authentication process | Ensure security of sensitive information |
| 3 | Increase Scalability (non-functional) | System User (Property Manager /Administrator/Maintenance Manager) | Add more properties, units, tenants, and requests to the system | Performance isn’t compromised or slowed down |
| 4 | Manage Property (Must-have) | Property Manager | Create, update, and delete properties | Have accurate knowledge of which properties I’m responsible for and to have property-specific information |
| 5 | Manage Unit (Must-have) | Property Manager | Create units of different types, manage their attributes, and associate them with properties | I can provide unit-specific information like monthly rent and rental status for current and future tenants |
| 6 | Manage Tenant (Must-have) | Property Manager | Create, update, delete tenants, associate them to a unit in a property, and capture tenant information like name, **payment information, move in/out dates** | Keep track of current residents, be able to contact them, and be aware when new units will open after tenants move out |
| 7 | Manage Maintenance Requests (Must-have) | Central Maintenance Manager | Enter requests that come in by phone, set status of requests, assign workers, and escalate after 48 hours if not resolved in that timeframe | Make sure each request is being assigned to a specific worker and ensure jobs are being completed efficiently |
| 8 | Determine solution to escalated request | Central Maintenance Manager | Brainstorm possible solutions | Complete the escalated request as quickly and efficiently as possible |
| 9 | Implement maintenance solution | Central Maintenance Manager | Execute the best determined solution | Alleviate the tenant’s maintenance request |
| 10 | Monitor solution to escalated request | Central Maintenance Manager | Monitor and assess that the solution worked | The tenant does not have the same problem again |
| 11 | Generate Tenant Report (Must-have)  DONE | System User (Property Manager /Administrator/Maintenance Manager) | Generate reports of tenant information (name, email) grouped by unit type and/or rental status | Keep accurate records of tenants residing in specific unit types |
| 12 | Generate Property Report (Must-have)  DONE | System User (Property Manager /Administrator/Maintenance Manager) | Generate reports of unit information (type, rental status, rent) grouped by property | Keep accurate records of what types of units exist in which properties, along with pricing and other attributes important to current and future residents |
| 13 | Generate Case Report (Must-have) | System User (Property Manager /Administrator/Maintenance Manager) | Generate reports with case information like type (electrical or plumbing) and date opened grouped by type and/or property/unit | Understand and visualize what types of maintenance issues are appearing in certain units/properties |
| 14 | Create Dashboards (Should-have)  DONE | System Administrator/RentoMax | Visualize pending maintenance requests and maintenance requests completed | Determine efficiency of maintenance resources and make any changes as needed |
| 15 | Allow tenants to submit maintenance requests by email (Could-have) | Tenant | Submit maintenance requests by email | Enter my request in a faster, more convenient way |
| 16 | Allow tenants to submit maintenance requests by web (Could-have) | Tenant | Submit maintenance requests by web | Enter my request in a faster, more convenient way |

# 2.1.2 An updated Use Case Diagram, brief Use Case Description table, and Domain Class Diagram

## Revised Use Case Diagram



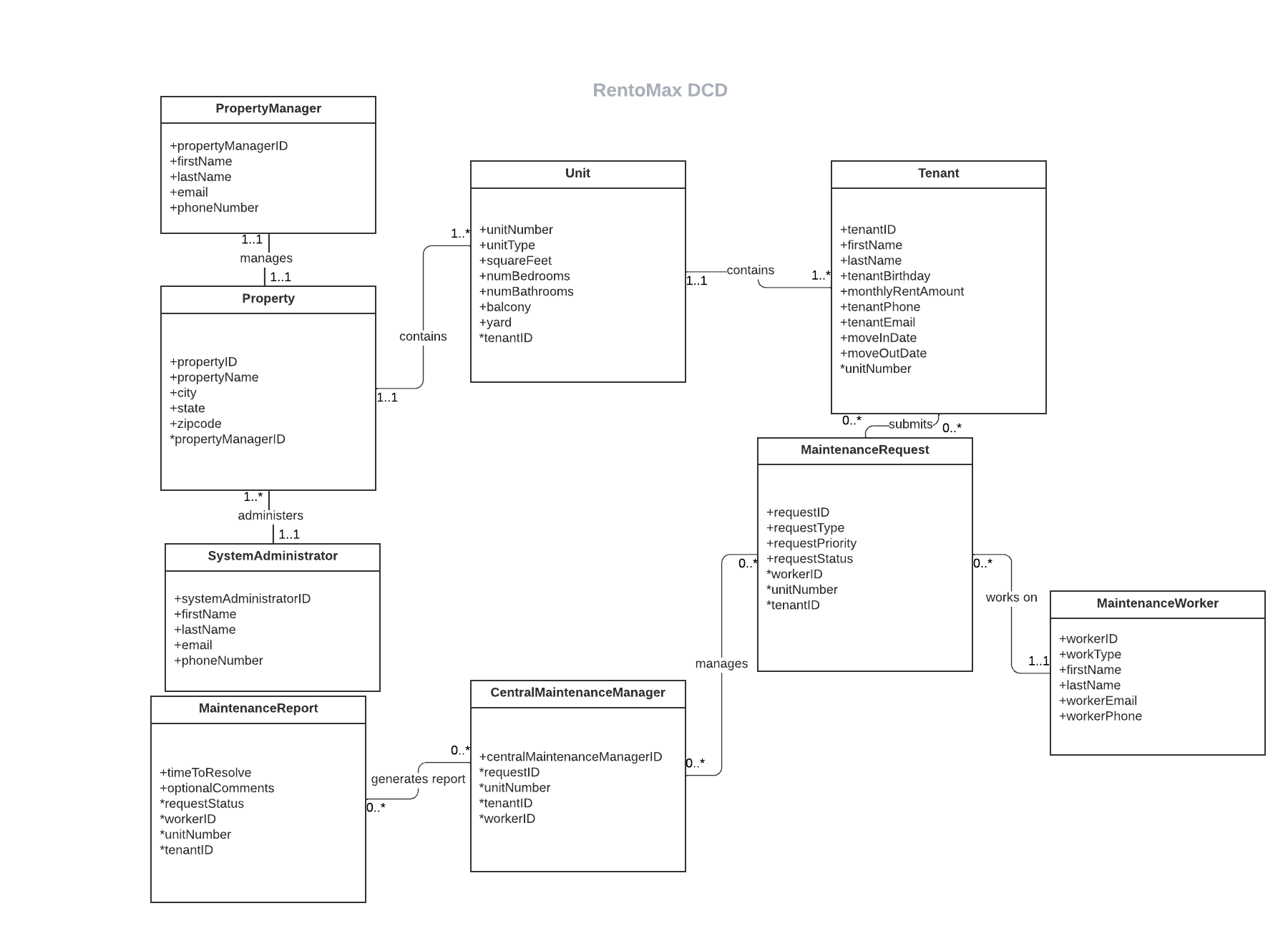
## Revised Use Case Description Table

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Brief Description** | **Actors** |
| Manage Property | The property manager will have the ability to keep track of the properties. The information that will be included is: number of bedrooms, number of bathrooms, whether the unit has a balcony, and square footage. | Property Manager, System Administrator |
| Manage Tenant | The property manager will have the ability to keep track of the properties’ tenants. The information that will be included is: name, age, etc. | Property Manager,  System Administrator |
| Manage Maintenance Requests | The central maintenance manager will have the ability to view requests, assign them to workers, update their status, escalate requests after 48 hours, etc. | Central Maintenance Manager |
| Generate Reports | The system should keep track of request details such as time to resolve maintenance requests and status of maintenance requests | System Administrator,  Central Maintenance Manager,  Property Manager |
| Generate Dashboard | The system should also allow RentoMax to see information such as pending maintenance requests and maintenance requests completed. | System Administrator, Central Maintenance Manager,  Property Manager |
| Determine solution to escalated request | The Central Maintenance Manager must be able to deal with escalated requests and determine appropriate solutions. | Central Maintenance Manager |

## 

## 

## Revised Domain Class Diagram



# 2.1.3 Starting Sprint Backlog

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story** | **Tasks** | **Team Member** | **Time** |
| Scrum Artifacts | Create sprint backlog | Riley | 20 min |
| Create Data Structures | Create entities for properties, units, and tenants | Everyone | 1.5 hours |
| Create relationships between entities | Everyone | 1.5 hours |
| Create Logins | Create contacts for all system users | Chandni | 30 min |
| Add information about each contact | Chandni | 20 min |
| Create logins for all system users | Chandni | 20 min |
| Increase scalability | Ensure that more properties, units, and tenants can be added as needed | Everyone | 30 min |
| Manage Property | Create properties (objects) that are under the RentoMax conglomerate | Bobby | 45 min |
| Add information about each property (name, address, etc.) | Bobby | 30 min |
| Manage Unit | Create custom objects for each unit | Charlie | 45 min |
| Add information about each unit (type, number of rooms, etc.) | Charlie | 30 min |
| Add ability to change unit status (Rented, Available, Unavailable) | Charlie | 20 min |
| Manage Tenant | Add contact information (phone, email, etc.) | Riley | 20 min |
| Be able to connect tenants to their unit and property | Riley | 30 min |
| Manage Maintenance Requests | Create form to enter maintenance requests | Emma | 2 hours |
| Create record types for maintenance issues (electrical or plumbing) | Emma | 20 min |
| Add ability to change request status (pending, in-process, resolved) | Emma | 20 min |
| Create escalation rules for requests open > 48 hours | Emma | 30 min |
| Determine Solution to Escalated Request | Brainstorm | Everyone | 20 min |
| Scrum Artifacts | Tutorial screencast | Chandni, Bobby, Charlie | 30 min |
| Sprint retrospective write-up | Riley | 30 min |

# 

# 2.1.4 A Brief Description of all User Stories Developed in the Sprint

# 2.1.5 Acceptance Criteria for each User Story in the Sprint

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story Name** | **Brief Description** | **Acceptance Criteria** | **Met?** |
| Create Data Structures | Create the necessary entities and their relationships | -Must have all necessary entities and their attributes  -Must be able to link entities through relationship | Yes |
| Create Logins | Create login credentials for each user | -Must have all necessary types of system users  -Each user must have a unique and secure login | Yes |
| Increase Scalability | Be able to add new properties, units, tenants, etc. as needed | -Must have ability to add new objects whenever it is needed | Yes |
| Manage Property | Be able to view, create, update, and delete properties and their attributes | -Must be able to create a new property  -Properties should have relevant attributes (e.g. address, name)  -A property consists of multiple units.  -Must be able to update a property  -Must be able to delete a property (and associated units) | Yes |
| Manage Unit | Be able to view, add, update, and delete units and their information | -Must be able to create a new unit and associate it with its property  -Units can be one of two types - apartment or townhouse  -Townhouses may have a yard and yard area. Apartments do not have yards.  -Each unit should have relevant attributes (number of bedrooms, number of bathrooms, square footage, etc.)  -Must be able to update an existing unit  -Must be able to delete a unit | Yes |
| Manage Tenant | Be able to view, add, update, and delete tenants and their information | -Must be able to create a new tenant. Each tenant needs to have one person’s name. Each tenant should be assigned to a unit in a property.  -A tenant should have relevant attributes (name, date of birth, etc.)  -Must be able to update an existing tenant  -Must be able to delete a tenant | Yes |
| Manage Maintenance Requests | Be able to view, create, update, and delete maintenance requests as well as escalate requests when needed | -The system user must be able to enter information for maintenance requests that come by phone from tenants  -Maintenance requests should have a status of “pending, in-process, or resolved”  -Maintenance requests should have relevant attributes (type, reason, tenant name, etc.)  -Once a new maintenance request arrives, and email notification should be sent to the respective maintenance person  -If a maintenance request is not resolved for more than 48 hours, it needs to be escalated to the system administrator | No  We were unable to set up email notifications once requests have been assigned due to insufficient Salesforce licenses. We will continue to work on this and plan to implement a solution in sprint 2. |
| Determine Solution to Escalated Requests | Brainstorm possible solutions to maintenance request escalations | -Must be able to understand and implement solutions for maintenance requests that have not been resolved | Yes |

# 2.1.6 Sprint Review - Tutorial Screencast

* Here is the URL to our salesforce sprint one video.
  + <https://youtu.be/Ez6aMcgRD5w>

# 

# 2.1.7 Sprint Retrospective

During our sprint retrospective, our team discussed our performance throughout the sprint, including any issues we faced as well as all of the positive observations about our work. By doing this, we were able to pinpoint what we want to continue doing throughout the next sprint, what we should start doing, and what we need to stop doing.

|  |  |
| --- | --- |
| **Continue doing** | |
| Communication | We did a great job communicating as a team. Our members were very responsive in the group chat and were open to discussing any issues or questions that were brought up. |
| Time management | We started our sprint early in the week which helped us not be as stressed when we ran into issues using salesforce. We also planned our meetings around when office hours were being held so that we would have questions to ask the Salesforce professionals. |
| Delegation | We split up the work and delegated it to members of the team so that we would be able to work on the project whenever each member had time as well as so that no work accidentally got done twice. We created a shared document where we could see the work that everybody had gotten done and share any issues that we were having. |
| Respect | Our team had great respect for each other which helped us avoid any clashes. We understood each other’s time commitments and made sure that we were all contributing to the project. We also felt comfortable with each other and respected others’ opinions even if they differed from our own. |

|  |  |
| --- | --- |
| **Start doing** | |
| Better preparation | We felt as though we could have prepared for Salesforce better. We spent a lot of time trying to figure out the basics of using Salesforce when we should have been working on the specific aspects of our project by that point. For the next sprint, we want to dedicate time to research exactly what we need to get done in Salesforce and the basics of how to do it before we try to implement any solutions. |
| Check-ins | Our team had an idea of what we needed to get done from the beginning and we focused on that idea without much flexibility. We ran into some frustrating issues with how we were trying to use Salesforce that we spent a lot of time trying to work out without much progress. For the next sprint, we want to do more check-ins with the developers and be more flexible when it comes to issues we face. |
| Discuss requirements | We ran into issues when some of the requirements were confusing or we did not understand exactly what we needed to do. Additionally, because this is an agile project, the requirements sometimes changed and we were not aware of this without talking to stakeholders. For the next sprint, we want to be better about discussing the requirements with our stakeholders and asking for clarification when needed. |

|  |  |
| --- | --- |
| **Stop doing** | |
| Zoom meetings | Our team found that we are much more productive in person rather than virtually. We attempted to conduct some of our meetings via Zoom, but realized that it is more difficult to communicate and work together when we are not physically in the same room. For the next sprint, we will ensure that we find time to meet in person whenever we need to work on the project. |
| Making assumptions | We ran into some issues where team members would make assumptions about what to do in Salesforce without everyone being on the same page. This caused us to have to do extra work to get back on track. For the next sprint, we want to have a clear outline of what work needs to get done so that nobody has to make assumptions that may hinder our progress. |