

Stonebridge Property Group

ServiceNow Administrator Fundamentals Capstone Project

Background

The Stonebridge Property Group manages commercial office buildings, coworking hubs, and shared spaces across several metropolitan areas. They pride themselves on having responsive, in-house facility teams who maintain the lighting, HVAC, janitorial, and general repair services. Until now, their facility issues were reported via email or paper forms – a process riddled with delays and lost requests.

You have been hired to create a centralized facility maintenance request system in ServiceNow to track and assign building maintenance tasks. This internal-facing app will serve both employees submitting issues and the facility team resolving them.

User Stories

As a clerk, I want to submit a maintenance request through a custom application form so I can report facility issues easily	<ul style="list-style-type: none">• Create a new application using App Engine Studio• Create a table for maintenance requests• Create an application menu and a module to create a new maintenance request
As a manager, I want to approve maintenance requests before they are marked as 'complete'	<ul style="list-style-type: none">• Create a flow for submitted maintenance requests• Include an approval step for managers to approve requests
As a facility technician, I want to view only requests assigned to me so I can focus on my own work	<ul style="list-style-type: none">• Assign tasks to groups based on request type in the maintenance request flow• Apply the necessary ACLs
As a requestor, I want to receive an email notification when my maintenance request is approved	<ul style="list-style-type: none">• Use an email script to build a notification• The recipient should be the requestor (Aka the person who triggered the flow)

Tasks:

- Create a new application called 'Stonebridge PG Maintenance'
- Create a new application menu for the application
- Create the necessary tables (**Note that these are the MINIMUM required tables.

Add additional tables and columns as you deem necessary**):

- Maintenance Request
 - Date
 - Time
 - Title
 - Description
 - Type:
 - Electrical
 - HVAC
 - Janitorial
 - General Repair
 - Requestor (reference to user)
 - Task (reference to Maintenance Task)
 - Priority
 - 1 – low
 - 2 – medium
 - 3 – high
 - Property (reference to Property)
 - State:
 - Active
 - Completed
 - Canceled
- Maintenance Task (Hint: this can extend the base 'task' table)
 - Title
 - Type:
 - Electrical
 - HVAC
 - Janitorial
 - General Repair
 - Technician (user)
 - Parent Request (reference to Maintenance Request)
- Property

- Name
 - Street Address
 - Unit #
 - Phone Number
 - City
 - Zip
 - State/province
 - Country
- Create the following groups and roles and assign AT LEAST 2 users to each:
 - SB Users(group)
 - SB Users(role)
 - SB Clerks (group)
 - Clerk (role)
 - SB Technicians (group)
 - Janitors(group)
 - Janitor(role)
 - HVAC(group)
 - HVAC(role)
 - Electricians(group)
 - Electrician(role)
 - SB Managers(group)
 - SB Managers(role)
- Create the following modules in the application menu:
 - Open Requests (view all requests where State = Active)
 - Visible to technicians/managers
 - My Requests (view all requests where requestor = me)
 - Visible to clerks/technicians/managers
 - Submit Request (open new maintenance request form)
 - Visible to clerks/technicians/managers
 - Maintenance Tasks (view all active maintenance tasks)
 - Visible to technicians/managers
 - My Maintenance Tasks (view all maintenance tasks where technician = me)
 - Visible to technicians
 - Urgent Requests (view all maintenance requests where priority = 3-high0)
 - Visible to managers
- Create a flow that is triggered when a new maintenance request is made
 - Based on request type, create a new task record for that type:

- If request type = electrical, create a new maintenance task record with type set to electrical and route to the electricians group
 - If request type = HVAC, create a new maintenance task record with type set to HVAC and route to the HVAC group
 - If the request type = janitorial, create a new maintenance task record with type set to Janitorial and route to the Janitors group
 - If the request type = general repair, create a new maintenance task record with type set to general repair and assign to the Technicians group
- Next, the assigned user must fill out the details of the form and submit it to progress the flow
- Next, an email script notification should be sent to the requestor that includes important details about the request and the associated task
- Finally, the manager should approve that the task has been completed and close the request and task records
- Create a UI Action that adds a 'Mark Complete' button on the Maintenance Task table
- Create a Data Policy that makes Type, Property, Priority, and Title mandatory on the Maintenance Request table
- Add a UI Policy that makes the 'state' field hidden when it is empty (the flow should set the state field to 'Active' automatically)
- Create the appropriate ACLs:
 - Requestors: create/view only their own requests, can view property table
 - Technicians: view/modify assigned tasks and create new tasks, can view property table
 - Managers: full access to requests, tasks, and property tables
- ALL development should be done within the application scope
- ALL changes should be tracked in an update set (NOT the default update set)
- Create SLAs for each activity in the flow (use your own judgement on how long to set the SLA timings for)

Other Considerations

This project is designed to imitate a real-world scenario with a client to prepare you for your future development projects using ServiceNow. The tasks listed above are not exhaustive when it comes to real-world development. Use your own judgement to implement other features that better mimic a real application. This is also an opportunity

to practice and push yourself to make an application using all of the skills you have acquired over the course. There may be some things that you need to research on your own in order to complete the tasks.

Try to push yourself to do more than just what is laid out in this document. Try creating something in the UI builder. Try making a report or a mobile app. Create different views for different types of users. Fine tune the layout of your forms. Show us what you can do!

Please be prepared to present your work to the group on the due date provided