Service Project

Create Service Request Use Case

User story

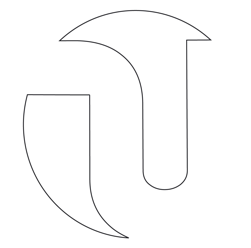
|  |
| --- |
| ***I as User need a way to report issues to both Property Managers and the public for the goal of getting the issues fixed.*** |

Abstract

|  |
| --- |
| ***This Create Service Request Use Case document specifies all requirements related to this use case, including the flows through the use case and any non-functional requirements that relate specifically to this use case. Service requests are the lifeblood of the Service application, serving as the means by which users document issues in need of service and share the information with other users and property managers.*** |

Version History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Description** |
| 30/10/2016 | 0.1 | Gavin Wolf | First pass through template. |
| 30/10/2016 | 0.2 | Gavin Wolf | Add/revise text content. |
| 01/11/2016 | 0.3 | Gavin Wolf | Work through Cockburn template. |
| 01/11/2016 | 0.4 | Gavin Wolf | Add use case diagram. |
| 01/11/2016 | 0.5 | Gavin Wolf | Work on incomplete sections. |
| 03/11/2016 | 1.0 | Gavin Wolf | Update per discussion w/ Dr. Huang. |
| 05/11/2016 | 1.1 | Gavin Wolf | Finishing touches. |



An Essential Unified Process Document

*Table of Contents*

[1 Introduction 3](#_Toc466182439)

[1.1 Document Purpose 3](#_Toc466182440)

[1.2 Document Scope 3](#_Toc466182441)

[1.3 Document Overview 3](#_Toc466182442)

[2 Use Case Diagrams 3](#_Toc466182443)

[2.1 Create Service Request Use Case Diagram 3](#_Toc466182444)

[3 Actor Catalog 4](#_Toc466182445)

[4 Use Case Catalog 5](#_Toc466182446)

[5 Use case Specification 5](#_Toc466182447)

[5.1 Introduction 6](#_Toc466182448)

[5.1.1 Document Purpose 6](#_Toc466182449)

[5.1.2 Document Scope 6](#_Toc466182450)

[5.1.3 Document Overview 6](#_Toc466182451)

[5.2 Brief Description 6](#_Toc466182452)

[5.3 Preconditions 7](#_Toc466182453)

[5.4 Basic Flow 7](#_Toc466182454)

[5.5 Alternative Flows 7](#_Toc466182455)

[5.5.1 Handle Database Error 7](#_Toc466182456)

[5.5.2 Handle Email System Error 7](#_Toc466182457)

[5.6 Post-Conditions 8](#_Toc466182458)

[5.7 Special Requirements 8](#_Toc466182459)

[5.8 Extensibility 8](#_Toc466182460)

[5.9 Scenarios 8](#_Toc466182461)

[6 Use case specification (Cockburn template) 9](#_Toc466182462)

# Introduction

## Document Purpose

The purpose of this document is to provide an overview of the system use case model in order to provide a high-level understanding of:

* Context – the people or things that interact with the system (the *Actors*)
* Scope – the things of value that the system performs for its Actors (the *Use Cases*).

## Document Scope

The scope of this document is limited to:

* Diagramming and cataloging the system actors and use cases for the Service project.

The scope of this document does *not* include consideration of:

* Detailed specification of each use case – this is provided separately in a Use Case Specification document for each use case.

## Document Overview

This document contains the following sections:

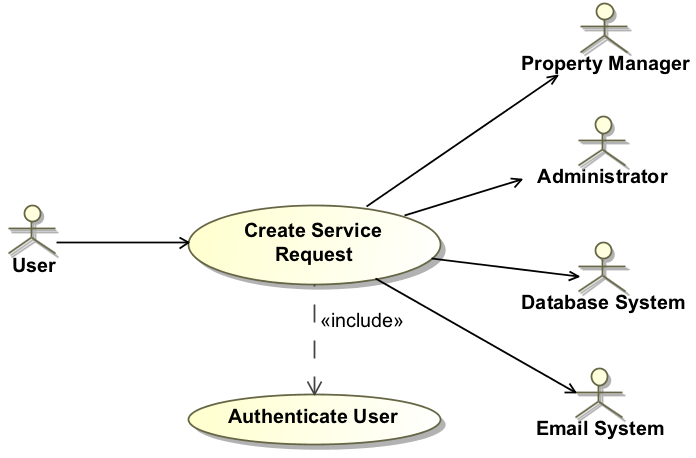
* **Brief Description** – reminder of why the use case is needed
* **Create Service Request Use Case Overview** – overview of the Create Service Request use case in the form of a use case diagram with supporting explanatory text
* **Actor Catalog** – catalog of all system actors
* **Use Case Catalog** – catalog of all system use cases
* **References** – provides full reference details for all documents, white papers and books that are referenced by this document.

Use Case Diagrams

## Create Service Request Use Case Diagram

A user creates a service request and the property manager for the related property is alerted of the new service request. The database system stores a record for the new service request. The email system handles the alert email to the property manager, as well as a confirmation email to the user. If this process does not go as planned, the errors are logged for the administrator to see. The Authenticate User use case is included to authenticate the user’s credentials.

Figure 1 – Use Case Diagram



Actor Catalog

The table below catalogs the system actors, specifying for each actor:

* Name – unique and meaningful name for the actor
* Brief Description – summarizing the role that the actor plays with respect to the system.

| Name | Brief Description |
| --- | --- |
| User | A person who has an account with Service and can create service requests. |
| Property manager | Is alerted when a new Service Request is created for the property they manage. |
| Administrator | Is responsible for monitoring the error log and addressing any issues that arise in the process of a new service request being created. |
| Database system | Stores service requests and user information. |
| Email system | Sends an email to the user confirming that their service request was received and to the relevant property owner notifying them that a service request for their property has been created. |

Use Case Catalog

The table below catalogs the system use cases, specifying for each use case:

* ID – unique identifier for the use case
* Name – unique and meaningful name for the use case
* Type:
  + *Base – end-to end interaction between an actor and the system*
  + *Inclusion – abstracted common part of many Base Use Cases that is explicitly referenced by these use cases*
  + *Extension – extension of one or more referenced Base Use Cases*
  + *Abstract – use case describing generic aspects of many Base Use Cases which in turn specialize this general case in some way.*
* Brief Description – summarizes the use case purpose in terms of the value produced for its actors and other stakeholders.

| ID | Name | Type | Brief Description |
| --- | --- | --- | --- |
| 1 | Create account | Base | Allows a person to register as a user, giving them access to use Service. |
| 2 | View service requests | Base | Allows users to view the service requests that have been created. |
| 3 | Create service request | Base | Allows a user to create a service request to communicate an issue that needs to be fixed. |
| 4 | Respond to service request | Base | Allows a property manager to communicate with users about the status of a service request. |
| 5 | Delete user | Base | Allows an administrator to remove users and posts that do not conform to Service’s policies. |
| 6 | Authenticate user | Inclusion | Allows a user to be identified and given access to Service. |

# Use case Specification

|  |
| --- |
| ***The Create Service Request Use Case specifies all requirements relating to this use case, including the flows through the use case and any non-functional requirements that relate specifically to this use case.*** |

## Introduction

### Document Purpose

The primary objectives of the Use Case Specification are to:

* Provide a complete set of requirements relating to the use case, including the flows through the use case and all significant external behavior of the use case
* Provide additional supporting information to support the communication of the purpose, requirements, constraints and flows through the use case.

### Document Scope

The scope of this document is limited to consideration of:

* The specification this specific use case, including any non-functional requirements and constraints that relate specifically to this use case.

This scope of this document does not include consideration of:

* Requirements that relate specifically to any other use case – these are described in separate specifications for the other use cases
* Requirements that do not relate to any specific use case – these are described in a separate Supplementary Requirements document.

### Document Overview

This document contains the following sections:

* **Brief Description** – focusing on the purpose of the use case
* **Preconditions** – the state the system must be in before the use case can performed
* **Basic Flow** – what normally happens when the use case is performed
* **Alternative Flows** – unusual, optional or exceptional use case behavior
* **Post-Conditions** –valid system states after the use case has finished
* **Special Requirements** – non-functional requirements that relate specifically to this use case
* **Extensibility** – where the use case can be extended by other use cases
* **Scenarios** – lists the key representative scenarios for this use case
* **Additional Information** – any other useful supporting information
* **References** – provides full reference details for all documents, white papers and books that are referenced by this document.

## Brief Description

Users will be able to create service requests that will be saved, visible to other users and communicated to the manager of the related property.

## Preconditions

* The user must be logged into the system
* The website must be connected to the system’s database
* The email system must be functioning normally

## Basic Flow

**{Select Create Service Request}**

1. The use case begins when a logged-in **user** selects the option in the navigation bar to create a service request.

**{Fill Out Service Request Form}**

1. The user populates the service request form, which includes fields for: “Location”, “Description”, and “Upload Photo”.
2. The user clicks the submit button.

**{Add Service Request to Database}**

1. The information submitted by the user is saved to the **database system**.

**{Send Emails}**

1. The **email system** sends an email to the user confirming that their service request was received.
2. The email system sends an email to the **property manager** (if one is on record for the particular property) confirming that their service request was received.

**{Use Case Ends}**

1. The use case ends.

## Alternative Flows

### Handle Database Error

At **{Add Service Request to Database}** the save operation is unsuccessful.

* + - 1. The user is informed that an error has occurred and they should try again at a later time.
      2. The system creates an **error log** entry to record the fact that a database error occurred. The error log entry includes information about the user and the service request.
      3. The use case resumes the basic flow at **{Use case ends}**

### Handle Email System Error

At **{Send Emails}** the email operation is unsuccessful.

1. The system creates an **error log** entry to record the fact that an email system error occurred. The error log entry includes information about the user and the service request.
2. The use case resumes the basic flow at **{Use case ends}**

## Post-Conditions

* The service request has been saved to the database and is available for other users and property managers to see.
* The user has received an email confirming that their request was received.
* The property manager (if one is on record) has received an email informing them that a new service request about their property has been submitted.
* Information about any system issues has been added to the error log.

## Special Requirements

* The system correctly registers the creation of a service request in at least 99% of instances where a user attempts to create a service request.
* Website actions complete in under 5 seconds.
* Database and email system meet system-specified availability requirements.

## Extensibility

No public extension points defined.

## Scenarios

A scenario is an instance or specific occurrence of a use case. This section lists key representative scenarios for this use case, listing for each scenario:

* Name – a unique and meaningful name for the scenario
* Flows Exercised – List of the use case flows exercised by the scenario
* Additional Notes – any supporting notes about the scenario or its purpose.

| Name | Flows Exercised | Additional Notes |
| --- | --- | --- |
| Create service request | 5.4 Basic Flow | When a user finds a property issue that they want to be fixed, they create a service request, which notifies the property owner that someone wants something to be fixed by them. |
| Database error | 5.5.1 Handle Database Error | If the system is unable to save a submitted form to the database, the user is alerted that the request was not submitted, and the details of the error are logged. |
| Email error | 5.5.2 Handle E-mail System Error | If after a service request is submitted, the emails to the user and the property manager fail to send, the details of the error are logged. |

# Use case specification (Cockburn template)

This section provides any additional information required to make the use case more accessible to the stakeholders and the development team.

This template is an extended version of the original template defined by Cockburn [1], in particular extended with a possibility to describe Requested Information Resources often found useful when dealing with data oriented systems. [1] Cockburn, A. Writing Effective Use Cases. ISBN-13: 9780201702255. Addison-Wesley (2001).

| **Use Case Template** | **Description** |
| --- | --- |
| Use Case Name | Create Service Request |
| Use Case ID | 3 |
| Revision and Reference | Revision = 1.1  Reference = not available |
| Use Case Diagram | Please refer to the use case diagram in Section 2.1. |
| Status | Planned |
| Priority of accomplishment (optional) | Must have. Service requests are the lifeblood of the Service product. Users must be able to submit service requests so that property managers and other users can see them, and property managers can respond to them. |
| Goal | Service request is posted to website, the user receives a confirmation email, and the property manager associated with the property receives an email alert. |
| Summary | The user fills out the service request form, including fields such as the location, description of the issue in need of repair, and a picture of the issue. |
| Category | GUI / user input. The information about a service request will be input into a form, which is part of the website’s graphical user interface. The information from the form will be saved to the database. |
| Actor | The actors involved are users, property owners, the administrator, the database system, and the email system. |
| Primary Actor (initiates) | User. |
| Requested Information Resources  (optional) | Database and email system – manage. The database will be updated with the new service request, the user will receive a confirmation email, the property manager will receive a notification email. |
| Preconditions | The user must be logged into the system.  The website must be connected to the system’s database  The email system must be functioning normally |
| Triggers (optional) | The user clicks on “Create Service Request”. |
| Main success scenario | 1. User clicks “Create Service Request”.  2. User enters description, enters location, uploads photo, and submits the form.  3. Database entry is created for the service request.  4. The user receives a confirmation email.  5. The property manager receives an email alert stating that a service request has been submitted for one of their properties.  6. The service request is viewable to the public on the main page.  7. Use case ends. |
| Extensions | 3a. Database error. User is notified and error details are logged.  4a. Email system error (user). The error details are logged.  5a. Email system error (property manager. The error details are logged. |
| Alternative paths (optional) | Not applicable. |
| Post conditions | The service request is viewable on the main page.  The user has been alerted via email that their submission was received.  The property manager has been alerted via email of the new request. |
| Non-functional requirements | The system correctly registers the creation of a service request in at least 99% of instances where a user attempts to create a service request.  Website actions complete in under 5 seconds.  Database and email system meet system-specified availability requirements. |
| Validation statement | After a new service request is created, the new service request is viewable on the main page. Emails have been sent to the user and property manager. |
| Notes | The error log will be accessible to the administrator, who will be responsible for monitoring the error log. |
| Author and date | Gavin Wolf. Updated 11/5/16. |