Service Project

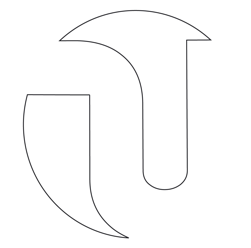
Create Service Request Use-Case

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. Therefore, the step of creating a service request is a critical step in the Service application.*** |

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 | 0.4 | Gavin Wolf | Work on incomplete sections. |
|  |  |  |  |
|  |  |  |  |



An Essential Unified Process Document

*Table of Contents*

[1 Introduction 4](#_Toc465781040)

[1.1 Document Purpose 4](#_Toc465781041)

[1.2 Document Scope 4](#_Toc465781042)

[1.3 Document Overview 4](#_Toc465781043)

[2 Use-Case Diagrams 5](#_Toc465781044)

[2.1 <Diagram Name> 5](#_Toc465781045)

[3 Actor Catalog 5](#_Toc465781046)

[4 Use-Case Catalog 6](#_Toc465781047)

[5 Use case Specification 7](#_Toc465781048)

[5.1 Introduction 7](#_Toc465781049)

[5.1.1 Document Purpose 7](#_Toc465781050)

[5.1.2 Document Scope 7](#_Toc465781051)

[5.1.3 Document Overview 7](#_Toc465781052)

[5.2 Brief Description 9](#_Toc465781053)

[5.3 Preconditions 9](#_Toc465781054)

[5.4 Basic Flow 9](#_Toc465781055)

[5.5 Alternative Flows 9](#_Toc465781056)

[5.5.1 Handle Database Error 9](#_Toc465781057)

[5.5.2 Handle Email System Error 10](#_Toc465781058)

[5.5.3 [Issues like photo size is too big?] 10](#_Toc465781059)

[5.6 Post-Conditions 10](#_Toc465781060)

[5.7 Special Requirements 11](#_Toc465781061)

[5.8 Extensibility 11](#_Toc465781062)

[5.9 Scenarios 11](#_Toc465781063)

[6 Example of use case specification 12](#_Toc465781064)

# 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:

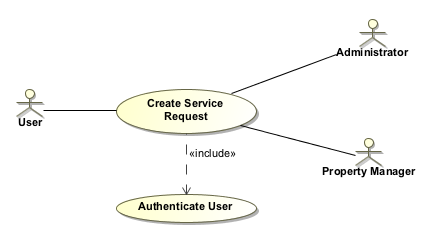
* **Brief Description** – reminder of why the use case is needed
* **Create Service Request Use Case Overview** – overview of the Create Service Request use-case in the form of a use-case diagram with supporting explanatory text
* **Actor Catalog** – catalog of all system actors
* **Use Case Catalog** – 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

The user creates a service request and the property manager for the related property is alerted of the new service request. [If there are any issues in the process, the administrator is notified.] [need to mention include authenticate user?]

<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 created an account with Service. |
| Database system | Stores service requests and user information. |
| Email system | Sends an email to the user confirming that their service request was received. |
| Property manager | Is alerted when a new Service Request is created for the property they manage. |
| Administrator | Is alerted to any issues in the process of a new service request being 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 |
| --- | --- | --- | --- |
| [xx] | Create service request | Base | A user creates a service request, which can then be viewed by other users, including property managers, who can then work on fulfilling the service request. |

# 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 and visible to other users.

## 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** the option in the navigation bar to create a service request.

**{Populate Service Request Form Fields}**

1. The system displays a service request form, with fields including: “Location”, “Description”, “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 **system database**.

**{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 **event log** entry to record the fact that a database error occurred. The **event log** entry includes the user and information about the service request if it is available.
      3. The use case resumes the basic flow at **{Use case ends}**

### Handle Email System Error

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

1. [The system creates an **event log** entry to record the fact that an email system error occurred. The **event log** entry includes the user and information about the service request if it is available. The administrator will be able to view the event log.]
2. The use case resumes the basic flow at **{Use case ends}**

### [What other potential issues am I missing?]

## Post-Conditions

* The service request has been saved to the database and is available for other users 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 event log.

**[SUBFLOWS REQUIRED??? WHAT WOULD THEY BE IN THIS CASE?]**

## Special Requirements

The system shall correctly register the creation of a service request in at least 99% of instances where a user attempts to create a service request.

## Extensibility

No public extension points defined.

## Scenarios

[this section is not in the ATM example…]

[Are some scenarios supposed to have multiple flows]

[What am I missing?]

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 an 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. |

# Example of use case specification

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 | [xx] |
| Revision and Reference | Revision = [0.4]  Reference = not available |
| Use Case Diagram | The actors in the use case are: User, Administrator and Property Manager.  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 and the user receives a confirmation email. The property manager associated with the property listed in the service request receives an email alert. |
| Summary | The user fills out the Service Request form, including fields such as the property, the issue to be fixed, and a picture of the issue. |
| Category | [Categorisation of use cases according to overall reference architecture.] [???]  *[Context dependent]* |
| Actor | The actors involved are users, the database system, the email system, the administrator, and property owners. |
| Primary Actor (initiates) | User. |
| Requested Information Resources  (optional) | Database – 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 property name.  3. User enters property location.  4. User enters description of service issue.  5. User uploads picture of issue.  6. User clicks submit.  7. Database entry is created for the service request.  8. The user receives a confirmation email.  9. The property manager is sent an email alert that a service request has been submitted for one of their properties.  10. The service request is viewable to the public on the main page. |
| Extensions | 7a. Database error. User is notified and error details are logged.  8a. Email system error. The error details are logged.  9a. Email system error. The error details are logged.] |
| Alternative paths (optional) | [Alternate path through the main success scenario w.r.t. an identified action.]  [e.g., 4a. User can select to view report in different formats, e.g. tabular or graphical map] |
| 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 | 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 prioritizing error types and arranging an appropriate alert system. |
| Author and date | Gavin Wolf. Updated 11/1/16. |