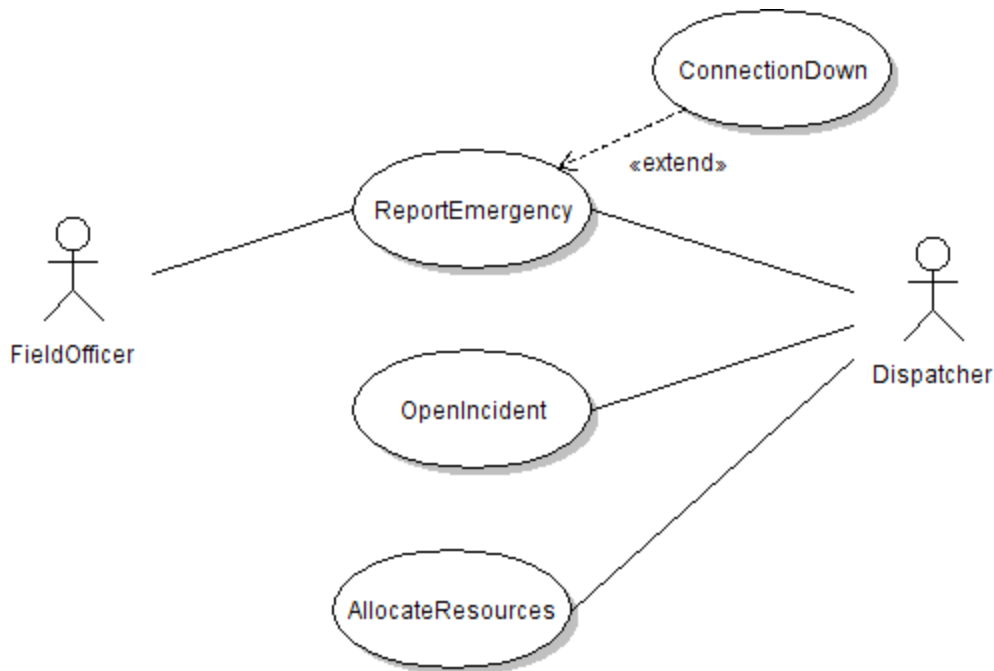


Here is an example of a use case template, based on the one used in the Bruegge & Dutoit textbook. It is less "formal" than the templates used by some use case modellers, but should be adequate for many small- to medium-scale projects.

The use case diagram for the FRIEND system:



The use case diagram clearly shows that the FieldOfficer and Dispatcher actors communicate with the ReportEmergency use case, but only the Dispatcher communicates with the OpenIncident and AllocateResources use cases.

Here is a slightly modified version of the ReportEmergency use case that is presented and refined across several chapters.

Use case name

ReportEmergency

Participating Actors

Initiated by FieldOfficer.

Communicates with Dispatcher.

Entry Condition

- The FieldOfficer is logged into FRIEND.

Flow of Events

1. The **FieldOfficer** activates the "Report Emergency" function of their terminal.
2. **FRIEND** responds by presenting a form to the **FieldOfficer**. The form includes an emergency type menu (general emergency, fire, transportation) and location, incident description, resource request and hazardous material fields.
3. The **FieldOfficer** completes the form by specifying minimally the emergency level and description fields. The **FieldOfficer** may also describe possible responses to the emergency situation and request specific resources. Once the form is completed, the **FieldOfficer** submits the form by pressing the "Submit Form" button.
4. **FRIEND** receives the form and notifies the **Dispatcher**.
5. The **Dispatcher** reviews the submitted information and creates an **Incident** in the database by invoking the **OpenIncident** use case. All the information contained in the **FieldOfficer**'s form is automatically included in the **Incident**. The **Dispatcher** selects a response by allocating resources to the **Incident** (with the **AllocateResources** use case) and acknowledges the emergency report by sending an **Acknowledgement** to the **FieldOfficer**. The **Acknowledgement** indicates to the **FieldOfficer** that the **EmergencyReport** was received, an **Incident** created, and resources allocated to the **Incident**. The **Acknowledgement** includes the resources (e.g. a fire truck) and their estimated arrival time.
6. **FRIEND** displays the **Acknowledgement** to the **FieldOfficer**.

Exit Conditions

- The **FieldOfficer** has received the **Acknowledgement** from the dispatcher, OR
- The **FieldOfficer** has received an explanation indicating why the transaction could not be processed.

Quality Requirements

- The **Field Officer**'s report is received by the **Dispatcher** within 30 seconds after it is sent.
 - The **Acknowledgement** arrives no later than 30 seconds after it is sent by the **Dispatcher**.
-

Exceptional cases are often modelled by extend relationships. The **ConnectionDown** use case extends the **ReportEmergency** use case, and handles the exceptional flow of events when the connection between the **FieldOfficer** and **Dispatcher** is broken. Note that the conditions under which this use case is initiated are described in the **ConnectionDown** use case, not the **ReportEmergency** use case.

Use case name

ConnectionDown

Participating Actors

Communicates with **FieldOfficer** and **Dispatcher**.

Entry Condition

- This use case is initiated by the system whenever the network connection between the **FieldOfficer** and **Dispatcher** is lost.
-

Flow of Events

1. The **FieldOfficer** and the **Dispatcher** are notified that the connection is broken. They are advised of the possible reasons why such an event would occur.
 2. The **FieldOfficer** and the **Dispatcher** communicate through other means and the **Dispatcher** initiates **ReportEmergency** from the **Dispatcher** station.
-

Exit Conditions

- ...
-

Quality Requirements

- ...
-