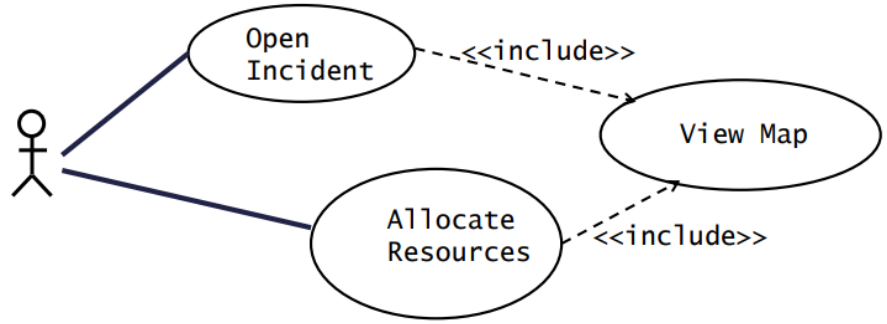
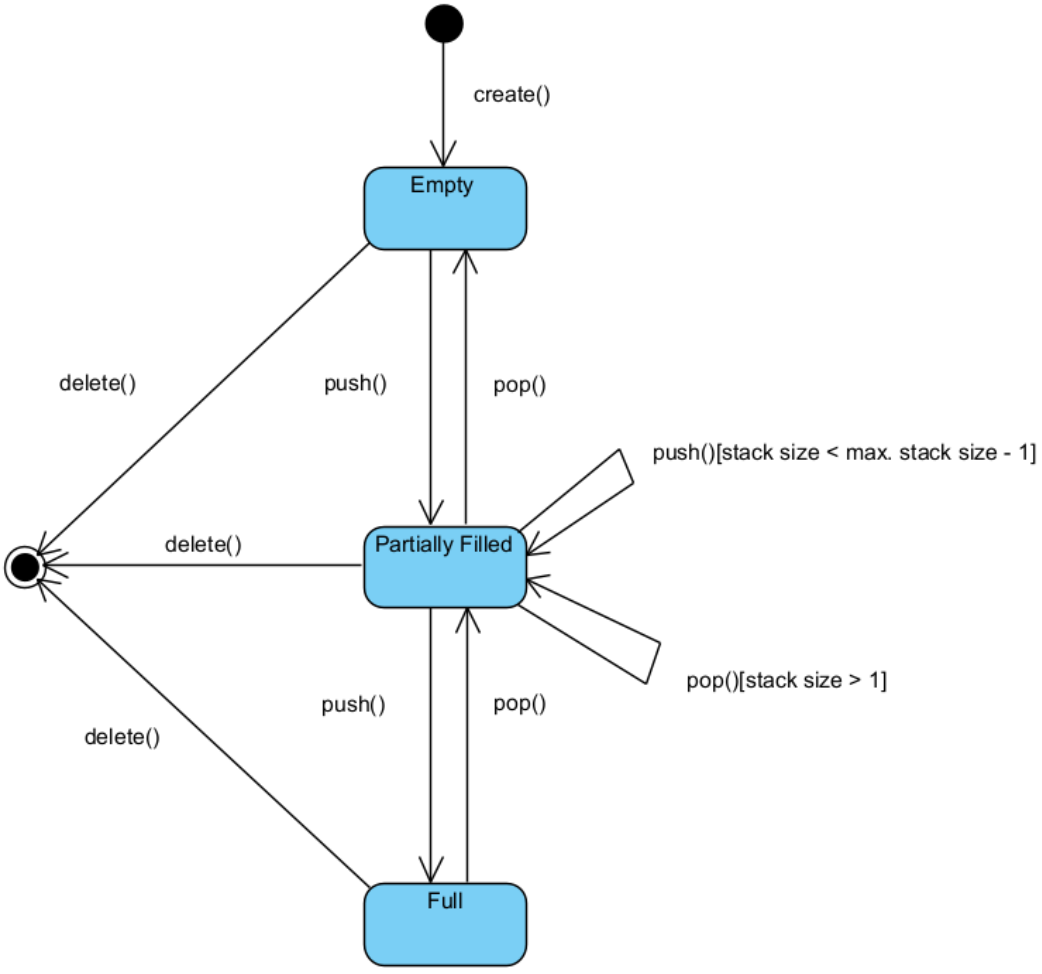
1 **(a)** No. Aside from Functional Decomposition (splitting a use-case into smaller use-cases), the *include* use-case also allows Reuse of Existing Functionality. When there are overlaps among use cases, we include the identical functionality as a use-case in each use-case that utilises the identical functionality. The included use-case can be used by use-cases that include it. This is illustrated in the example below where the "View Map" use-case is the identical functionality which is included, thus can be used by the "Open Incident" and "Allocate Resources" use-cases.

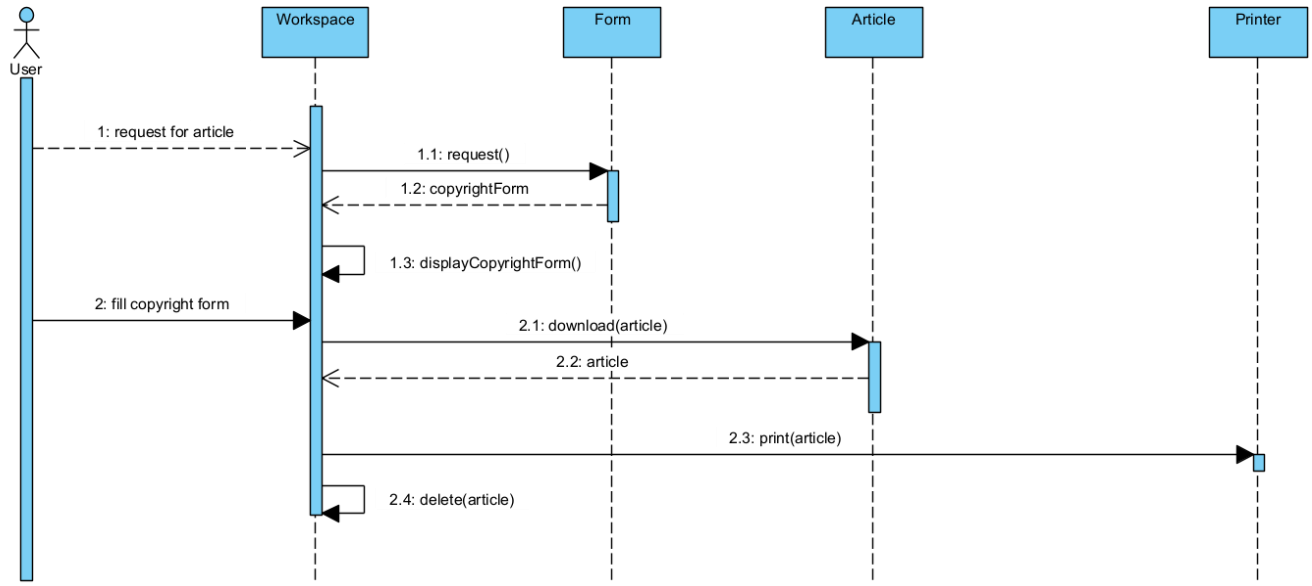


(b)



<u>Editor's note:</u> Push and Pop are not states, they are operations.

(c) Assume the user interacts with Workspace and the article requested for is valid.



(d) Dependency: CourseSchedule uses Course.

2 **(a)** Agile Methods were created to focus on code rather than design, they are based on an iterative approach to software development and are intended to deliver working software quickly and evolve quickly to meet changing requirements.
The fundamental premises of such methods are:

1. Customer involvement - customers should be closely involved throughout development to provide and prioritise new system requirements and evaluate iterations of the system.
2. Incremental delivery - the software is developed in increments; the customer specifies the requirements to be included in each increment.
3. People not process - the skills of the development team should be recognised, team members are left to develop their own ways of working without prescriptive processes.
4. Embrace change - system requirements are expected to change, the system should be designed to accommodate these changes.
5. Maintain simplicity - focus on simplicity in both the software and development process, actively eliminate complexity from the system.

(b) The Scrum sprint cycle is a fixed length of 2-4 weeks.

1. Review work to be done - the product backlog (list of work to be done on the project) is referenced to identify to-dos.
2. Select items - the team chooses features and functionality from the product backlog to be developed during the sprint (prioritise urgent and important ones).
3. Plan sprint - selected items are added into the sprint backlog.
4. Sprint - team organise themselves to develop the software, they are isolated from the customer and organisation, all communications are done through the Scrum Master who protects the developers from external distractions. (Scrum - a daily meeting held for the Scrum team to review progress and prioritise work to be done that day.)

5. Review sprint - the work completed during the sprint is reviewed and presented to stakeholders, work that is not done will be added back into the product backlog for future sprints.

(c) A good product backlog that enables dynamicity has the following properties:

1. Detailed appropriately - product backlog items that are to be worked on soon will be at the top of the backlog, small in size, and very detailed; product backlog items that will be worked on later are placed lower, can be larger, and can be less detailed.
2. Emergent - the product backlog is never complete or frozen, it is continuously updated based on a stream of economically valuable information that is constantly arriving.
3. Estimated - a product backlog item's size is estimated based on the effort required to develop it, the product owner use these estimates to determine the item's priority (and therefore position) in the product backlog.
4. Prioritised - product backlog items to be completed in the next few sprints are prioritised.

(d) $$Velocity = \frac{128\ points}{8\ sprints} = 16$$

$$Cost = 8\ sprints * 8\ Scrum\ team\ members * \$4000 = \$256000$$

Solver: Lim Her Huey (hlim062@e.ntu.edu.sg)