

Contract CO1: Create Task

Operation: createTask(title, description, priority, dueDate)

Cross References: Use Cases: Create Task, Record Activity.

Preconditions:

- A TaskCatalog exists and is associated with the SystemController.

Postconditions:

- A Task instance t was created.
 - t.taskId, t.title, t.description, t.priority, and t.dueDate were initialized from the parameters.
 - t.status was initialized to open.
 - t.creationDate was set to the current system date.
 - t was added to the TaskCatalog.
 - An ActivityEntry instance ae was created with a description detailing the change.
 - ae was associated with t and added to the ActivityLog
-

Contract CO2: Create Project

Operation: createProject(name, description)

Cross References: Use Cases: Manage Projects.

Preconditions:

- A ProjectCatalog exists and is associated with the SystemController.

Postconditions:

- A Project instance p was created.
 - p.projectId, p.name, and p.description were initialized from the parameters.
 - p was added to the ProjectCatalog.
-

Contract CO3: Update Task Status

Operation: updateTaskStatus(taskId, newStatus)

Cross References: Use Cases: Track Task Status, Record Activity.

Preconditions:

- A Task instance t with the provided taskId exists in the TaskCatalog.
- newStatus is a valid member of the Status enumeration (open, completed, or cancelled).

Postconditions:

- t.status was modified to newStatus.
 - An ActivityEntry instance ae was created with a description detailing the change.
 - ae was associated with t and added to the ActivityLog.
-

Contract CO4: Assign Task to Project

Operation: assignTaskToProject(taskId, projectId)

Cross References: Use Cases: Organize Tasks into Projects.

Preconditions:

- A Task instance t with the provided taskId exists in the TaskCatalog.
- A Project instance p with the provided projectId exists in the ProjectCatalog.

Postconditions:

- An aggregation association was formed between p and t.
 - Any existing project association for t was removed (satisfying the "at most one project" constraint).
 - An ActivityEntry instance ae was created with a description detailing the change.
 - ae was associated with t and added to the ActivityLog.
-

Contract CO5: Remove Task From Project

Operation: removeTaskFromProject(taskId)

Cross References: Use Cases: Organize Tasks into Projects.

Preconditions:

- A Task instance t with the provided taskId exists.
- t is currently associated with a Project.

Postconditions:

- The aggregation association between t and its Project was broken.
 - An ActivityEntry instance ae was created with a description detailing the change.
 - ae was associated with t and added to the ActivityLog.
-

Contract CO6: Create Subtask

Operation: createSubtask(parentTaskId, title)

Cross References: Use Cases: Manage Subtasks.

Preconditions:

- A parent Task instance t with the provided parentTaskId exists in the TaskCatalog.

Postconditions:

- A Subtask instance st was created.
 - st.title was initialized from the parameter and st.status was set to open.
 - st was added to the subtasks collection of t (composition association).
 - An ActivityEntry instance ae was created with a description detailing the change.
 - ae was associated with t and added to the ActivityLog
-

Contract CO7: Add Tag to Task

Operation: addTagToTask(taskId, tagKeyword)

Cross References: Use Cases: Manage Tags.

Preconditions:

- A Task instance t with the provided taskId exists.

Postconditions:

- A Tag instance tg with the provided tagKeyword was identified (or created if it did not already exist).
 - An association was formed between t and tg.
 - An ActivityEntry instance ae was created with a description detailing the change.
 - ae was associated with t and added to the ActivityLog
-

Contract CO8: Search Tasks

Operation: searchTasks(keyword)

Cross References: Use Cases: Search Tasks.

Preconditions: None.

Postconditions:

- The system identified all Task instances in the TaskCatalog where title or description contained keyword.
 - The list of matching tasks was returned to the user.
-

Contract CO9: View Tasks

Operation: viewTasks(criteria)

Cross References: Use Cases: View Tasks.

Preconditions:

- criteria represents a valid filter (e.g., specific date, priority, or project ID).

Postconditions:

- The system filtered the TaskCatalog based on the specified criteria.
- The resulting collection of Task instances was returned to the user.

Contract CO10: View Task History

Operation: viewTaskHistory(taskId)

Cross References: Use Cases: View Activity History.

Preconditions:

- A Task instance t with the provided taskId exists.

Postconditions:

- All ActivityEntry instances associated with t were retrieved from the ActivityLog.
 - The entries were returned to the user in chronological order.
-

Contract CO11: Create Activity Entry

Operation: createActivityEntry(taskId, description)

Cross References: All critical use cases involving updates or creation.

Preconditions:

- A Task instance t with the provided taskId exists.

Postconditions:

- An ActivityEntry instance ae was created.
- ae.timestamp was set to the current system time and ae.description was initialized.
- ae was added to the global ActivityLog.
- ae was associated with t.