

Unified Process Document Version Control System

Group 4

**Henrik Bäck
Mathias Andersson**

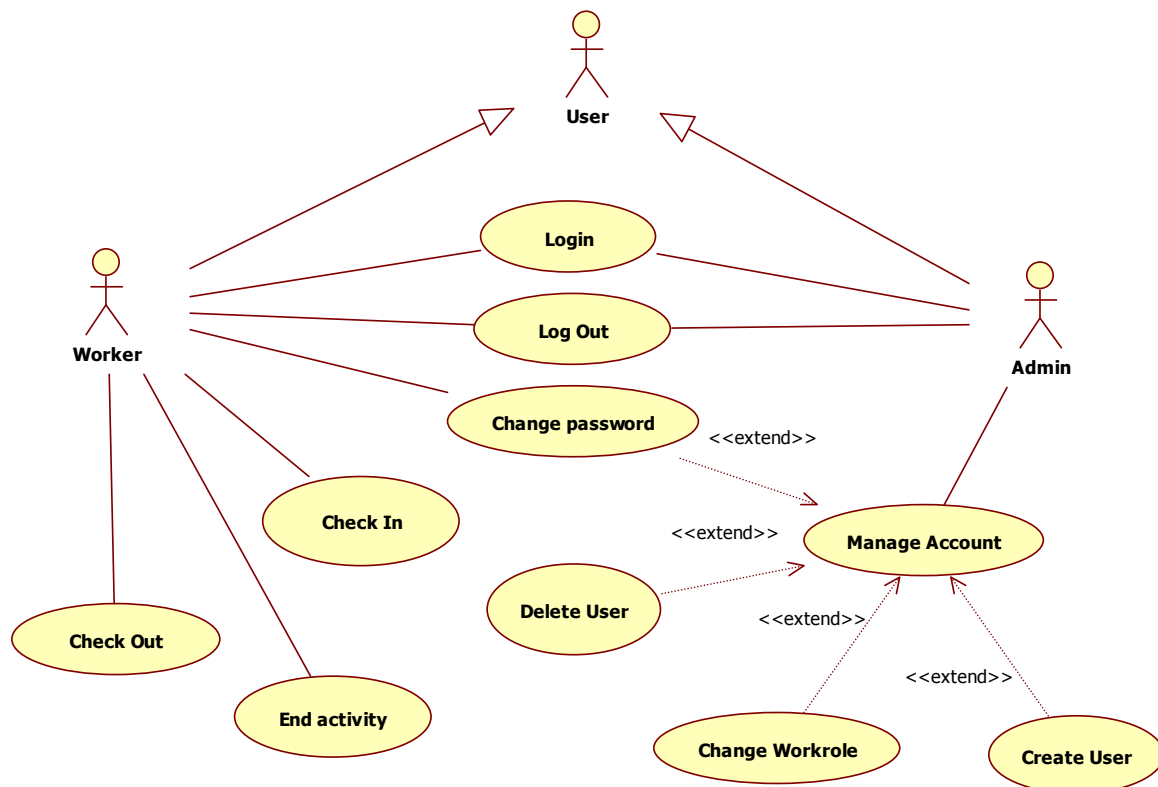
12/09/2006
University of Karlstad

Page of Contents

Page of Contents.....	2
1. Use-Case Model ver. 1.1.....	3
1.1. Use Cases	3
1.1.5 Use-Case "Check In"	3
Brief Description.....	3
Flow of events	5
Preconditions	5
Post conditions.....	6
Special requirements.....	6
Extension points	6
1.1.7 Use-Case "End activity"	6
Brief Description.....	6
Flow of events	7
Preconditions	7
Post conditions.....	7
Special requirements.....	7
Extension points	7
2 Analysis Model ver. 1.2.....	8
2.1. Analysis Class	8
2.2. Use-Case Realizations	9
2.2.9 Use-Case "Check In, First Time"	9
2.2.10 Use-Case "Check In, Update artefact – No conflict"	10
2.2.11 Use-Case "Check In, Update artefact – Conflict"	11
2.2.10 Use-Case "End activity"	12
3. Design Model ver. 1.1.....	13
3.1. Deployment Model	13
3.2. Class Model	14
3.2.1 Server Package	14
3.3 Use-Case Realisation	18
3.3.1 Check in artifact, First time, Successful execution.	19
3.3.3 End Workflow, Successful.....	19
3.3.3 End Workflow, Successful.....	20

1. Use-Case Model ver. 1.1

1.1. Use Cases



1.1.5 Use-Case “Check In”

Brief Description

This Use-case describes the possibility for a worker to check in an artefact to system. The worker can only check in an artefact related to his assigned work role and current workflow.

A **Architect** may check in

Use-case model during the Requirements Workflow.
Architectural description during the Requirements Workflow.
Analysis-model during the Analysis Workflow.
Design-model during the Design Workflow.
Implementation-model during the Implementation workflow.

A **System analyst** may check in

Domain-model during the Requirements Workflow.

Business model during the Requirements Workflow.
Use-Case model during the Requirements Workflow.
Glossary during the Requirements Workflow.

A **Test Engineer** may check in Test-model during the Test Workflow.

A **Integration tester** may check in Test-model during the Test Workflow.

A **Component Engineer** may check in Analysis-model during the Analysis Workflow.
Design-model during the Design Workflow.
Implementation-model during the Implementation Workflow.
Test-model during the Test Workflow.

A **Use-Case Engineer** may check in Use-Case model during the Analysis Workflow.
Analysis-model during the Analysis Workflow.

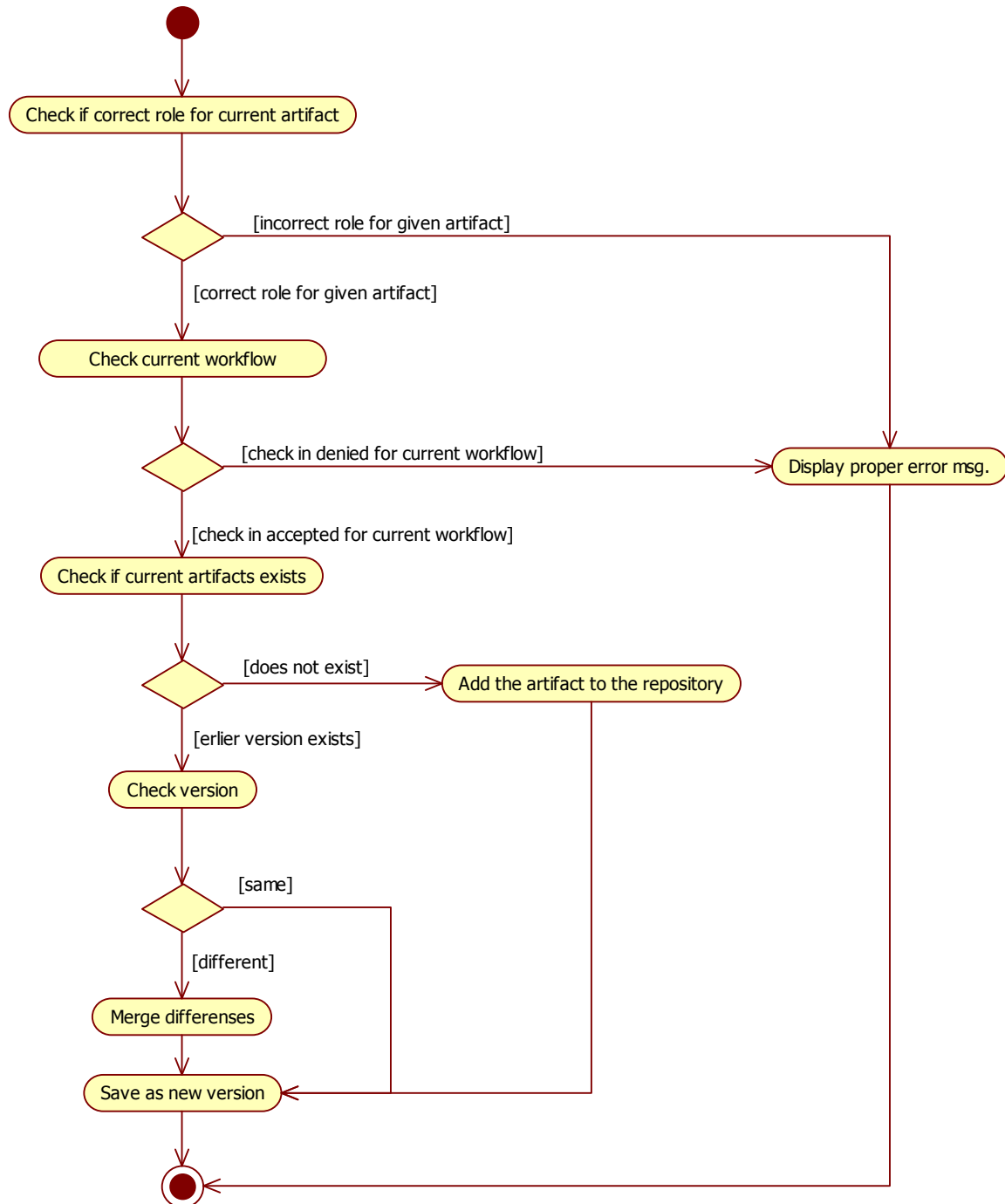
A **System tester** may check in Test-model during the Test Workflow.

A **System integrator** may check in Implementation-model during the Implementation workflow.
Test model during the Test Workflow.

A **Use-Case specifier** may check in Use-case-model during the Requirements Workflow.

A **Interface designer** may check in Use-Case-model during the Requirements Workflow.

Flow of events



Preconditions

User is logged in.

A repository exists.

Post conditions

If the worker tries to check in an artefact, in the wrong workflow or the worker do not have permission to check the specified artefact an proper error message is shown. Else the artefact has been updated in or added to the repository.

Special requirements

None

Extension points

None

1.1.7 Use-Case “End activity”

Brief Description

Gives a worker the possibility to end a Workflow and begin the next subsequent one. This will give the...

System Analyst to end the **Requirements Workflow** when he has checked that all of the artefacts, in the current workflow, are correct.

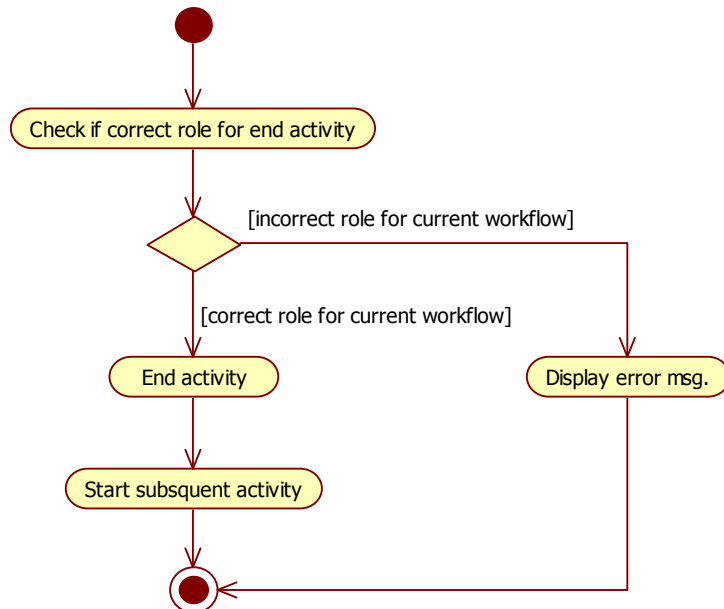
Architect to end the **Analysis Workflow** when he has checked that all of the artefacts, in the current workflow, are correct.

Architect to end the **Design Workflow** when he has checked that all of the artefacts, in the currents workflow, are correct.

Architect to end the **Implementation Workflow** when he has checked that all of the artefacts, in the current workflow, are correct.

Test Engineer to end the **Test Workflow** when he has checked that all of the artefacts, in the current workflow, are correct.

Flow of events



Preconditions

User is logged in.

Post conditions

If the worker is in the correct workflow and has the correct work role for that workflow the activity ends and the subsequent one begins. If not, an error message is shown.

Special requirements

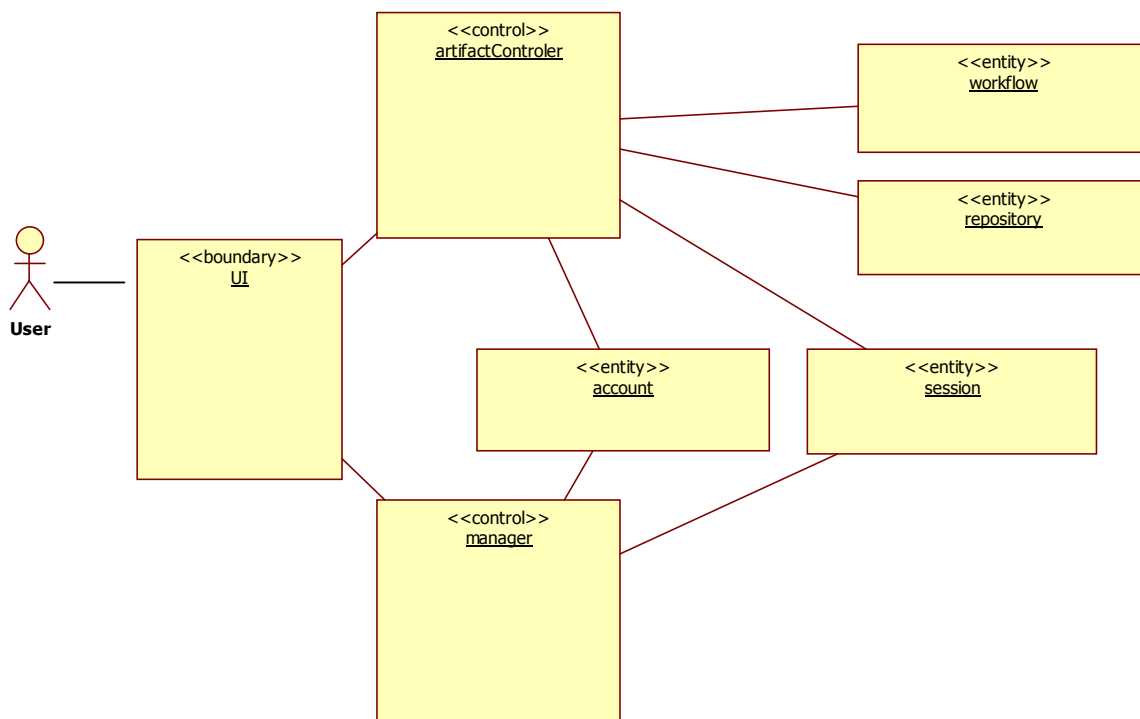
None

Extension points

None

2 Analysis Model ver. 1.2

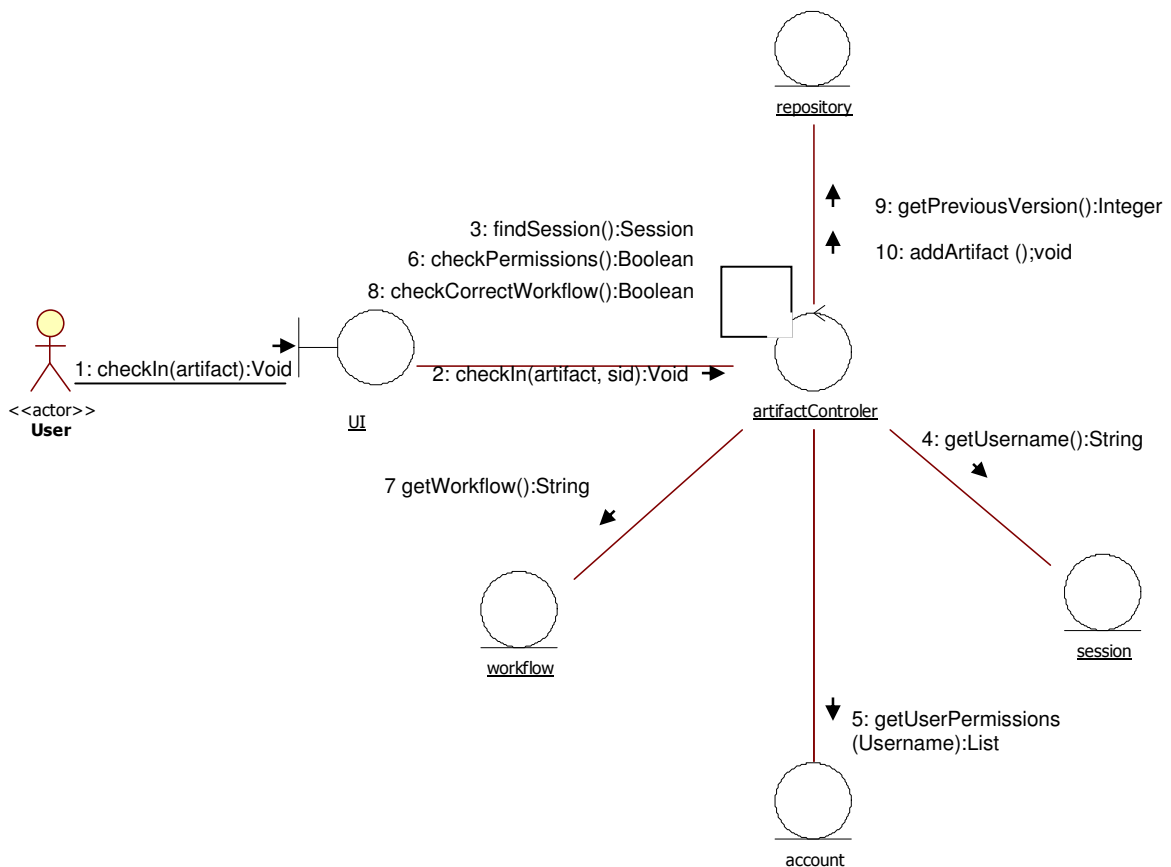
2.1. Analysis Class



2.2. Use-Case Realizations

2.2.9 Use-Case "Check In, First Time"

Interaction



Event flow

A user uploads a new artefact to the user interface (IU). The system checks if the user has the right to upload the specified artefact in the current workflow. If so, the artefact is given a version number and is added to the repository.

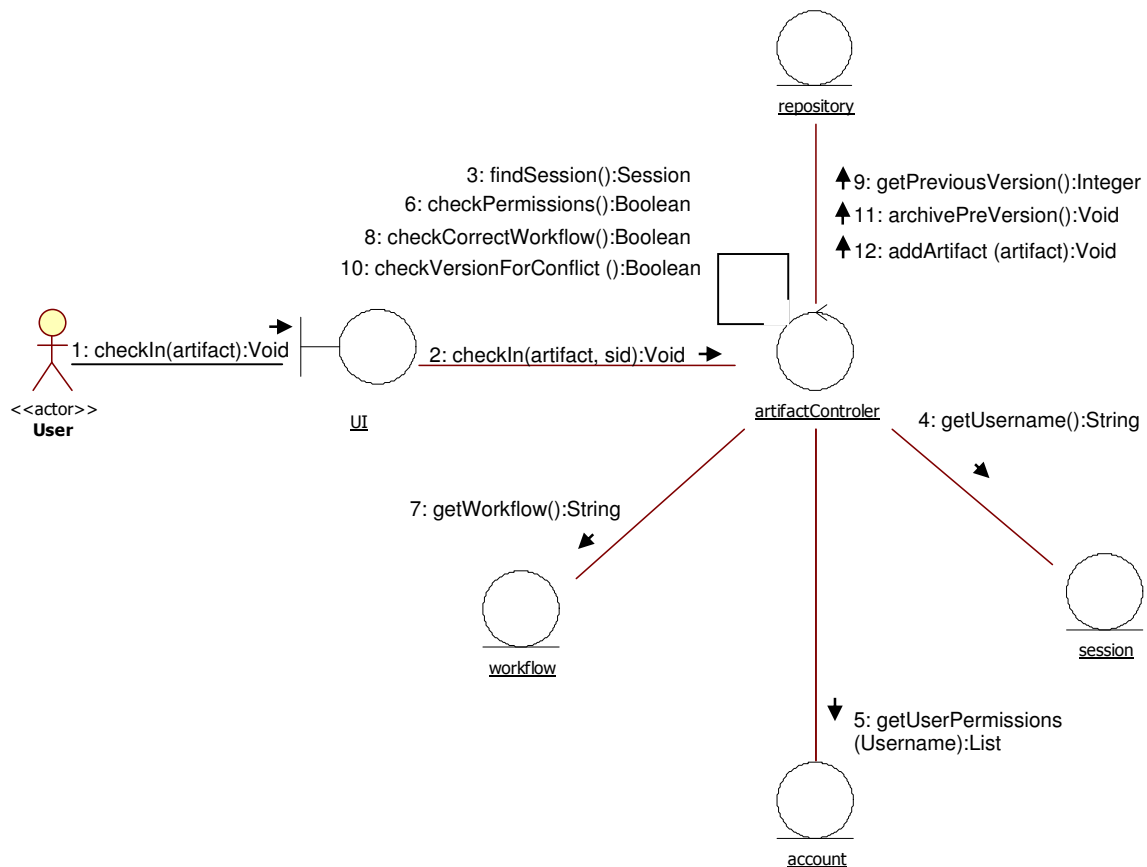
If the user does not have permission to upload the artefact in the current workflow, the user interface shows an error message.

Special requirements

None

2.2.10 Use-Case "Check In, Update artefact – No conflict"

Interaction



Event flow

A user uploads an artefact to the user interface (IU). The system checks if the user has the right to upload the specified artefact in the current workflow. If so, the system checks for version differences and if there is no conflict, the system archives the previous version and adds the uploaded artefact to the repository with a new version number.

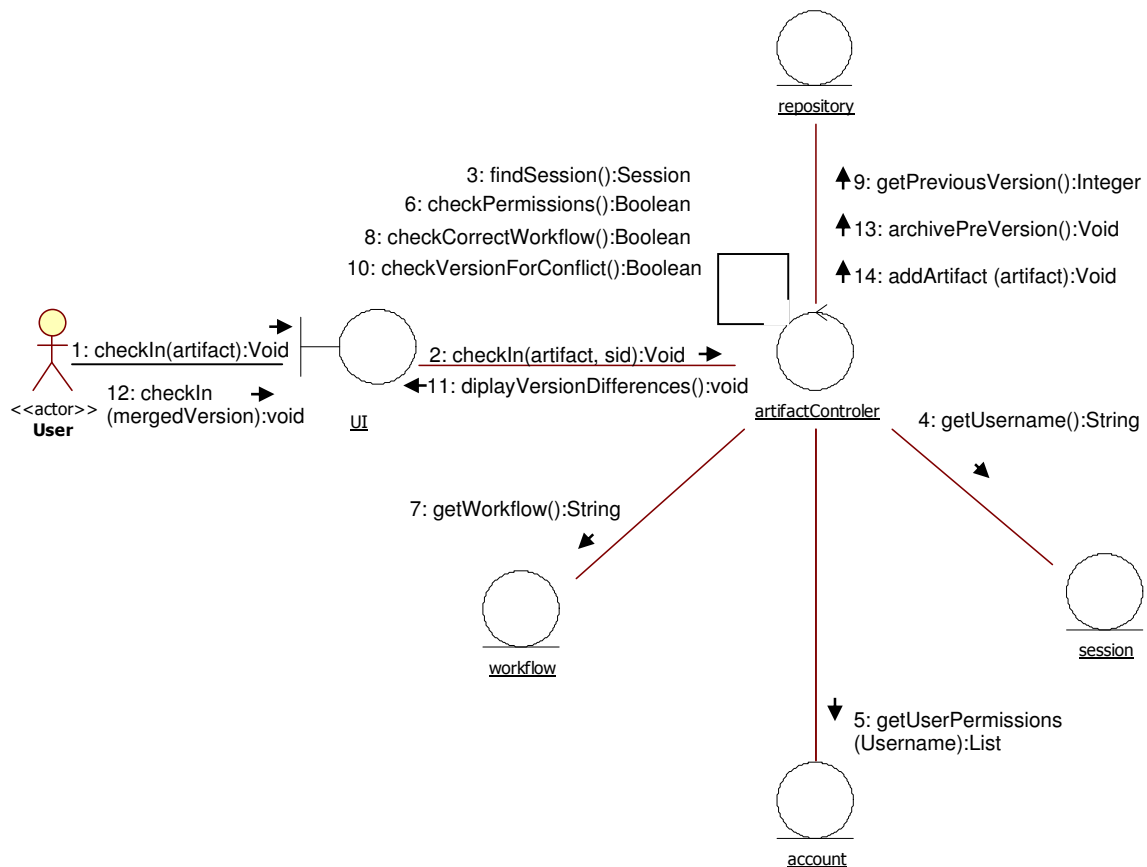
If the user does not have permission to upload the artefact in the current workflow, the user interface shows an error message.

Special requirements

None

2.2.11 Use-Case "Check In, Update artefact – Conflict"

Interaction



Event flow

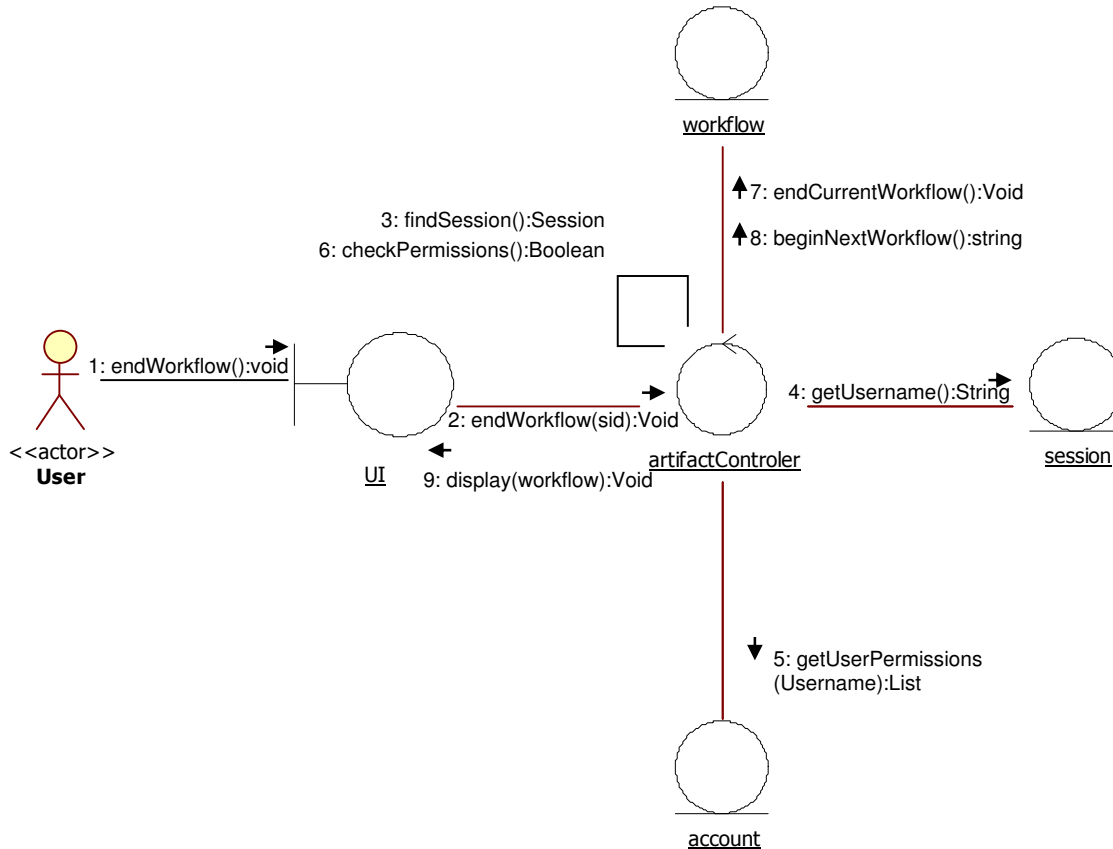
A user uploads an artefact to the user interface (IU). The system checks if the user has the right to upload the specified artefact in the current workflow. If so, the system checks for version differences and if there is no conflict, the system archives the previous version and adds the uploaded artefact to the repository with a new version number.

If the user does not have permission to upload the artefact in the current workflow, the user interface shows an error message.

Special requirements

None

2.2.10 Use-Case "End activity"



Event flow

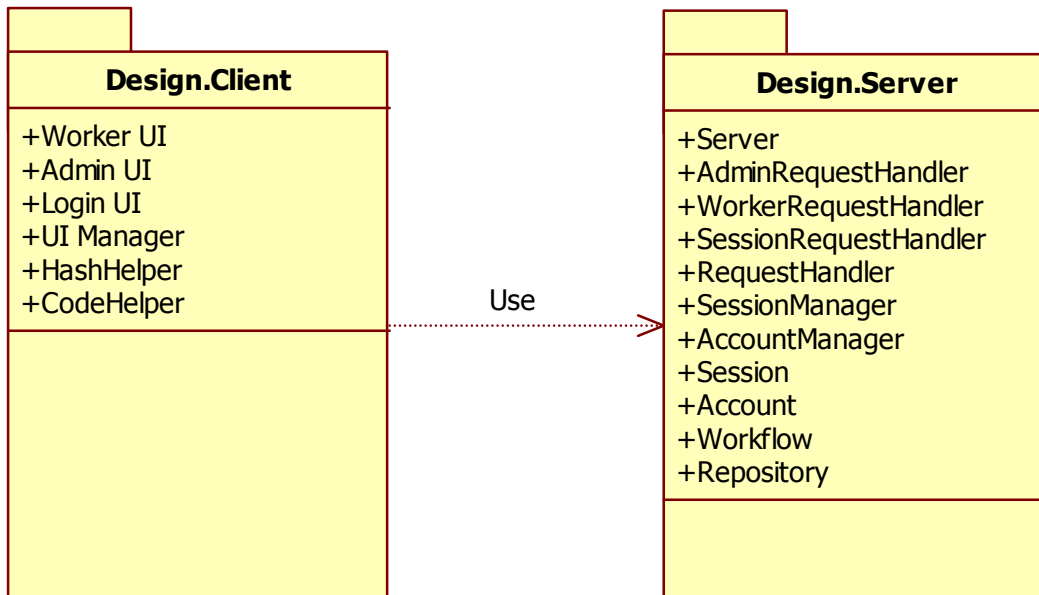
A users ask to end the current workflow. If the user has permissions to end the current workflow the systems ends the current workflow and begins the next, subsequent, one. If the user lacks permission to and the current workflow a error message is shown.

Special requirements

None

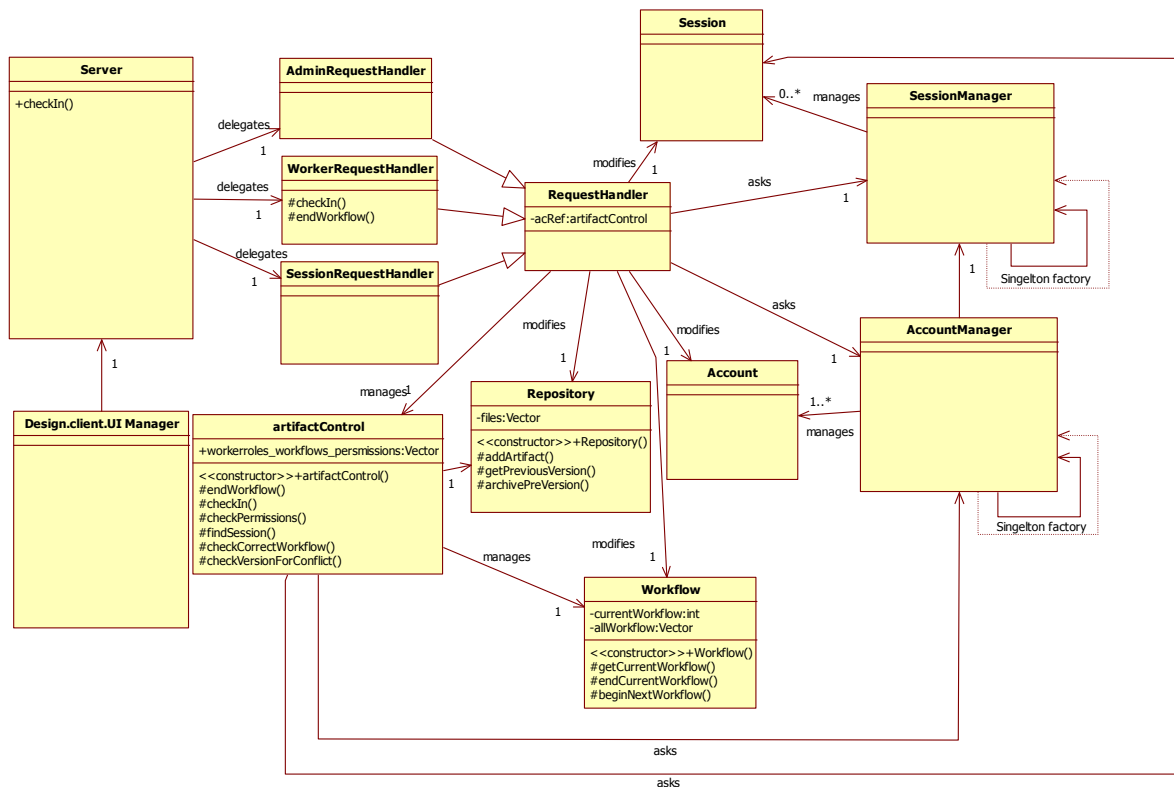
3. Design Model ver. 1.1

3.1. Deployment Model



3.2. Class Model

3.2.1 Server Package



Class WorkerRequestHandler

Package: Server

Method Detail

`checkIn`

Protected void `checkIn(artifact file)`

This enables the function to upload and checkin an artifact.

`endWorkflow`

Protected void `endWorkflow()`

This enables the function to end a workflow and begin the next one.

Class requestHandler

Package: Server

Field Detail

acRef

private artifactControl acRef

This is a reference to the artifactControl

Class Repository

Package: Server

Public class repository

This class contains the information about the repository.

Field Detail

Files

Private Vector Files

This is a vector of files and versions of all the documents in the repository

Constructor Details

repository

public repository()

Stereotype: Constructor

Method Detail

addArtifact

Protected void addArtifact(artifact file)

Adds a new version of the artifact to the repository.

getPreviousVersion

Protected int getPreviousVersion(artifact file)

This returns the version of the current file in the repository and -1 if it is not there.

archivePreVersion

Protected void archivePreVersion(artifact file)

This archives the old version and gives place for the new version

Class artifactControl

Package: Server

Public class artifactControl

This class manages the workflow and repository.

Constructor Details

artifactControls

public artifactControls()

Stereotype: Constructor

Method Detail

endWorkflow

protected void endWorkflow()

This function ends the existing workflow and begins the next one.

checkIn

protected void checkIn(artifact file)

This function checks in / adds a new file to the repository.

checkPermissions

protected bool checkPermissions(int type)

Returns true if the user has access to upload an artifact or end a workflow

findSession

protected int findSession()

Returns the sid for the current session

checkCorrectWorkflow

protected bool checkCorrectWorkflow()

Returns true if the artifact can be checked in at the current workflow.

checkVersionForConflict

protected bool checkVersionForConflict()

Returns true if there is a conflict with another version of the artifact.

Workflow

Package: Server

Public class Workflow

This class controls the current workflow.

Field Details

currentWorkflow

private int currentWorkflow

This contains the current workflow.

allWorkflow

private vector allWorkflow

This contains all of the Workflows and their names.

Constructor Details

Workflow

```
public Workflow()
```

Stereotype: Constructor

Method Detail

getCurrentWorkflow

```
protected string getCurrentWorkflow()
```

Returns the name of the current workflow

endCurrentWorkflow

```
protected void endCurrentWorkflow()
```

Ends the current workflow

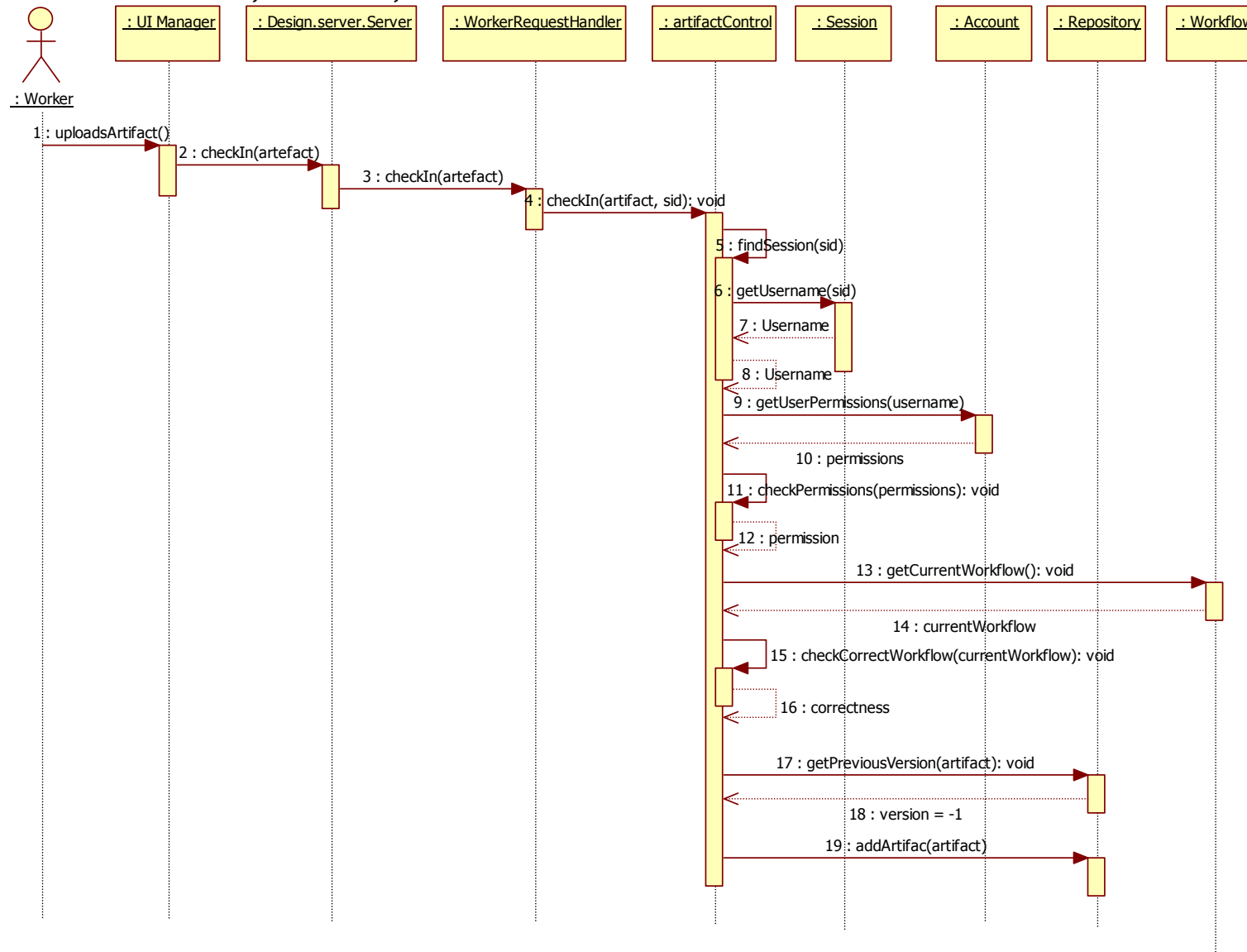
beginNextWorkflow

```
protected void beginNextWorkflow()
```

Begins the next, subsequent, workflow.

3.3 Use-Case Realisation

3.3.1 Check in artifact, First time, Successful execution.



3.3.3 End Workflow, Successful

