

SOFTWARE DESIGN DOCUMENT

Project name: xPages

Team members

- Salem Al-Zahrani
- Mousa Al-Mousa
- Abdullah Al-Nuaim
- Mouath Ba-garish
- Basim Bakhurji
- Houssam Waly



KFUPM

*Senior project
SWE417*

Contents

Deployment Architecture:.....	3
Server Side	3
Language.....	3
PHP	3
Database	3
MY SQL.....	4
Server	4
Apache	4
Platform.....	4
Linux	4
Framework	4
Code Igniter	4
Client Side.....	5
Client Browser	5
Java Script.....	5
JQuery	5
HTML 5	5
CSS 3	6
First Level Package Model	7
First Level Package Model (with Façade Classes)	8
Second Level Package Model (Final)	10
Final Class Model	11
Updated Sequence Diagrams	12
Create situation	12
Create context (copy/paste).....	12
Customize situation.....	13
Customize context.....	14
Export situation.....	15
Add situation links	15
Upload file	16
Add component.....	17
Search situation.....	18

Smart search..... 19

Create profile 20

Change personal information..... 20

Add friend..... 21

Delete friend..... 22

Log in..... 22

Log out..... 23

Activity Models 25

Database Models & Architecture..... 29

ER Model 29

EER Model..... 30

Database DDL..... 31

Situation 31

 Context..... 31

Contains 32

 Media Object..... 32

Situation Friend List 32

Situation List 33

Template..... 33

User Profile..... 33

Deployment Architecture:

NOTE: The main reason for choosing only open source products is because we are preparing for an open source competition which requires only an open source project.

Server Side

Language

PHP

- PHP can run on both UNIX and Windows servers, which makes it more accessible than Windows (ASP).
- PHP is growing day by day.
- PHP is a particularly useful programming language because it allows for advanced programming and is easy to integrate with web pages.
- PHP interfaces very well with MySQL.
- PHP is an open source server side programming language, which is very easy to get from the market, and the coding style of PHP language is very understandable.
- known as server independent object oriented programming language which increases the flexibility of PHP towards its users
- PHP is more powerful than ASP and JSP and runs on all kinds of operating systems and web browsers.
- It's available with documentation in many languages.
- Programmers of Java, PERL, BASIC, and other popular languages can find many parallels to ease transition to PHP
- It has a simple syntax that is easy to parse. It is also a very stable language with powerful problem solving features.
- This language can be easily embedded into page HTML so there isn't any requirement of separate coding in PHP.
- It doesn't consume many system resources and doesn't slow other processes which are trying to run like other languages.
- the community associated with PHP might be its strongest asset

Database

MY SQL

- Open source and compatible more with PHP language
- Is the most familiar DBMS for the team members
- large community support

Server

Apache

- The Apache Web Server is the Internet's most popular Web server,
- Free license and Open source
- Active support and development.
- portable(multiple platform)
- Most web-based companies uses Apache
- best server for php

Platform

Linux

- Apache is very reliable and powerful with Linux
- More secure
- Open source
- Very reliable and strong

Framework

Code Igniter

- more familiar with,
- well documented framework,
- large support community
- Code Igniter is powerful, high performance, open Source PHP framework that help to write MVC PHP applications rapidly. Code Igniter is known for having a light footprint, thereby reducing server's work.

Client Side

Client Browser

based on our technologies, these are the browsers that are supported by our system

- FireFox v3.5 and above
- IE9
- Chrome
- Safari 4 and above

Java Script

- multi-paradigm (including object oriented) programming language on client-side
- widely used than other client-side scripting languages like: VBScript
- Supported almost by all browsers

JQuery

- JavaScript library that simplify client-side scripting.
- Has plugging that support drag and drop events.
- Light weight library . It is 77 KB in production mode.

HTML 5

HTML 5 vs. HTML 4	
HTML 5	HTML 4
Is being developed by web hypertext application technology working group (WHATWG) and W3C HTML WG.	Was developed by World Wide Web consortium and WHATWG (web hypertext application technology working group)
New tags like <nav> website navigation block and <footer > for bottom lines in web page have been added.	Lacks rules for parsing and hence it is difficult to handle errors.
Strict parsing and flexing rules are introduced to handle any errors.	Contains some deprecated tags such as and <centre> that are removed in next version.
It includes multimedia elements (<audio> and <video>)	No multimedia support without third party software.

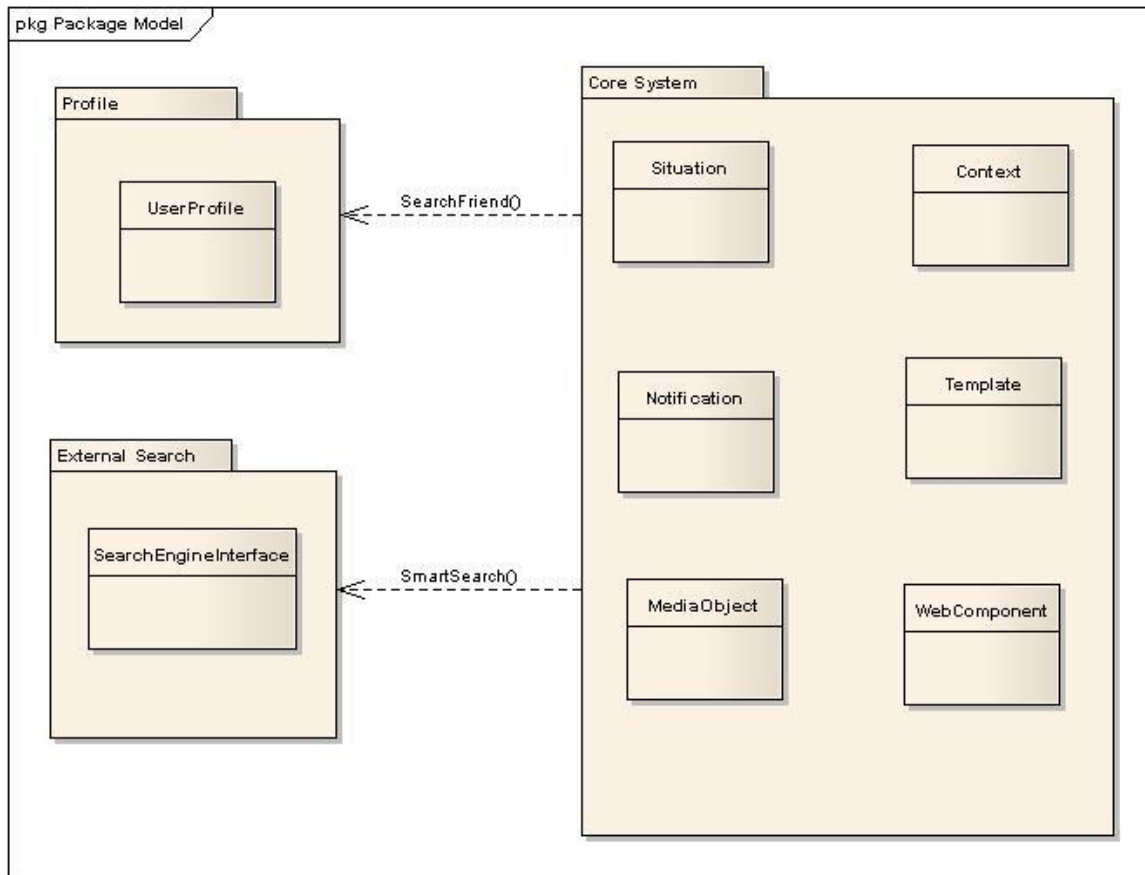
CSS 3

Cascading Style Sheets or CSS manage the whole designing of a website with one-click changes. CSS3 version differs from CSS2 in the incorporation of Modules and other changes. Modules enable the designing to be done in a lesser time with more ease while updating individual features and specifications.

Other major changes/additions include:

- New Combinator
- New CSS Selectors
- New Pseudo-elements
- New Style properties
- Modules for grid creations and template layout.
- User interface module has been updated with the feature of adding/designing many elements.
- Ruby and Paged Media module allow more support for languages, paged media and with flexibility in using style sheets by Media Queries is also present.
- CSS3 mainly gives a simple interface that can be worked by beginners as well for making better and more appealing website styles.

First Level Package Model



Core System Package: Situation. Boolean. addFriend(String)



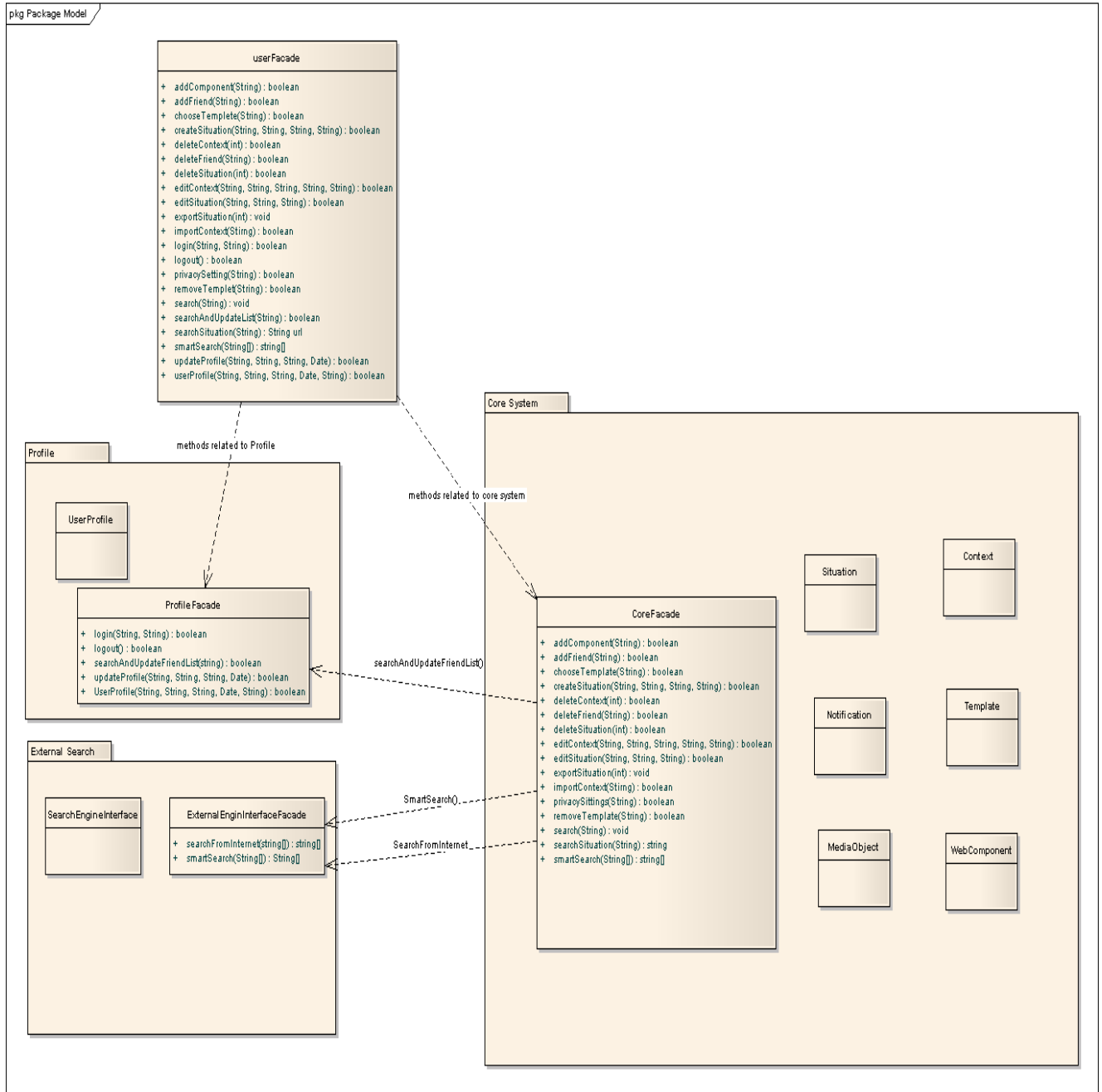
Profile Package: UserProfile.Boolean searchFriend(String)

Core System Package: Situation. String [] smartSearch(String [])

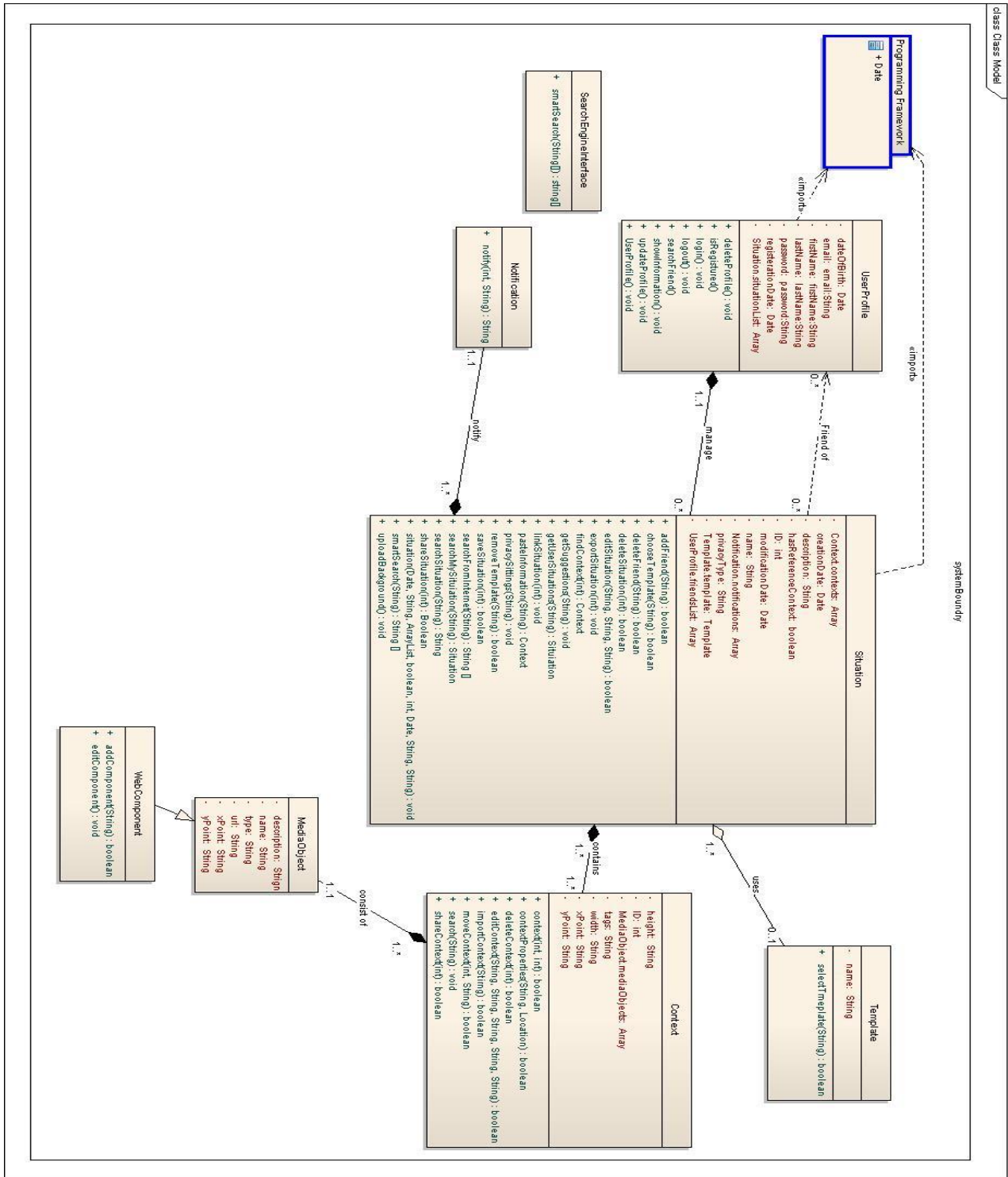


External Search Package : SearchEngineInterface. String [] smartSearch(String [])

First Level Package Model (with Façade Classes)

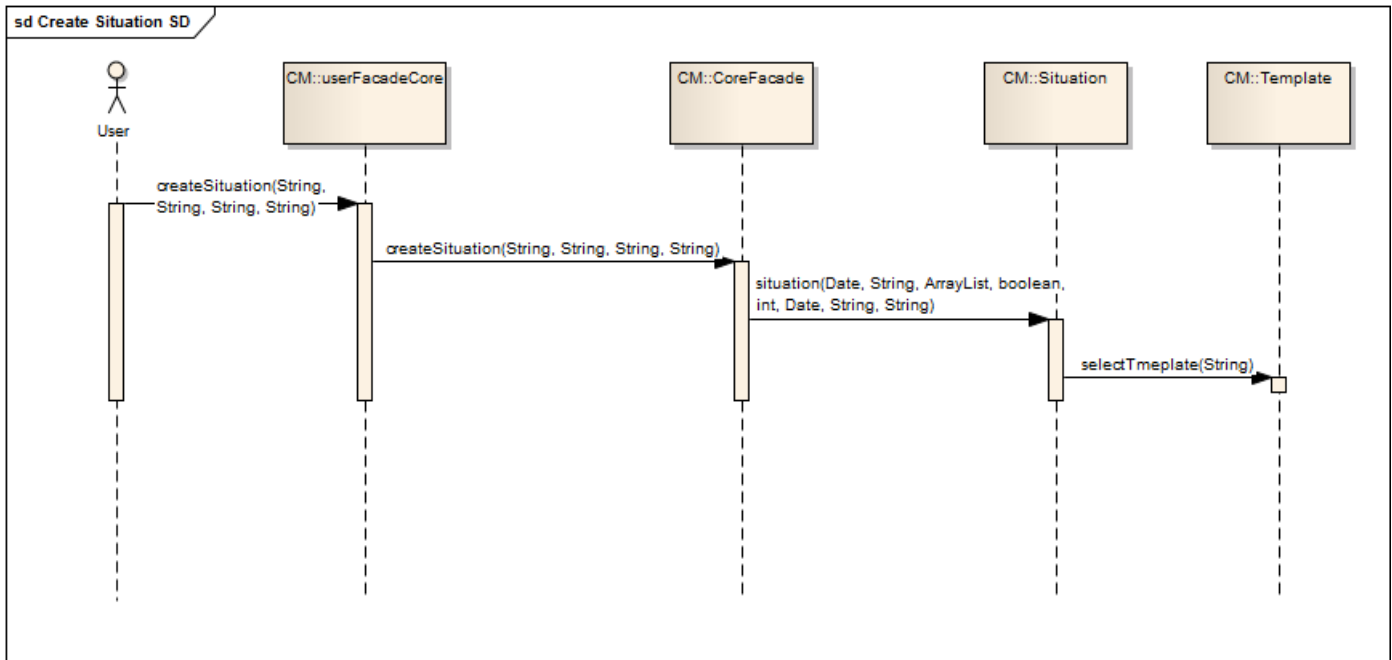


Final Class Model

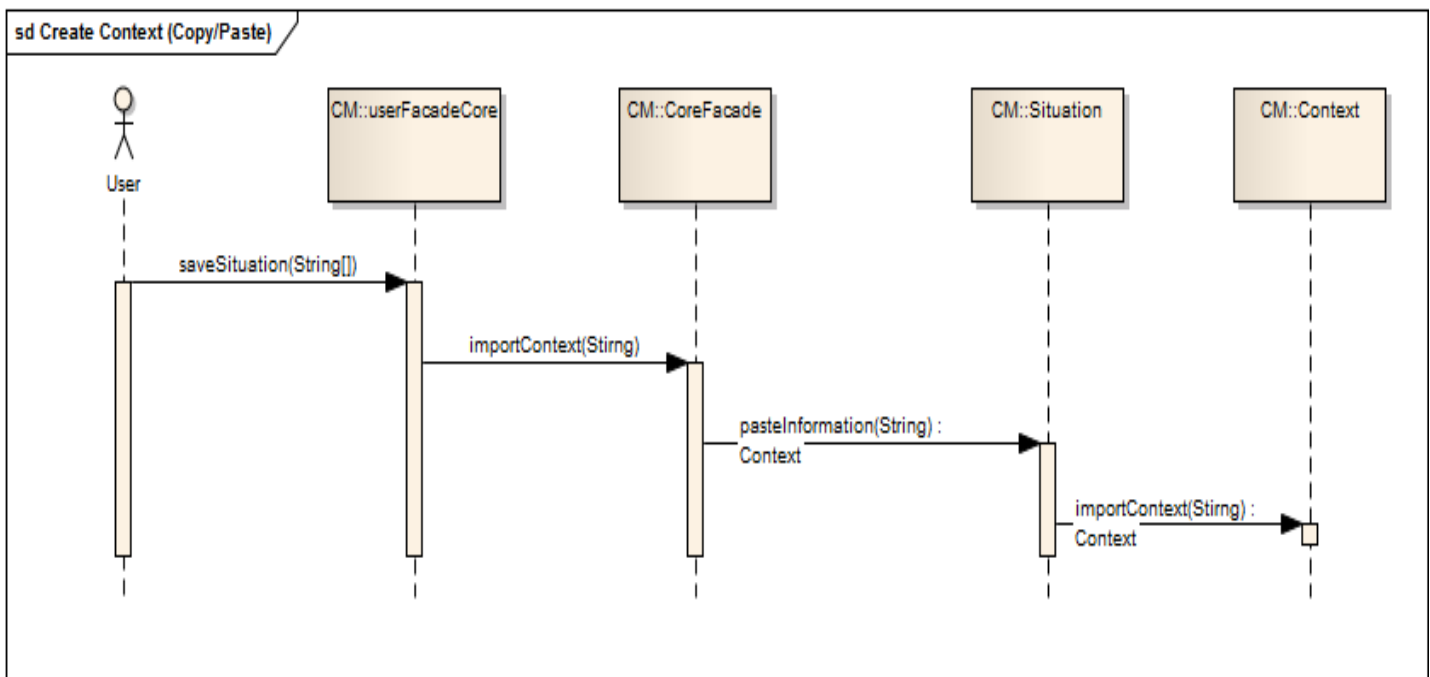


Updated Sequence Diagrams

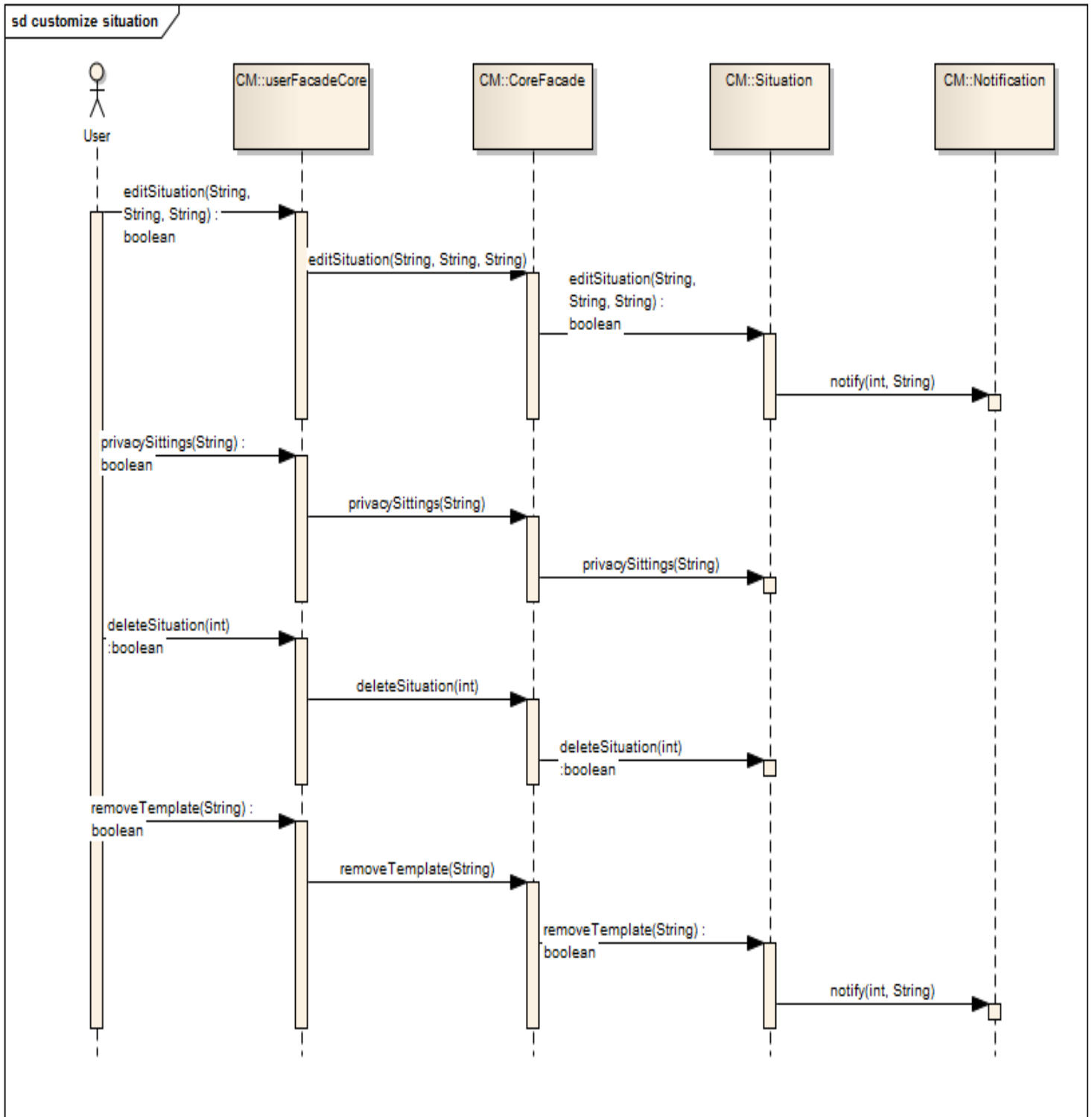
Create situation



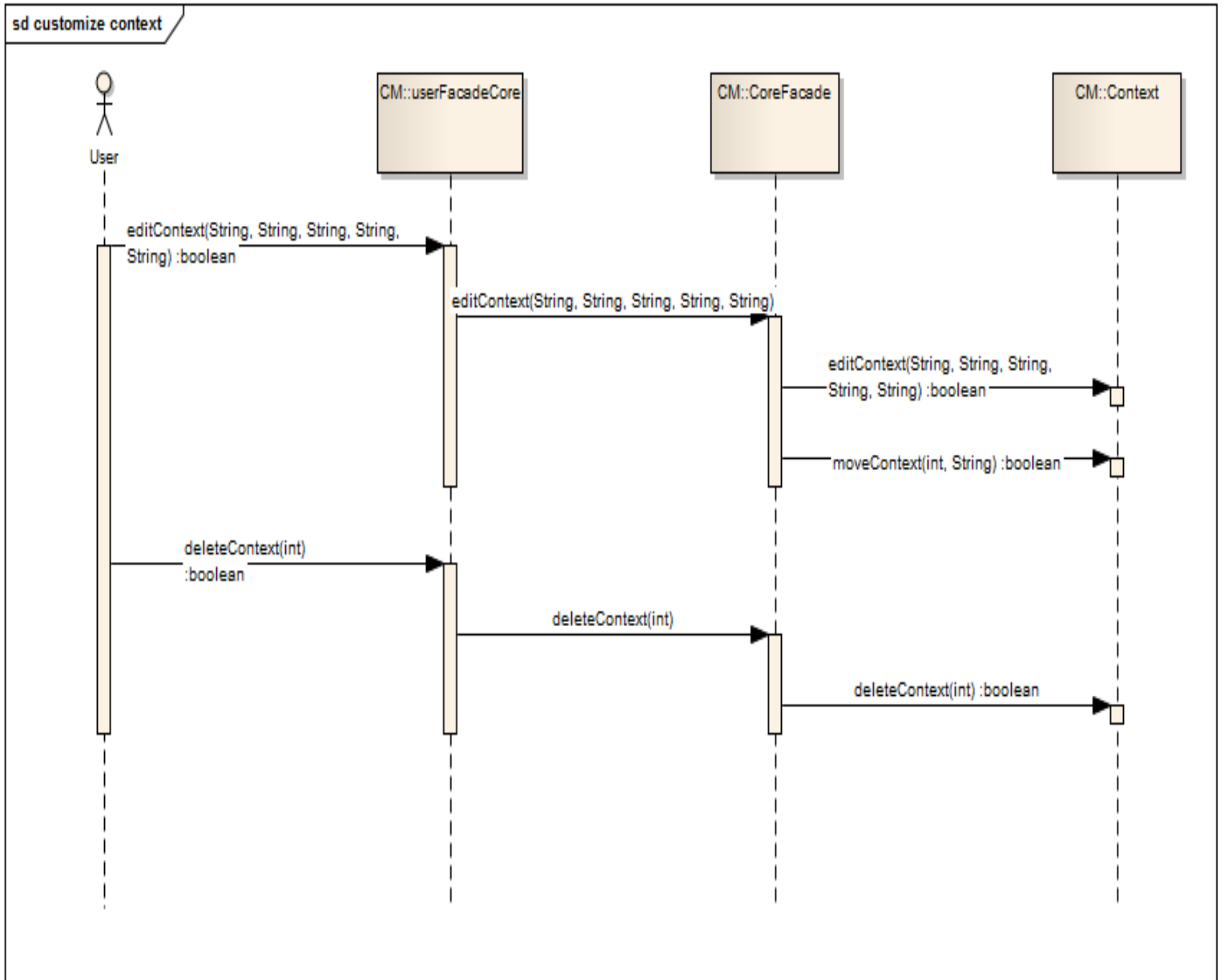
Create context (copy/paste)



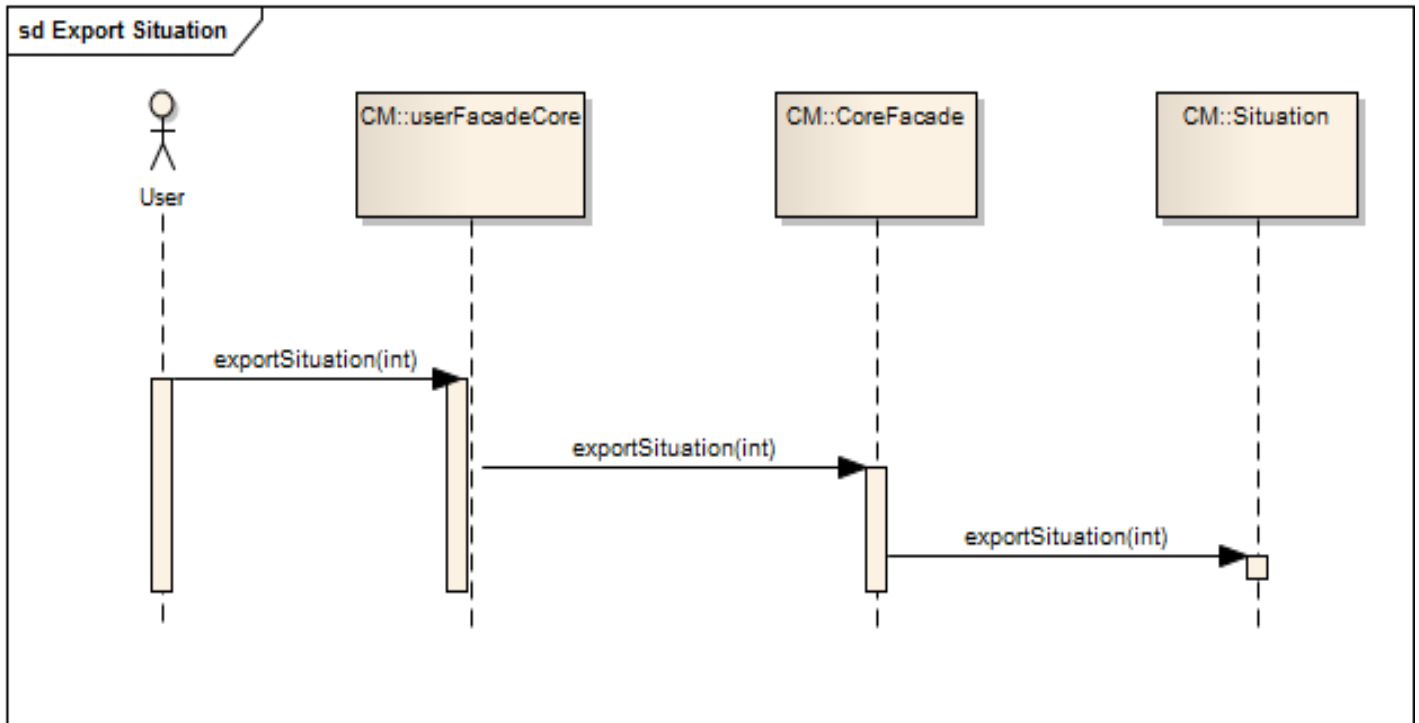
Customize situation



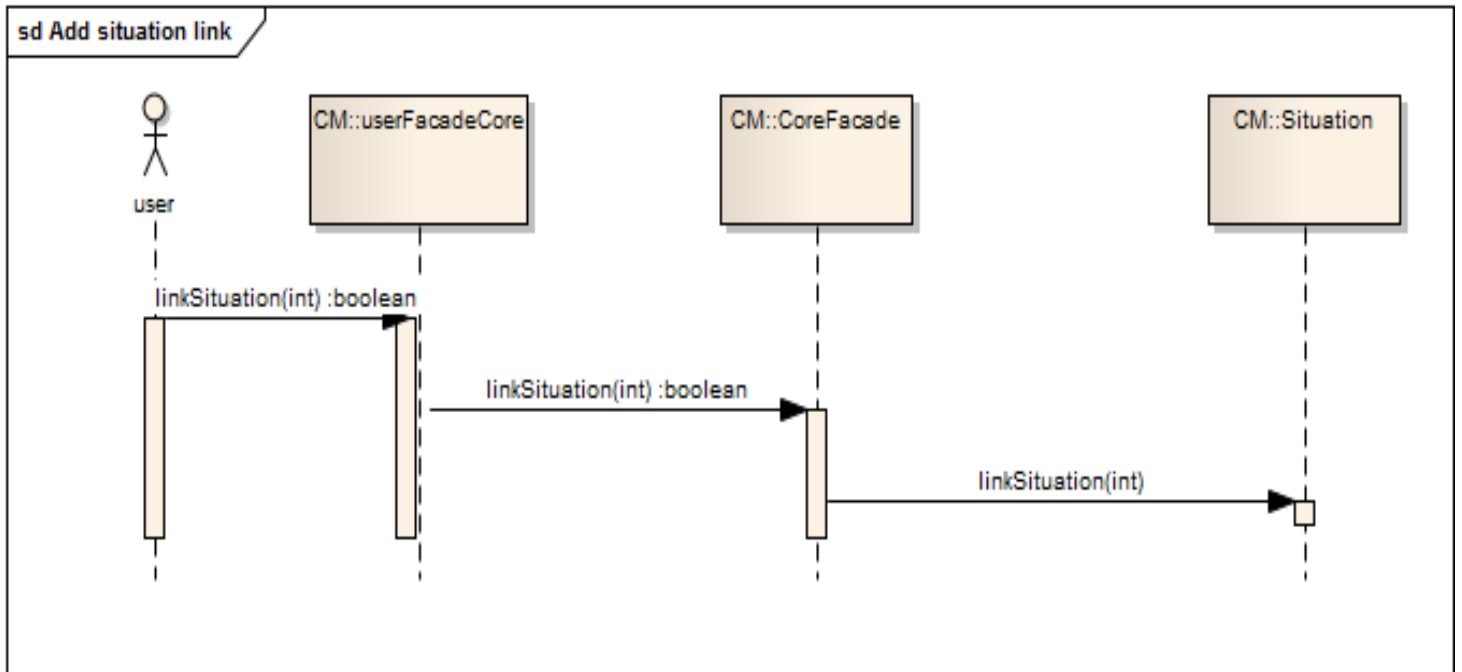
Customize context



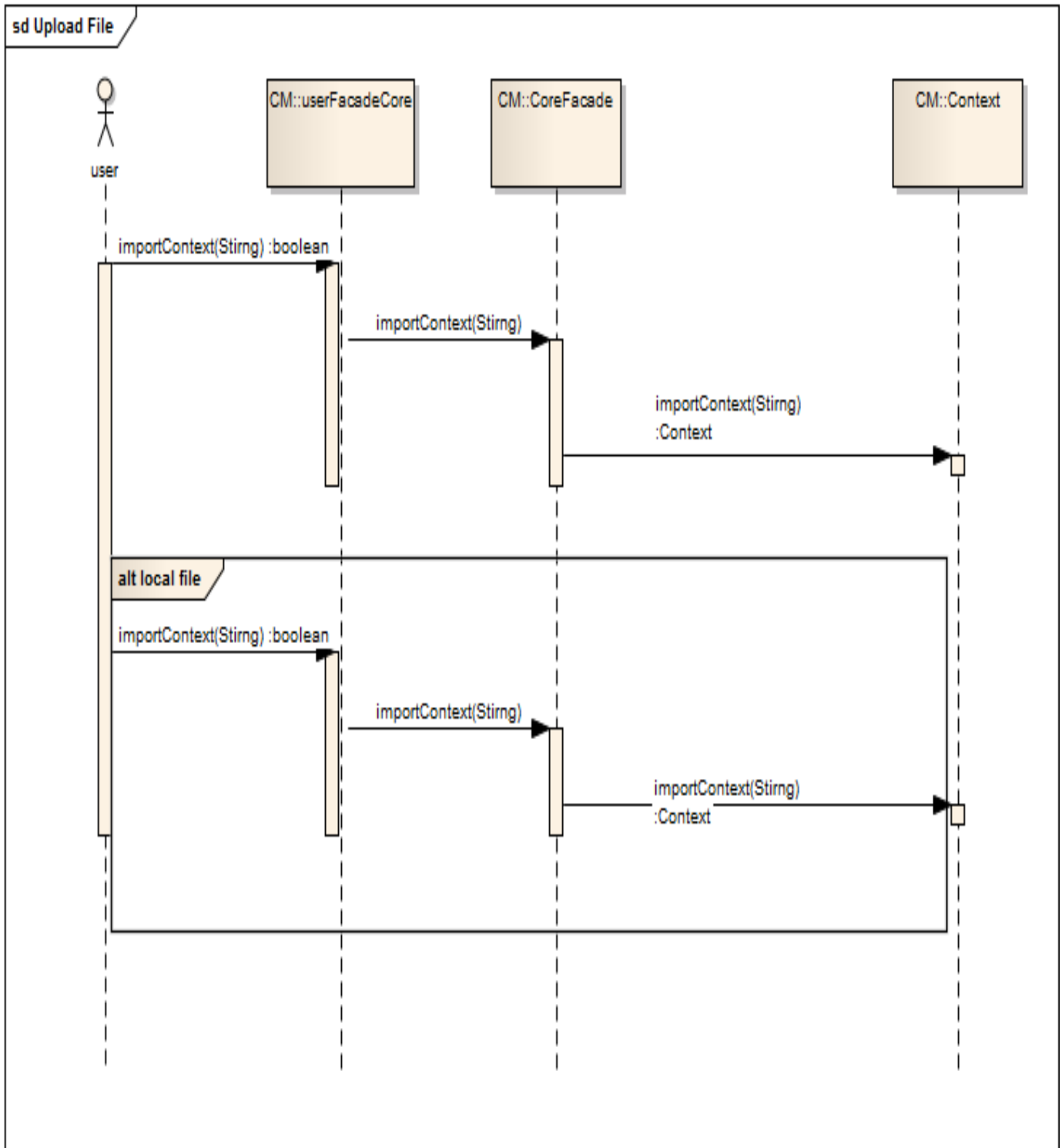
Export situation



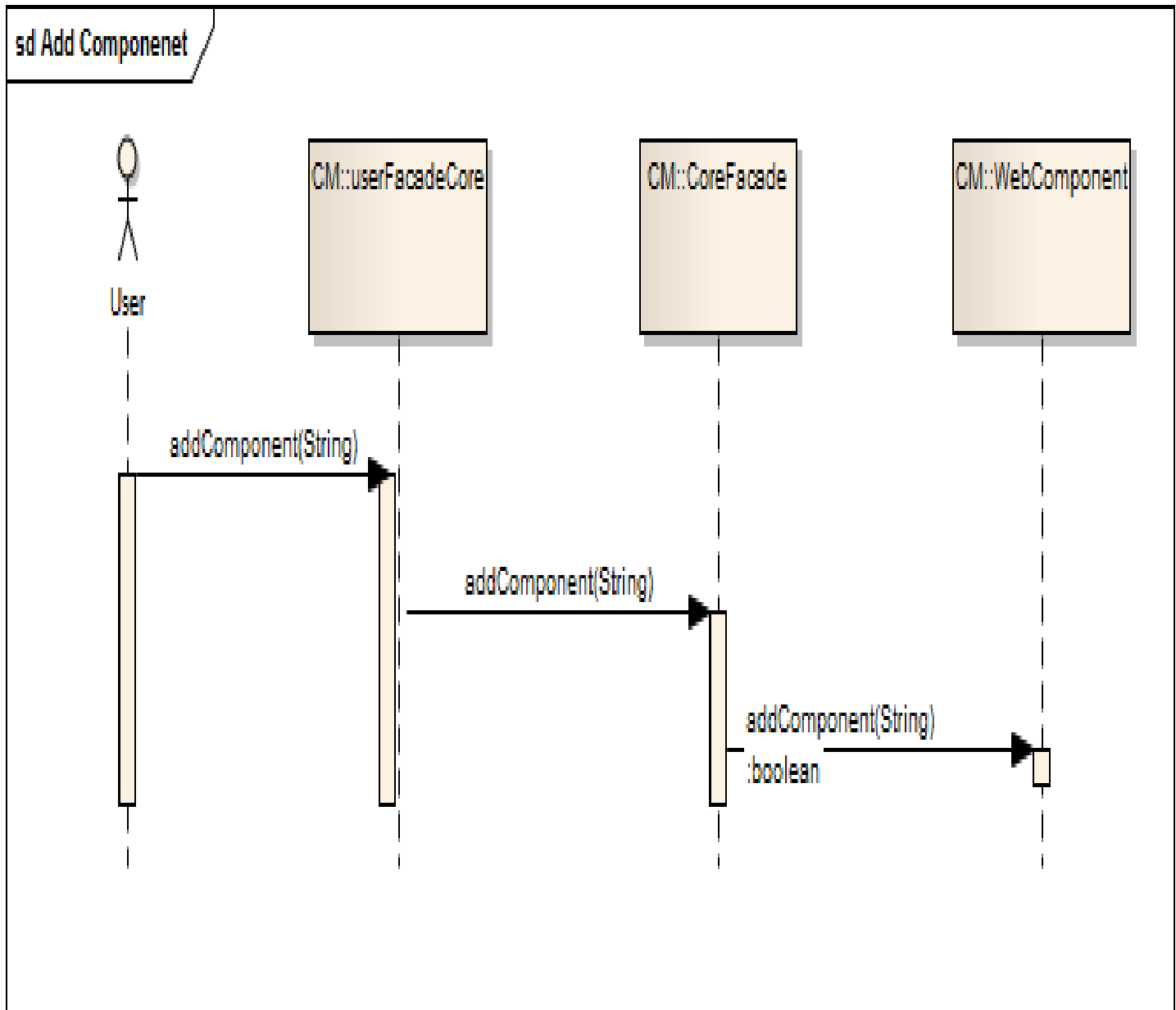
Add situation links



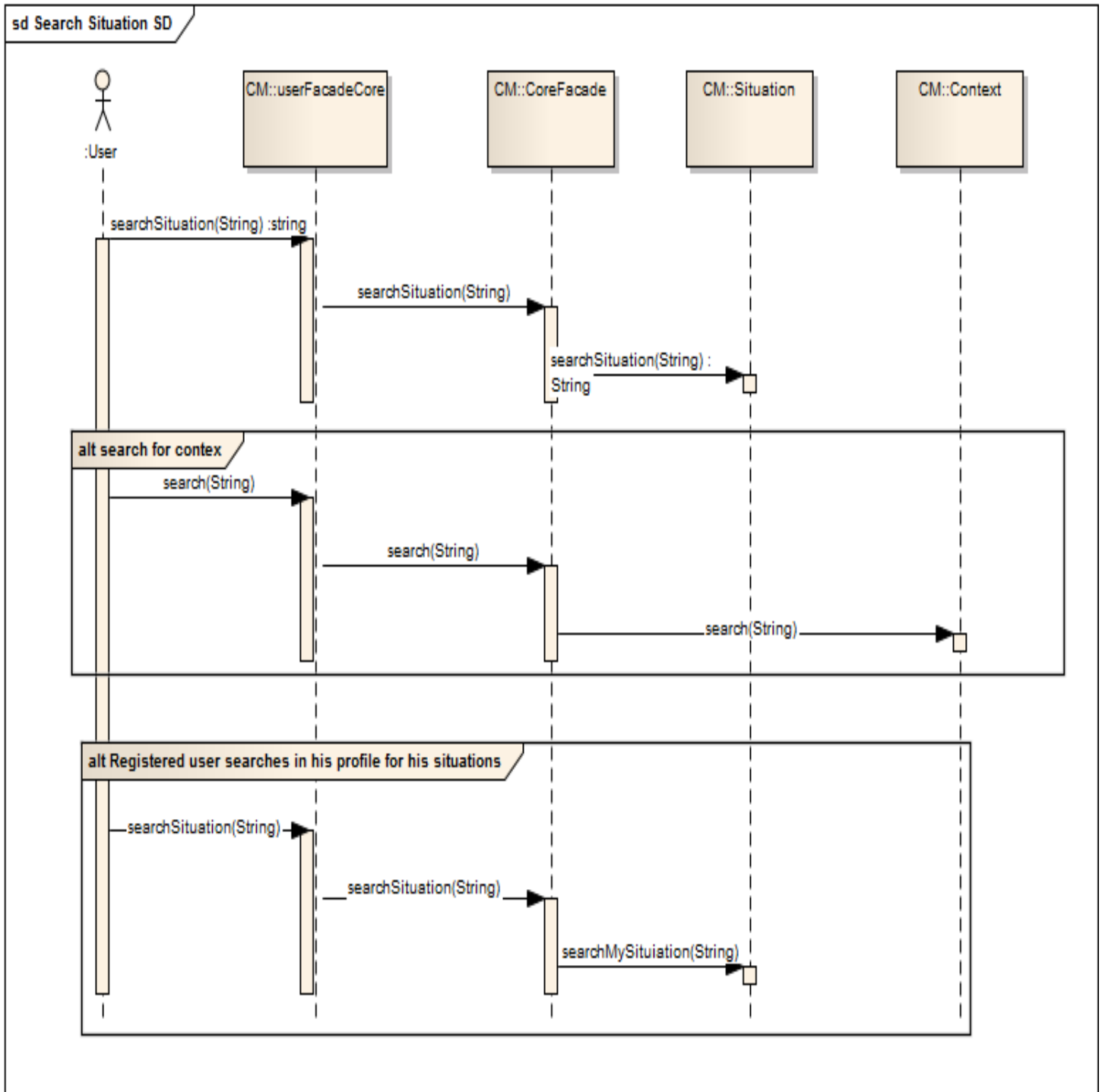
Upload file



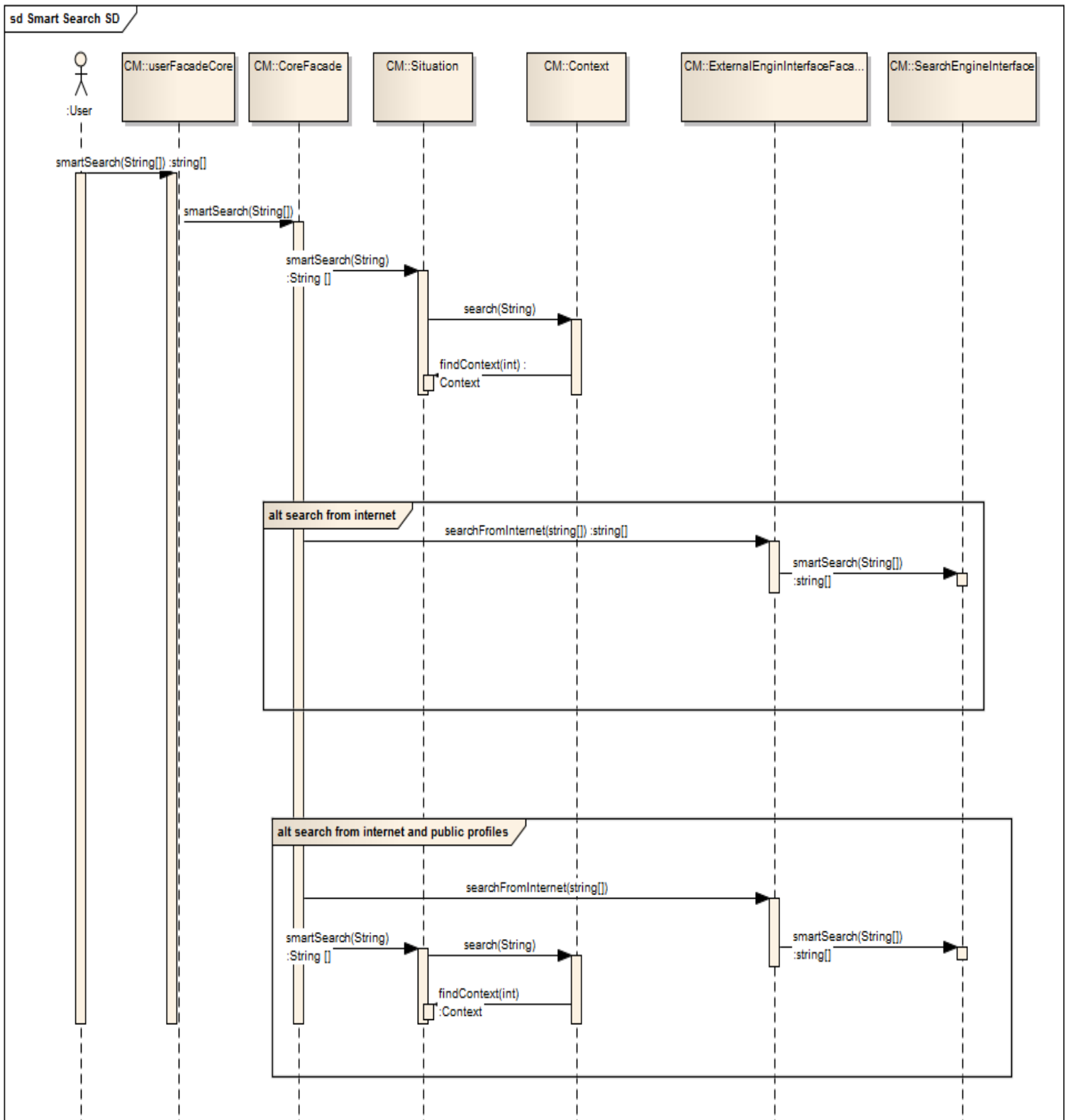
Add component



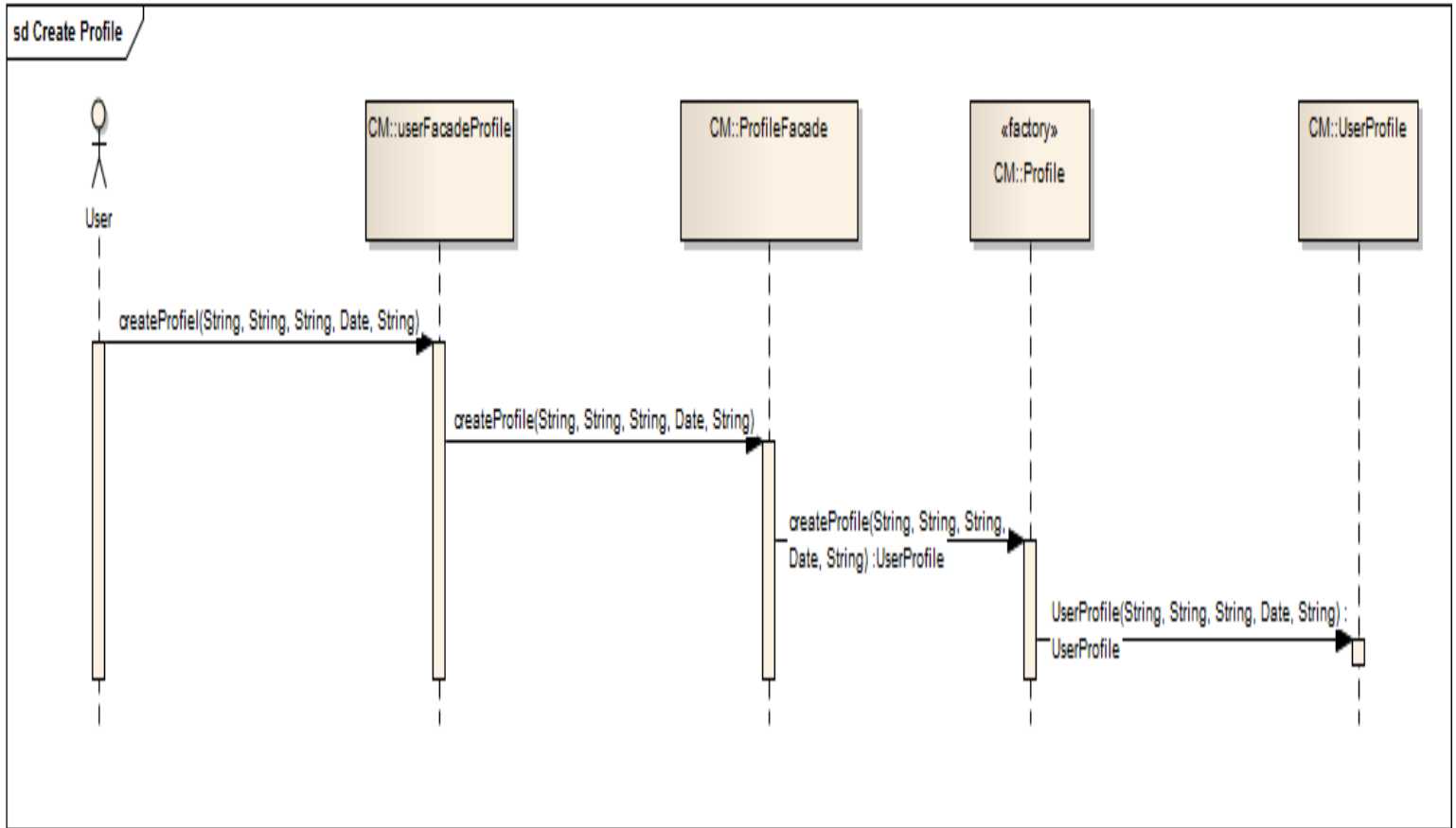
Search situation



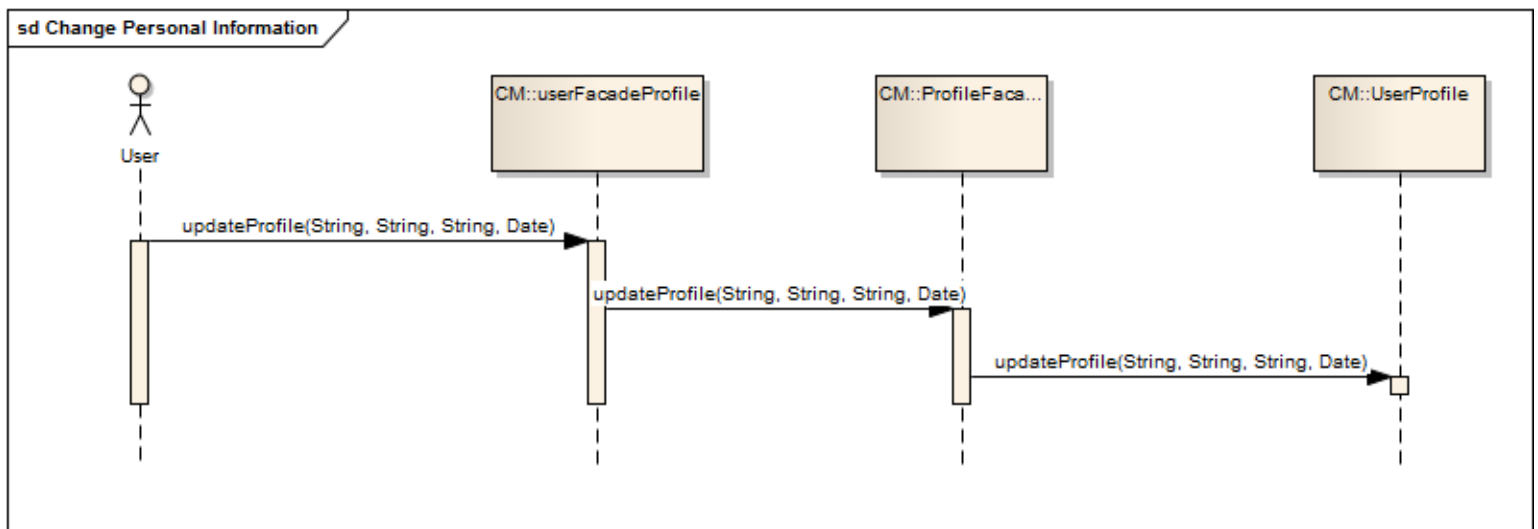
Smart search



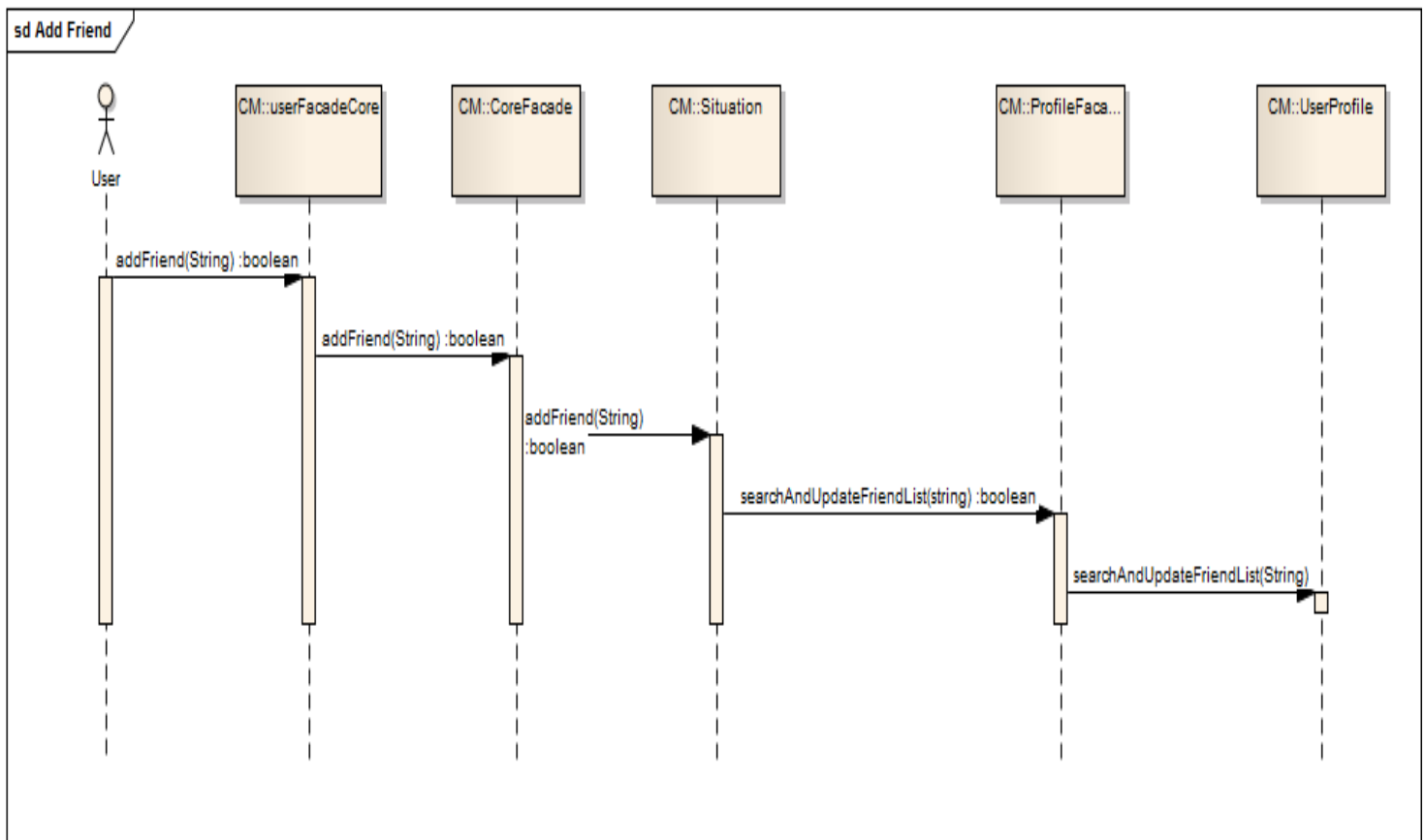
Create profile



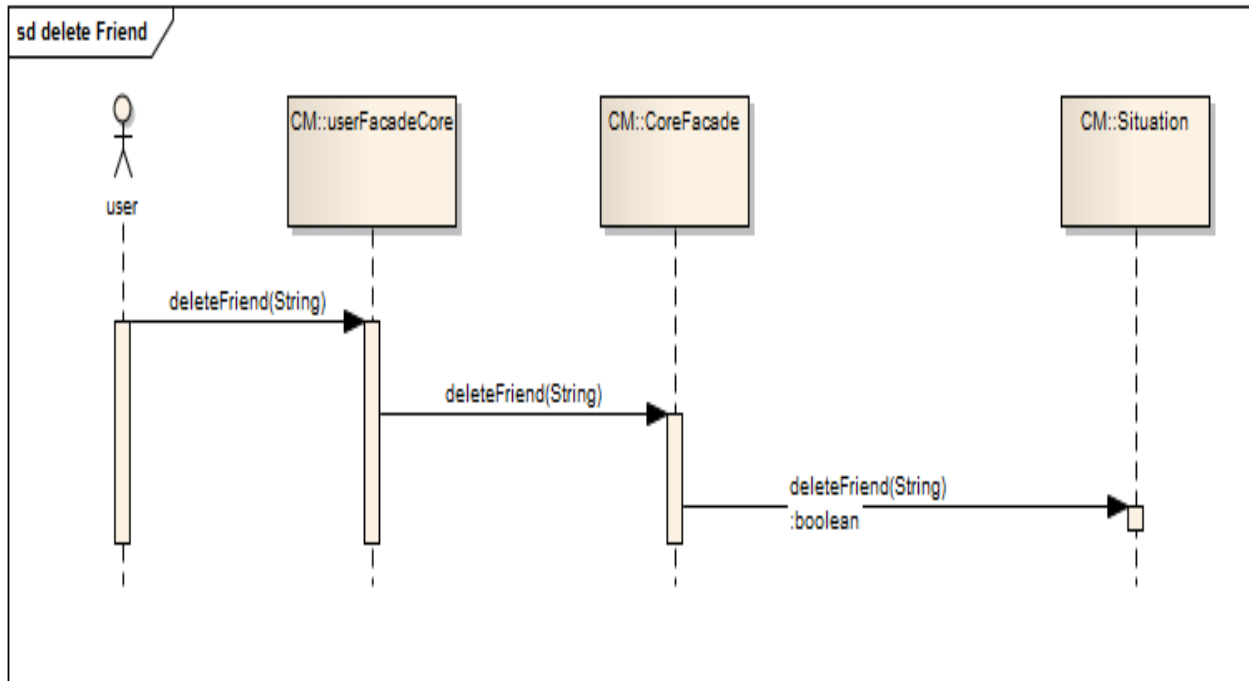
Change personal information



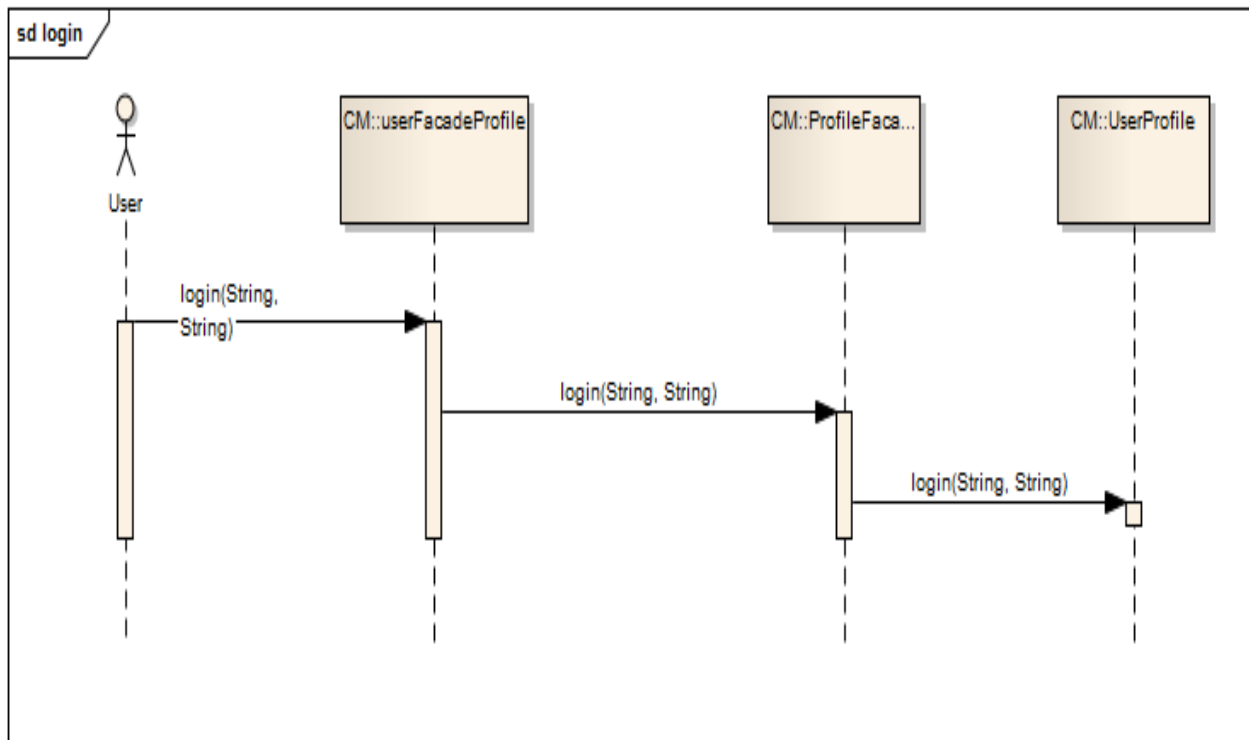
Add friend



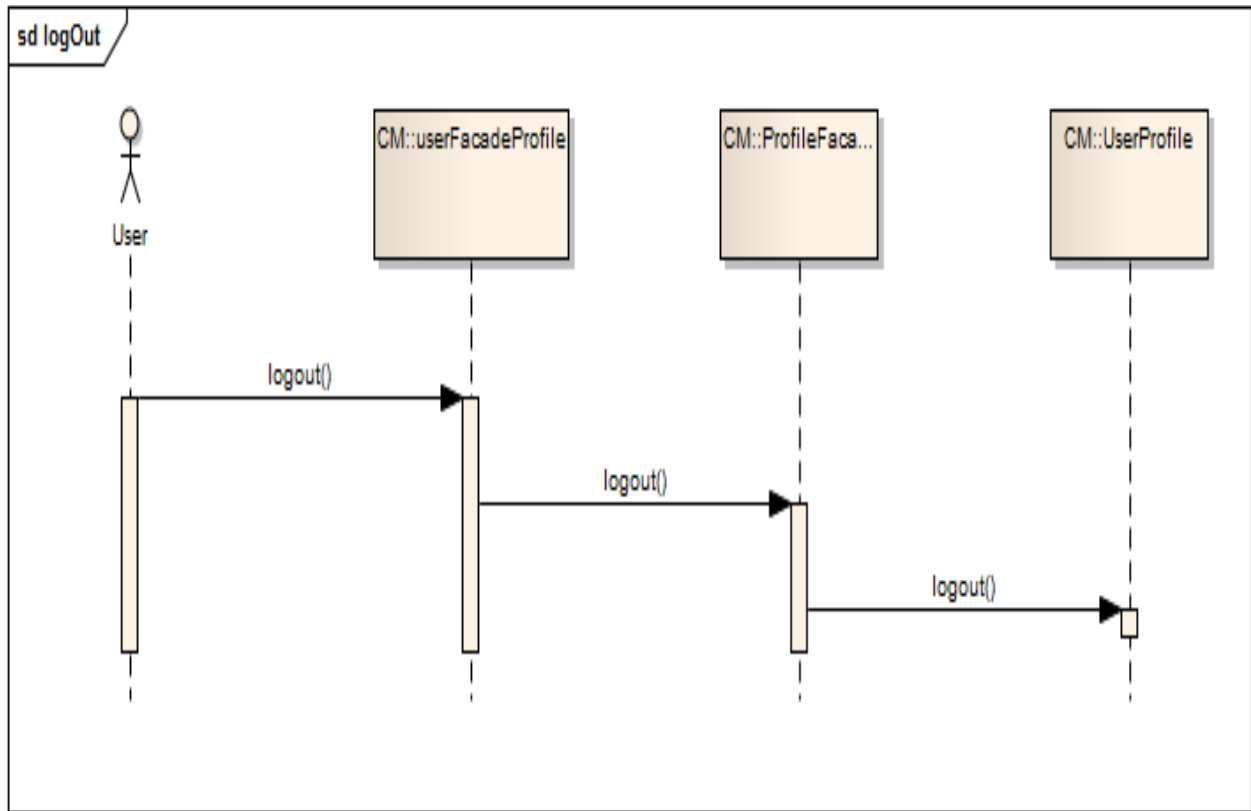
Delete friend



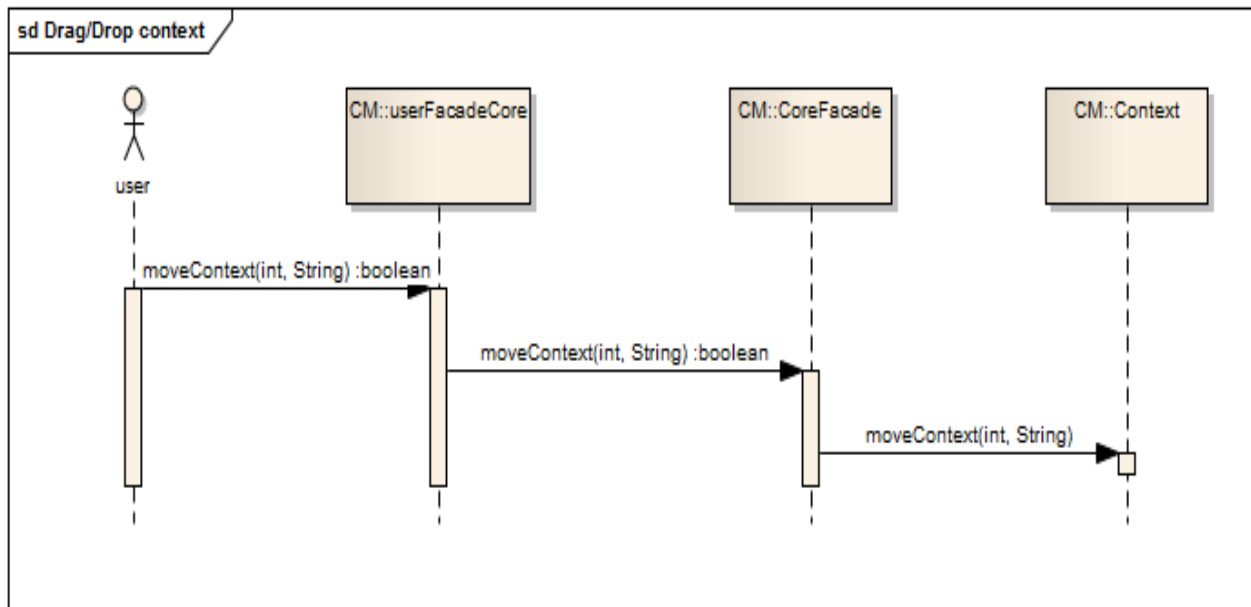
Log in



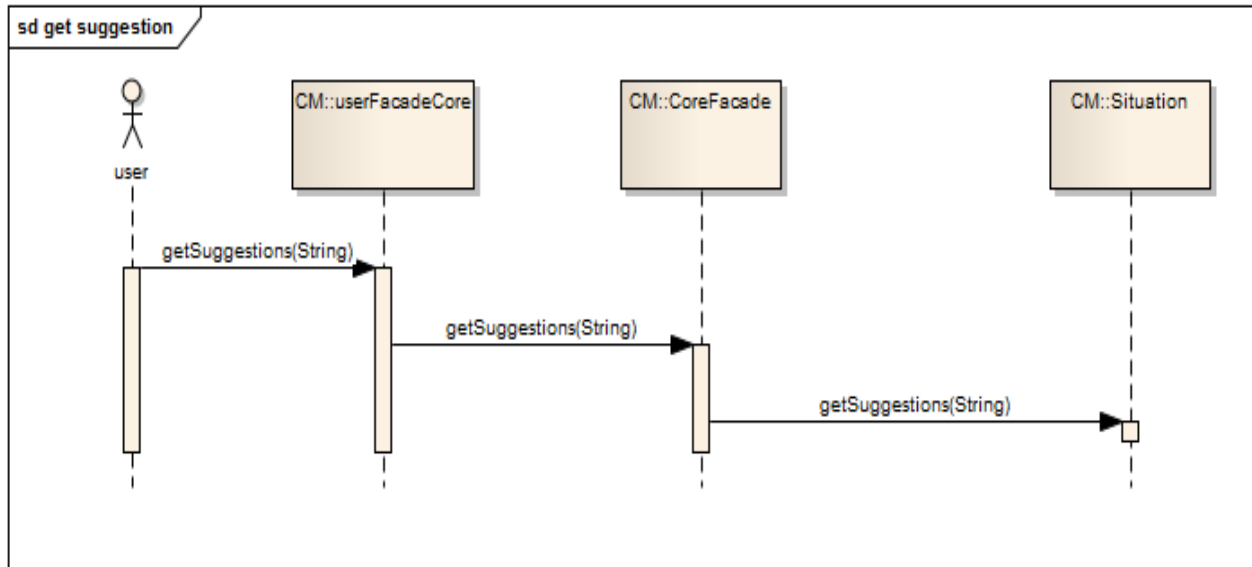
Log out



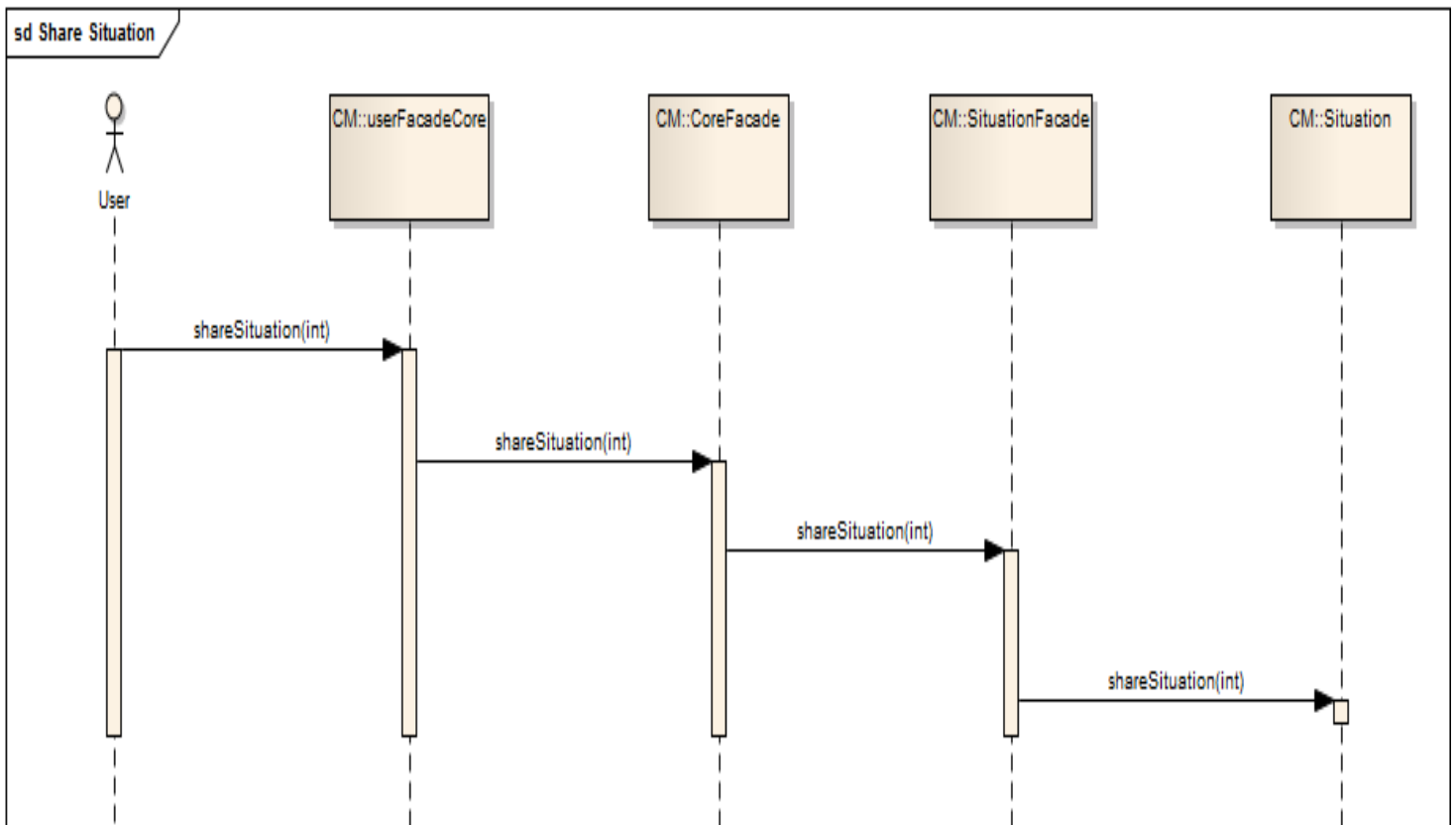
Drag Drop context



Get suggestion



Share Situation



Activity Models

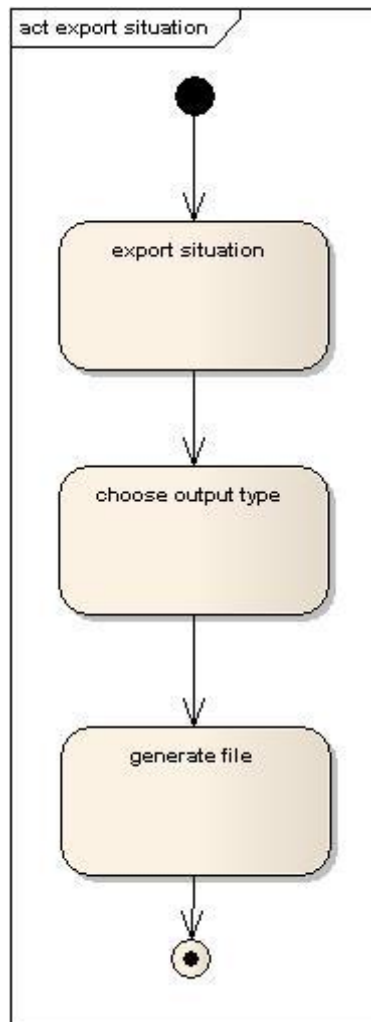


Figure 1 : export situation

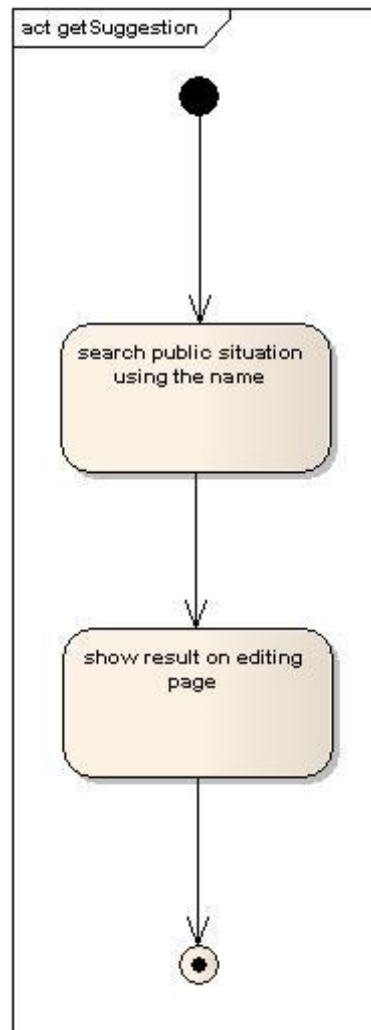


Figure 2 : get Suggestion

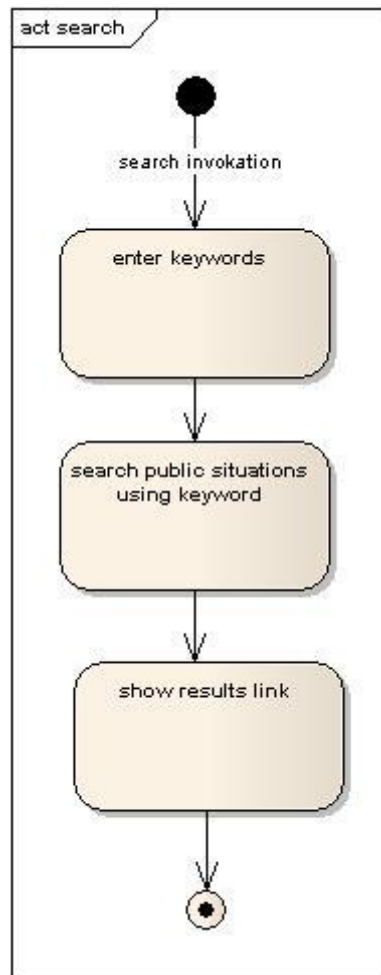


Figure 3 : search

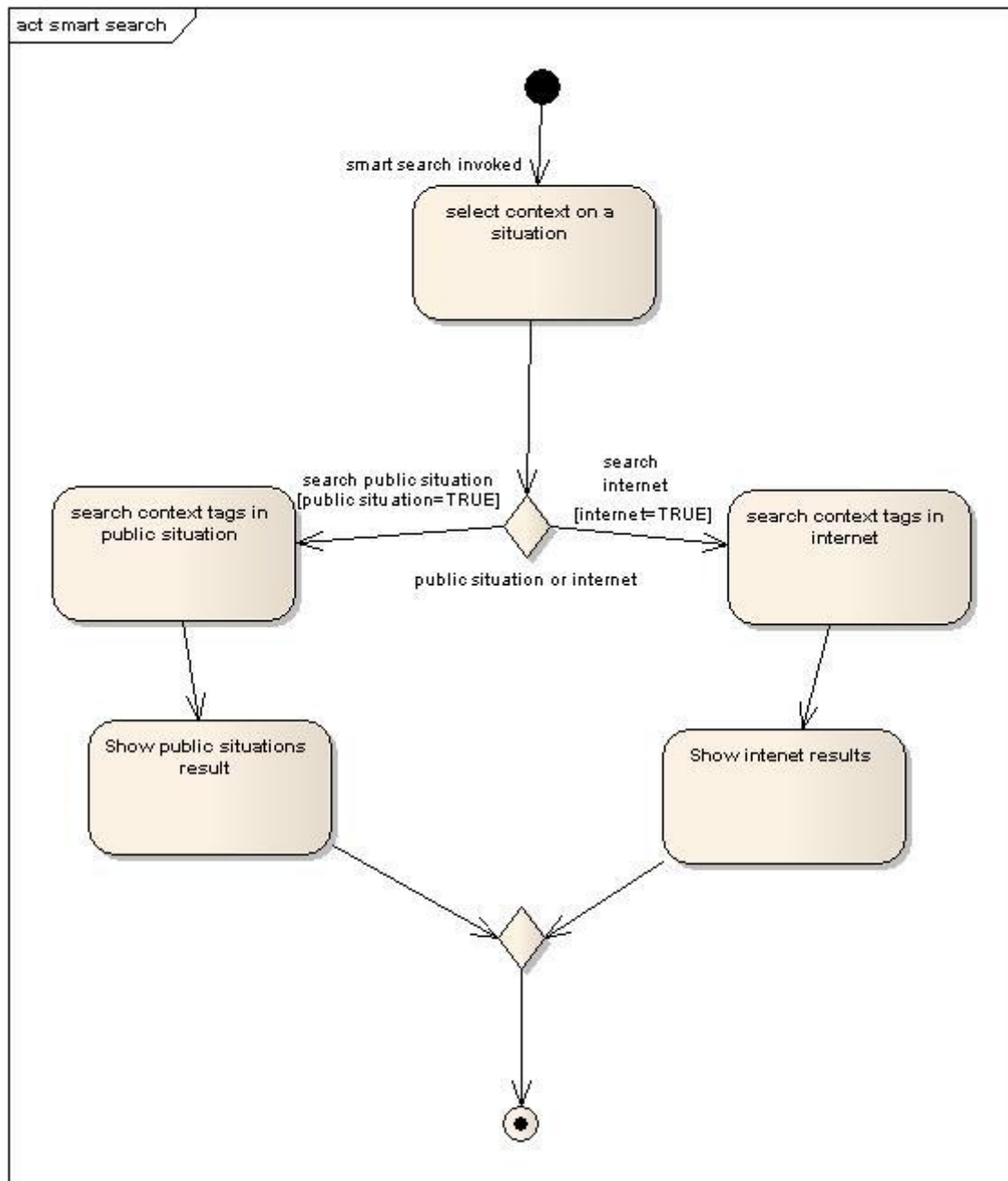
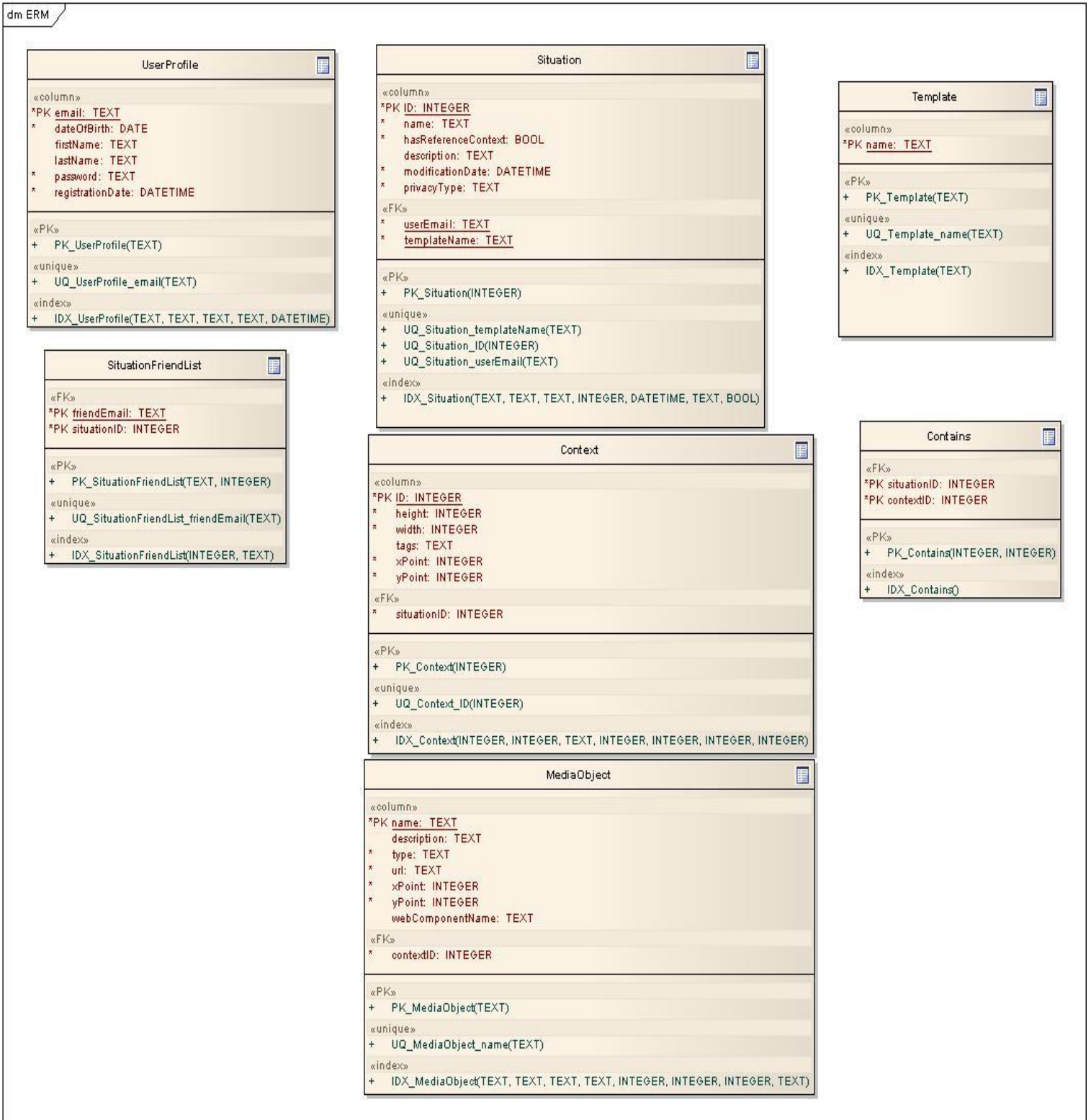


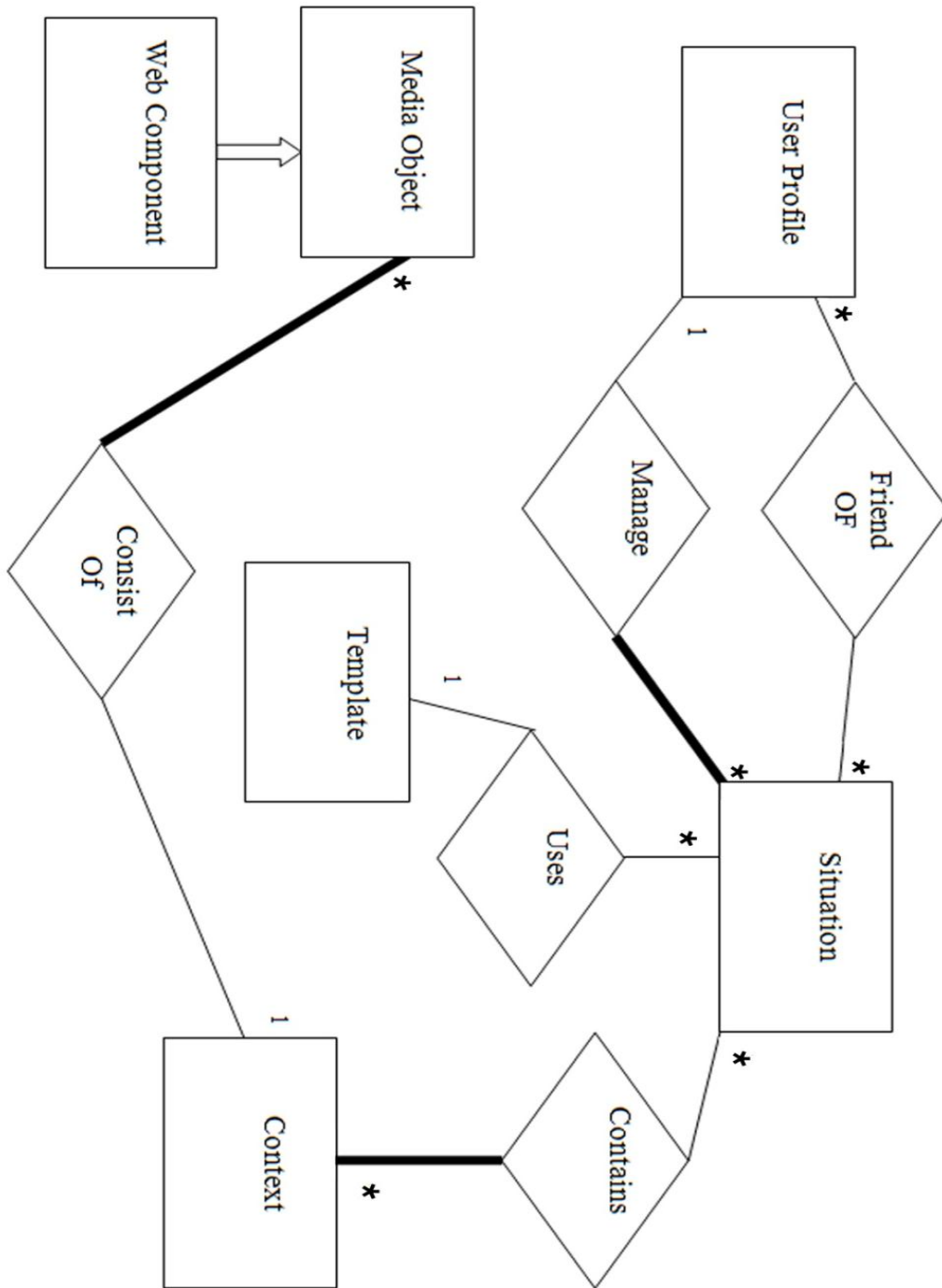
Figure 4 : smart search

Database Models & Architecture

ER Model



EER Model



Database DDL

Situation:

```
CREATE TABLE Situation
(
    ID INTEGER NOT NULL,
    name TEXT NOT NULL,
    hasReferenceContext BOOL NOT NULL,
    description TEXT,
    modificationDate DATETIME NOT NULL,
    privacyType TEXT NOT NULL,
    userEmail TEXT NOT NULL,
    templateName TEXT NOT NULL,
    PRIMARY KEY (ID),
    UNIQUE (templateName),
    UNIQUE (ID),
    UNIQUE (userEmail),
    INDEX IDX_Situation (name ASC, description ASC, userEmail ASC, ID ASC,
modificationDate ASC, privacyType ASC, hasReferenceContext ASC)
);
```

Context:

```
CREATE TABLE Context
(
    ID INTEGER NOT NULL,
    height INTEGER NOT NULL,
    width INTEGER NOT NULL,
    tags TEXT,
    xPoint INTEGER NOT NULL,
    yPoint BIGINT NOT NULL,
    situationID INTEGER NOT NULL,
    PRIMARY KEY (ID),
    UNIQUE (ID),
    INDEX IDX_Context (ID ASC, situationID ASC, tags ASC, height ASC, width ASC,
xPoint ASC, yPoint ASC)
);
```

Contains:

```
CREATE TABLE Contains
(
    situationID INTEGER NOT NULL,
    contextID INTEGER NOT NULL,
    PRIMARY KEY (situationID, contextID),
    INDEX IDX_Contains (situationID ASC, contextID ASC)
);
```

Media Object:

```
CREATE TABLE MediaObject
(
    name TEXT NOT NULL,
    description TEXT,
    type TEXT NOT NULL,
    url TEXT NOT NULL,
    xPoint INTEGER NOT NULL,
    yPoint INTEGER NOT NULL,
    webComponentName TEXT,
    contextID INTEGER NOT NULL,
    PRIMARY KEY (name),
    UNIQUE (name),
    INDEX IDX_MediaObject (name ASC, webComponentName ASC, type ASC,
description ASC, contextID ASC, xPoint ASC, yPoint ASC, url ASC)
);
```

Situation Friend List:

```
CREATE TABLE SituationFriendList
(
    friendEmail TEXT NOT NULL,
    situationID INTEGER NOT NULL,
    PRIMARY KEY (friendEmail, situationID),
    UNIQUE (friendEmail),
    INDEX IDX_SituationFriendList (situationID ASC, friendEmail ASC)
);
```

Situation List:

```
CREATE TABLE SituationList
(
    profileEmail TEXT NOT NULL,
    situationID INTEGER NOT NULL,
    PRIMARY KEY (profileEmail, situationID),
    UNIQUE (profileEmail),
    INDEX IDX_SituationList (profileEmail ASC, situationID ASC)
)
;
```

Template:

```
CREATE TABLE Template
(
    name TEXT NOT NULL,
    PRIMARY KEY (name),
    UNIQUE (name),
    INDEX IDX_Template (name ASC)
)
;
```

User Profile:

```
CREATE TABLE UserProfile
(
    email TEXT NOT NULL,
    dateOfBirth DATE NOT NULL,
    firstName TEXT,
    lastName TEXT,
    password TEXT NOT NULL,
    registrationDate DATETIME NOT NULL,
    PRIMARY KEY (email),
    UNIQUE (email),
    INDEX IDX_UserProfile (firstName ASC, lastName ASC, email ASC, password ASC,
registrationDate ASC)
);
```