# CO6021 Advanced Systems Analysis and Design

**Dance Studio Solution 1**

Functional and Non-functional Requirements

## 1.1 Functional Requirements

* The system must allow for dance tutors to change content displayed within the website.
* The system must allow customers to book dance classes online.
* The system must allow customers to cancel bookings if the dance class is more than 2 days away.
* The system must allow dance tutors to cancel dance classes that are more than 2 days away and notify/refund attendees of the cancellation.
* The system must feature an online payment system for customers.
* The system must be able to store customer credit card details safely for future purchases.
* The system must keep records of all online transactions.
* The system must display information about each of the classes the dance studio offers.
* The system must allow dance tutors to upload rich media content such as images and videos.
* The system must contain a feedback form for customers and unregistered users to use.
* The system must allow dance tutors to respond to feedback sent by users.
* The system must allow dance tutors to add and delete classes and events as necessary.
* The system must allow new customers to register.
* The system must allow existing customers to delete their accounts if necessary.

## 1.2 Non-Functional Requirements

* The system must be assessable through a web browser.
* The system must comply with modern web standards to ensure website is displayed correctly in all browsers and environments.
* The system design must incorporate the company’s logo.
* The system must allow dance tutors to enter and edit content in a well-designed interface similar to MS Office Word.
* The system design and content must appear to all age ranges with a focus on children.
* The system must utilise the .NET 2.0 framework.

# 2.0 Use Cases and Use Case Diagram

## 2.1 Use Case 1: Customer Registration

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use case name: Customer Registration | | | ID: 1 | Importance Level: High | |
| Primary Actor: Customer | | | | | |
| Short Description: Customer registers to the website. | | | | | |
| Trigger: Customer clicks register button from the website  Type: External | | | | | |
| Major Inputs | | Major Outputs | | | |
| Description  First Name  Second Name  E-mail  Address  Tel Number  Gender  DOB  Password  Re-entered Password  Secret Question  Secret Answer | Source  Registration Form  Registration Form  Registration Form  Registration Form  Registration Form  Registration Form  Registration Form  Registration Form  Registration Form  Registration Form  Registration Form | Description  Customer Account Details | | | Destination  Accounts database |
| Major steps performed:   1. New customer enters details required within the registration form. 2. New customer clicks submit/register button. 3. Data entered is validated 4. E-mail is checked against accounts database to ensure it is unique. 5. E-mail validation request is sent to the customers e-mail address 6. Customer account created (with limited access until e-mail validated) | | | | Information of Steps: | |

## 2.2 Use Case 2: Login

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use case name: Login | | | ID: 2 | Importance Level: High | |
| Primary Actors: Customer / Employees / Admin | | | | | |
| Short Description: A person logs in to the website | | | | | |
| Trigger: A person clicks the login button from the website  Type: External | | | | | |
| Major Inputs | | Major Outputs | | | |
| Description  E-mail Address  Password | Source  Login form  Login form | Description | | | Destination |
| Major steps performed:   1. Person enters details required within the login form. 2. Person clicks the login button. 3. System attempts to find the e-mail address within the accounts database. If e-mail address does not exist the system displays an error. 4. If the e-mail address exists, the system checks the password entered against the password stored within the accounts database. If the password is incorrect the system displays an error. 5. If the password is correct, the person is logged into the system. | | | | Information of Steps: | |

## 2.3 Use Case 3: Book dance class

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use case name: Book Dance Class | | | ID: 3 | Importance Level: High | |
| Primary Actor: Customer | | | | | |
| Short Description: Customer books a dance class through the website. | | | | | |
| Trigger: Customer clicks book button from the website.  Type: External | | | | | |
| Major Inputs | | Major Outputs | | | |
| Description  Account ID  Class ID  Class type  Class time  Class date  Class duration  Card details | Source  Accounts database  Classes Database  Classes Database  Classes Database  Classes Database  Classes Database  Payment Database | Description  Customer Invoice  Order details | | | Destination  Customers e-mail  Orders Database |
| Major steps performed:   1. System checks to see whether the dance class selected by the customer is not fully booked. If the class is fully booked the system displays an error. 2. If the class is not fully booked, the system prompts the user to select a credit card that is already saved to their account or asks the customer to add a credit card. 3. Once the customer clicks the pay/confirm button, the system validates the payment information entered. If the information is invalid the system displays an error. 4. If the payment details are valid the system processes the payment and shows confirmation of the payment to the customer. 5. The system adds the customer’s details to the attendee’s database for the selected dance class. 6. An invoice is sent to the customers e-mail and a log of the order is added to the orders database. | | | | Information of Steps: | |

## 2.4 Use Case 4: Cancel dance class attendance

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use case name: Cancel dance class attendance | | | ID: 4 | Importance Level: Medium | |
| Primary Actor: Customer | | | | | |
| Short Description: Customer cancels a dance class attendance through the website. | | | | | |
| Trigger: Customer clicks cancel button  Type: External | | | | | |
| Major Inputs | | Major Outputs | | | |
| Description  Class ID  Customer ID | Source  Class Database  Customer Database | Description  Cancellation confirmation | | | Destination  Customer e-mail |
| Major steps performed:   1. Customer clicks cancel button 2. If the class is more than 48 hours away cancellation process begins. If the class is not more than 48 hours away, an error is displayed. 3. The system searches for the class and removes the customer from the attendee’s database for the dance class. 4. The system refunds the cost of the dance class minus the cancellation fee to the customer credit card account. 5. A cancellation confirmation is sent to the customers e-mail address. | | | | Information of Steps: | |

## 2.5 Use Case 5: Add new dance class

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use case name: Add new dance class | | | ID: 5 | Importance Level: High | |
| Primary Actor: Employee / Admin | | | | | |
| Short Description: A dance tutor / admin adds a new dance class. | | | | | |
| Trigger: A dance tutor / admin clicks the add class button through the website.  Type: External | | | | | |
| Major Inputs | | Major Outputs | | | |
| Description  Account ID  Tutor name  Class type  Class time  Class date  Class Duration  Room number  Price  No of places | Source  Accounts database  Accounts database  Add class form  Add class form  Add class form  Add class form  Add class form  Add class form  Add class form | Description  New class record | | | Destination  Classes Database |
| Major steps performed:   1. Check whether time/date/room number, do not clash with any existing dance classes. If details clash display an error. 2. Check whether dance tutor is free during the time slot entered. If tutor is not free display an error. 3. Prompt tutor / admin to give confirmation that the details are correct. 4. If confirmation is given add the class to the classes database. | | | | Information of Steps: | |

## 2.6 Use Case 6: Customer feedback form

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use case name: Customer feedback form | | | ID: 6 | Importance Level: Medium | |
| Primary Actor: Customer | | | | | |
| Short Description: Customer submits a feedback form through the website | | | | | |
| Trigger: Customer clicks send feedback button from the website  Type: External | | | | | |
| Major Inputs | | Major Outputs | | | |
| Description  Customer ID  Feedback category  Feedback description | Source  Customer Database  Feedback form  Feedback form | Description  Feedback details | | | Destination  Feedback Database |
| Major steps performed:   1. Customer is prompted to check feedback form to ensure feedback is correct. 2. If feedback is correct the feedback details and the customers ID are stored in the feedback database. 3. The system informs Tutor and Admin users that there is new feedback | | | | Information of Steps: | |

## 2.7 Use Case 7: Unregistered user feedback form

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use case name: Unregistered user feedback form | | | ID: 7 | Importance Level: Medium | |
| Primary Actor: Unregister user | | | | | |
| Short Description: An unregistered user sends a feedback form through the website. | | | | | |
| Trigger: Customer clicks submit feedback button from the website.  Type: External | | | | | |
| Major Inputs | | Major Outputs | | | |
| Description  First name  Second name  E-mail address  Feedback category  Feedback description | Source  Feedback form  Feedback form Feedback form Feedback form Feedback form | Description  Feedback details | | | Destination  Feedback Database |
| Major steps performed:   1. Customer is prompted to check feedback form to ensure feedback is correct. 2. If feedback is correct the feedback details and the users details are stored in the feedback database. 3. The system informs Tutor and Admin users that there is new feedback | | | | Information of Steps: | |

## 2.8 Use Case 8: Tutor cancel dance class

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use case name: Tutor cancel dance class | | | ID: 8 | Importance Level: High | |
| Primary Actor: Dance tutor / Admin | | | | | |
| Short Description: A tutor / admin cancels a upcoming dance class | | | | | |
| Trigger: A tutor / admin clicks the cancel class button.  Type: External | | | | | |
| Major Inputs | | Major Outputs | | | |
| Description  Class ID  Customer ID | Source  Classes Database  Attendees Database | Description  Cancellation notice | | | Destination  Customer e-mail |
| Major steps performed:   1. Tutor / admin is prompted to confirm they want to cancel the selected class. If the class is less than 48 hours away an error is displayed. 2. If the class is further than 48 hours away all customer attendees are informed by e-mail of the cancellation and refunded the full cost of the dance class. 3. Class is removed from the classes database and all attendees are removed from the attendees database. | | | | Information of Steps: | |

## 2.9 Use Case 9: Add credit card

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use case name: Add credit card | | | ID: 9 | Importance Level: High | |
| Primary Actor: Customer | | | | | |
| Short Description: A customer adds a credit card to their account. | | | | | |
| Trigger: Customer clicks the add credit card button from the website or is prompted by the system while attempting to book a dance class.  Type: External | | | | | |
| Major Inputs | | Major Outputs | | | |
| Description  Customer ID  Card type  Credit card number  Expiration date  Security number | Source  Customer Database  Add credit card form  Add credit card form  Add credit card form  Add credit card form | Description  Credit card details | | | Destination  Payment Database |
| Major steps performed:   1. Customer is prompted to enter the details required to add a credit card to the customers account. 2. The system prompts the customer the check that the details entered are correct. 3. The system validates the details entered to ensure they are correct, if not an error is displayed. 4. The system encrypts the credit card details given by the user. 5. The system adds the encrypted credit card details along with the customers ID to the payment database. | | | | Information of Steps: | |

## 2.10 Use Case 10: Add news/testimonials items

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use case name: Add news/testimonial item | | | ID: 10 | Importance Level: High | |
| Primary Actor: Dance tutor / Admin | | | | | |
| Short Description: A tutor / admin adds a new news/testimonial item to the website | | | | | |
| Trigger: A tutor / admin clicks the add news button from the website  Type: External | | | | | |
| Major Inputs | | Major Outputs | | | |
| Description  Account ID  Title  Content  Uploads | Source  Accounts Database  Add News Form  Add News Form  Add News Form | Description  News Record  Uploads | | | Destination  News Database  Upload Database |
| Major steps performed:   1. Tutor / admin clicks the add news button. 2. Tutor / admin is prompted to input a title, and any content needed through a WYSIWYG editor. 3. If any uploads such as videos/pictures are required tutor / admin clicks the upload button and selects file. 4. System validates file and then adds it to the WYSIWYG editor. 5. Tutor / admin clicks the submit button. 6. System prompts tutor / admin to confirm news post is correct. 7. If post correct, a new record is added to the news database. | | | | Information of Steps: | |

## 2.11 Use Case 11: Respond to feedback

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use case name: Respond to feedback | | | ID: 11 | Importance Level: High | |
| Primary Actor: Dance tutor / Admin | | | | | |
| Short Description: A tutor / admin responses to feedback send by a user | | | | | |
| Trigger: A tutor / admin clicks the respond button  Type: External | | | | | |
| Major Inputs | | Major Outputs | | | |
| Description  Feedback ID  Category  Description  Account ID (Employee)  User e-mail | Source  Feedback Database  Feedback Database  Feedback Database  Accounts Database  Feedback Database | Description  Response e-mail | | | Destination  Users e-mail address |
| Major steps performed:   1. Tutor / admin clicks the respond button. 2. System displays feedback category and description to the employee (admin/tutor) 3. If the tutor/admin is able to answer the feedback they fill out the answer box. 4. The tutor/admin clicks the send button once completed. 5. System prompts user to confirm feedback response is correct. 6. If feedback response is correct it is sent to the users e-mail address. 7. Confirmation of this action is displayed to the tutor/admin | | | | Information of Steps: | |

## 2.12 Use Case Diagram

## Description: Macintosh HD:Users:Lewis:Desktop:Screen shot 2011-04-05 at 00.27.55.png

# 3.0 UML modeling for the proposed system

## 3.1 Class diagram

# Description: Macintosh HD:Users:Lewis:Desktop:Screen shot 2011-04-05 at 00.32.40.png

## 3.2 Sequence diagrams

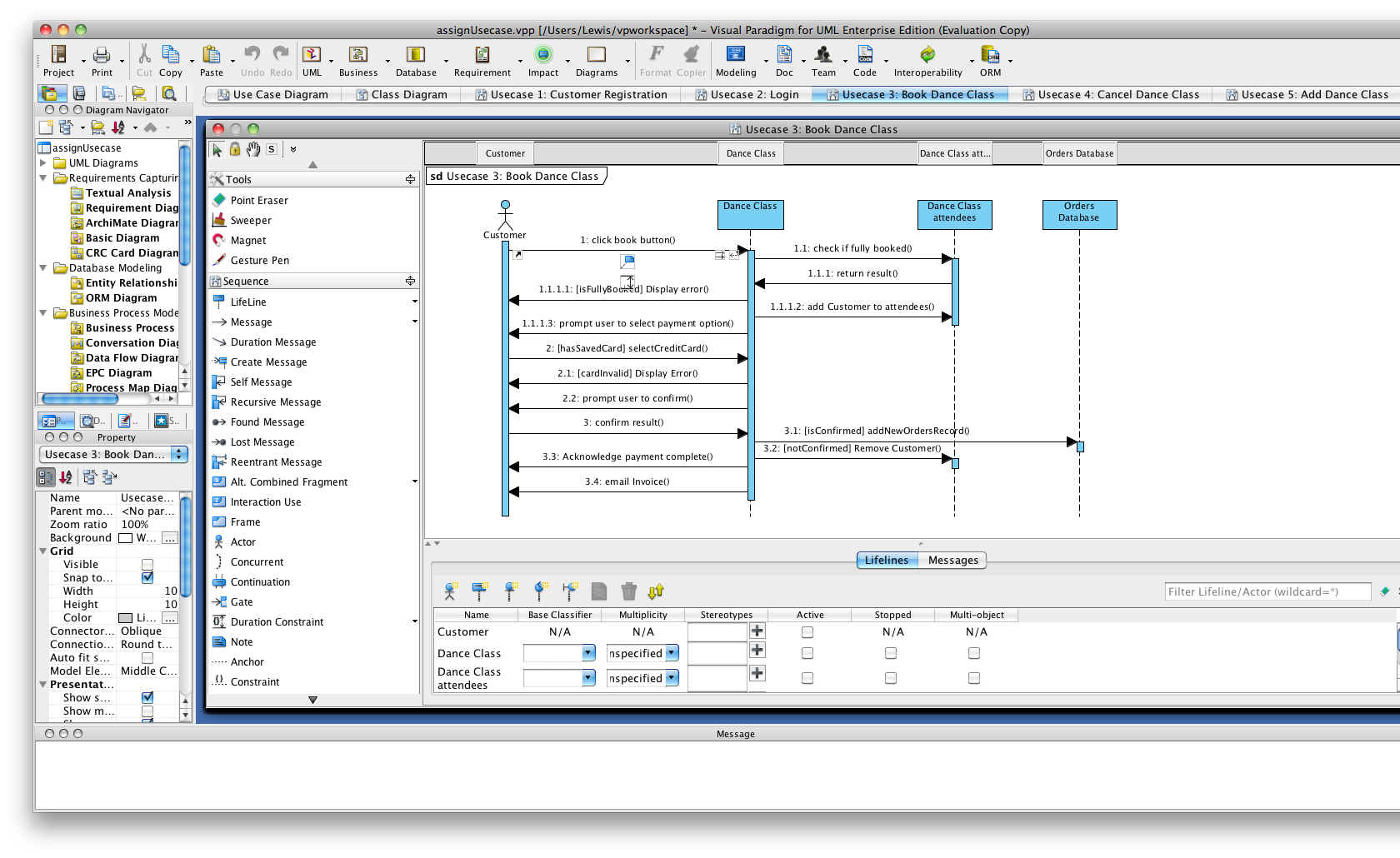
### 3.2.1 Use Case 1: Customer registration

### Description: Macintosh HD:Users:Lewis:Desktop:Screen shot 2011-04-05 at 00.35.43.png

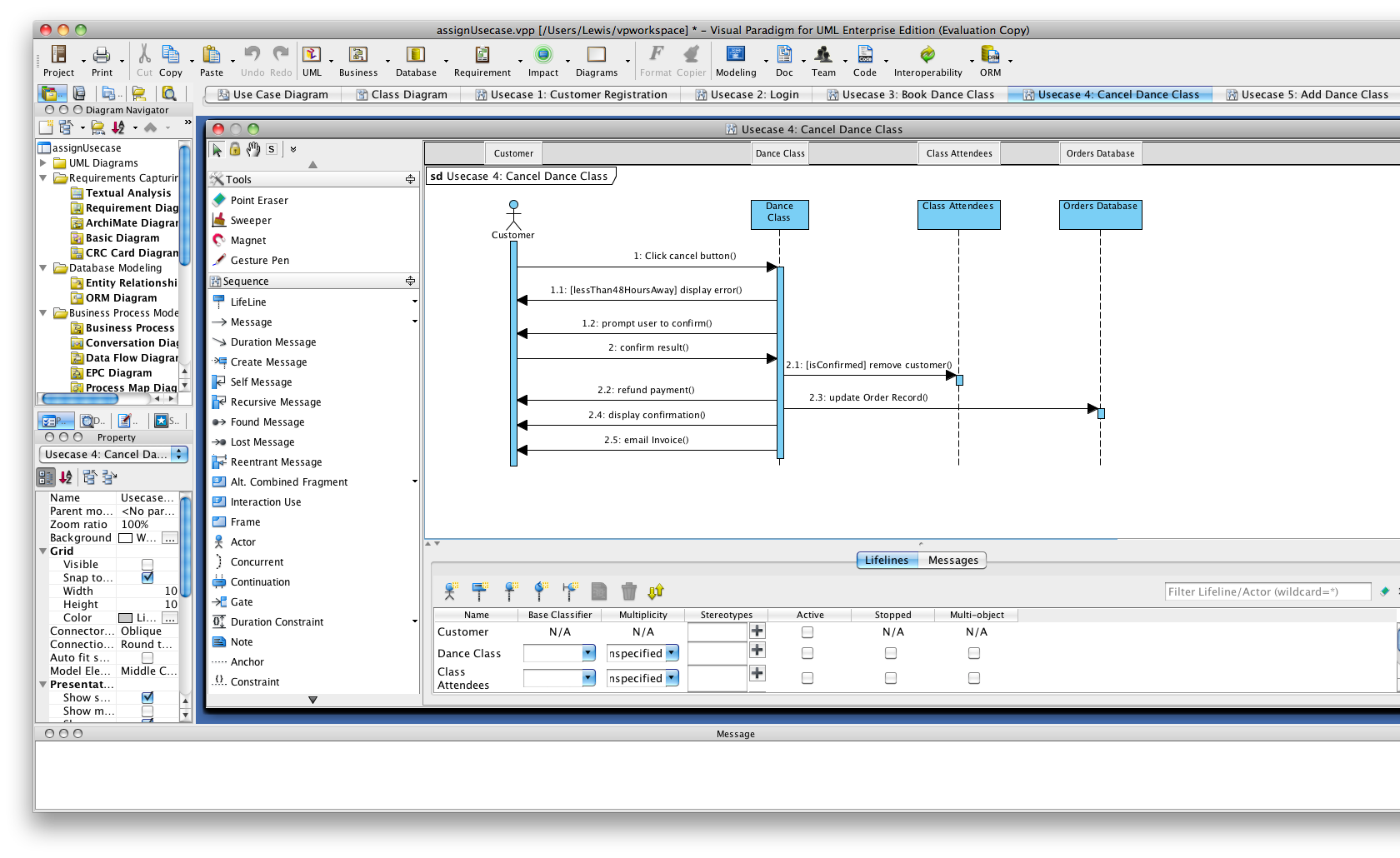
### 3.2.2 Use Case 2: Login

### Description: Macintosh HD:Users:Lewis:Desktop:Screen shot 2011-04-05 at 00.37.43.png

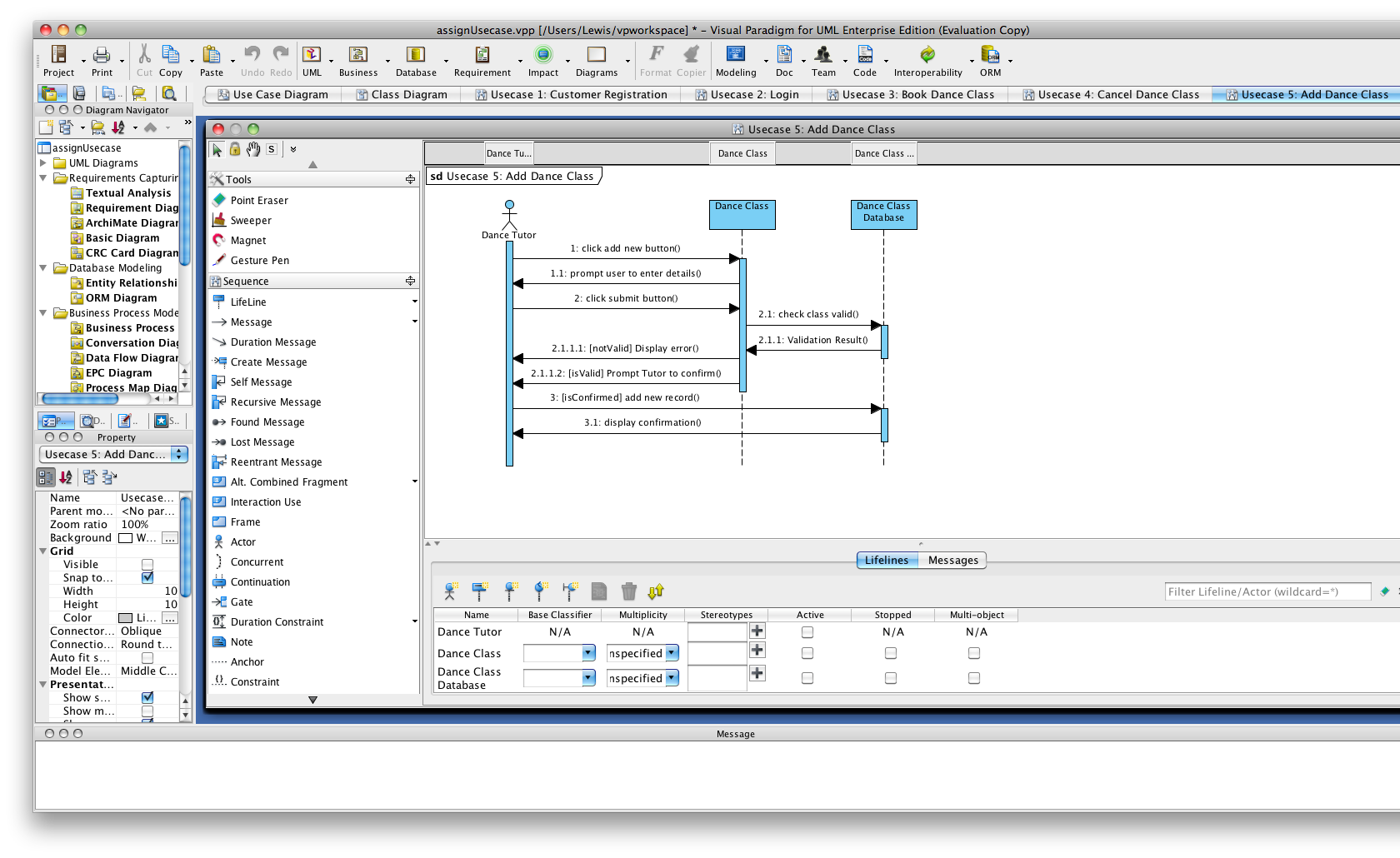
### 3.2.3 Use Case 3: Book dance class



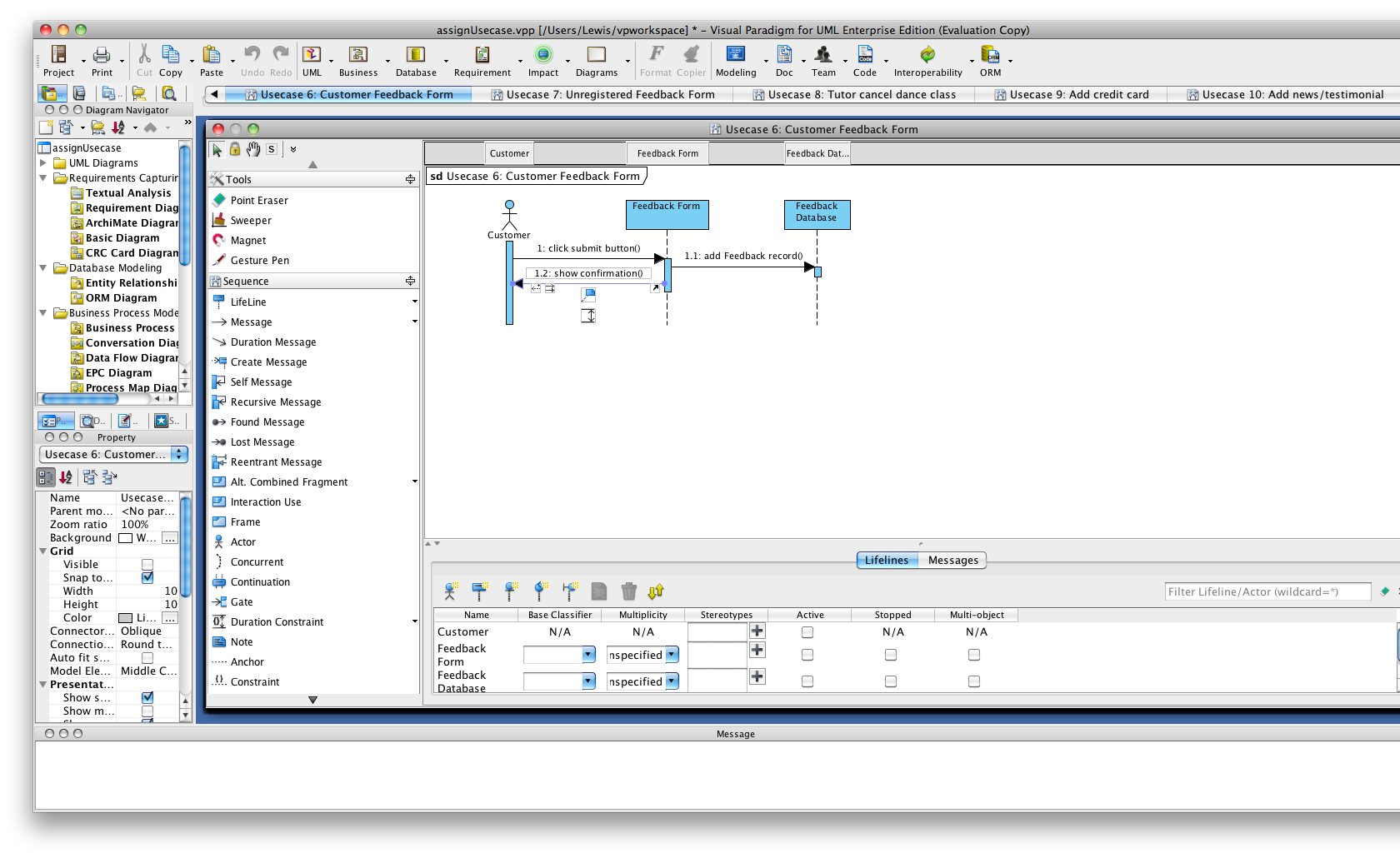
### 3.2.4 Use Case 4: Cancel dance class attendance



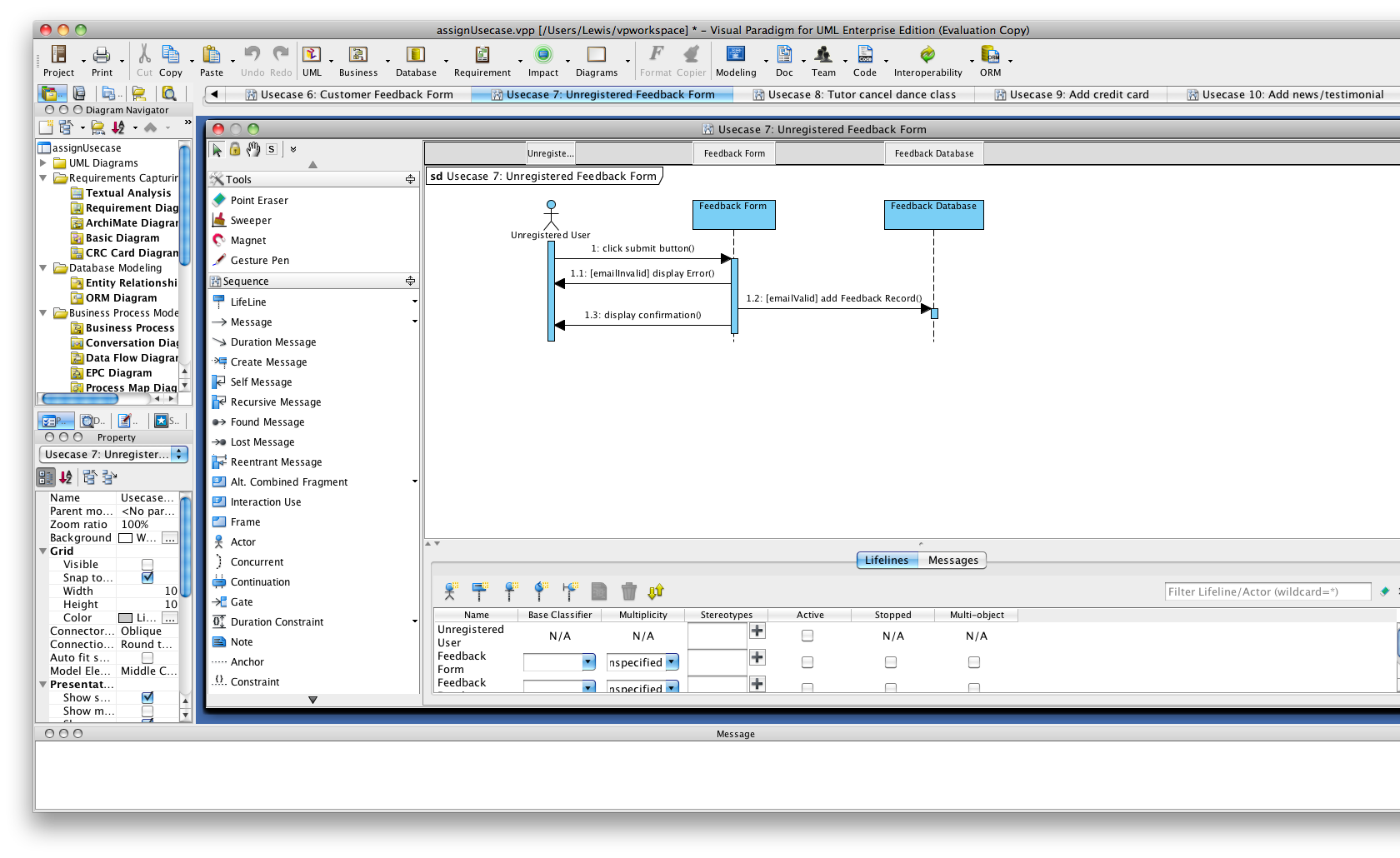
### 3.2.5 Use Case 5: Add dance class



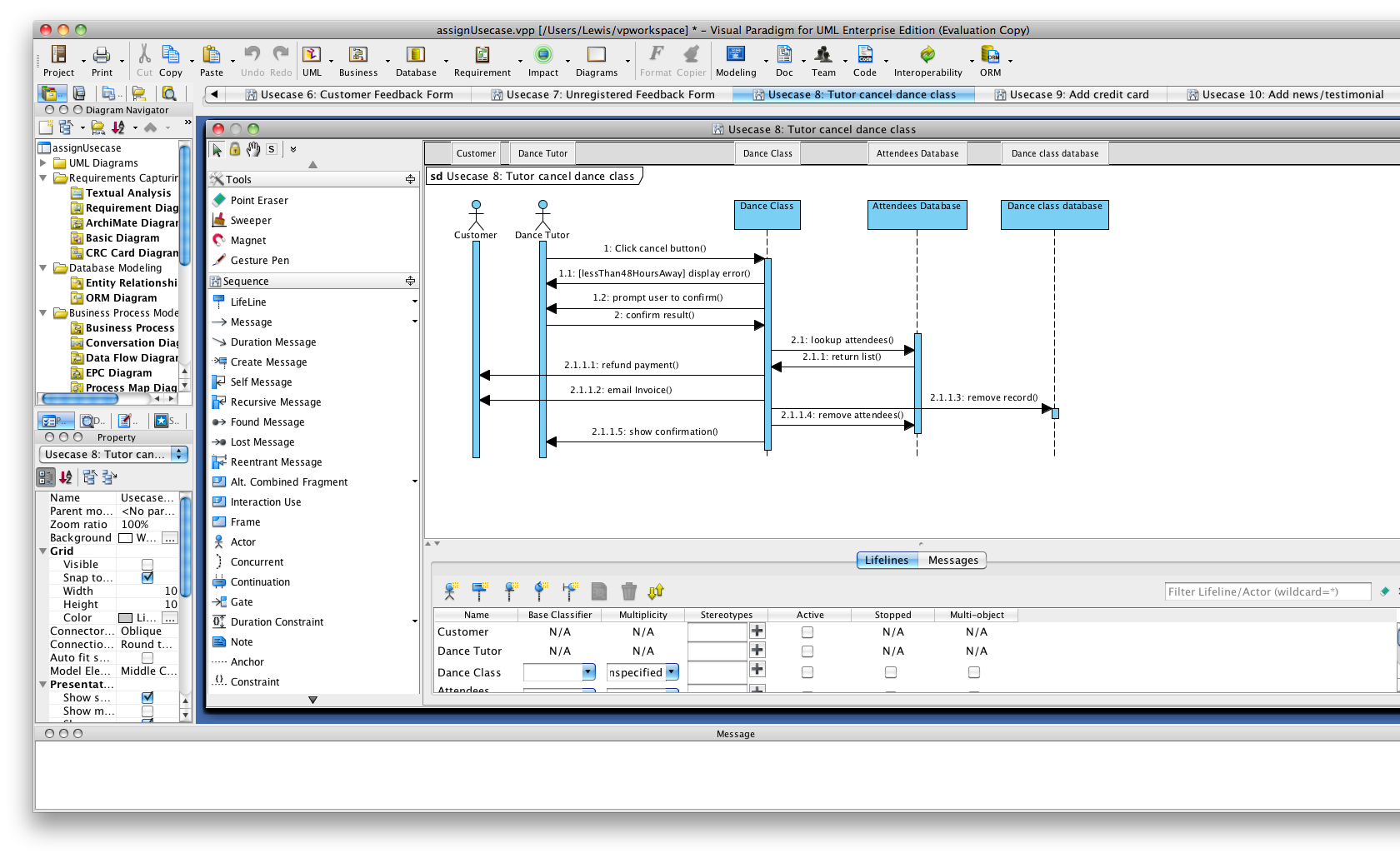
### 3.2.6 Use Case 6: Customer feedback form



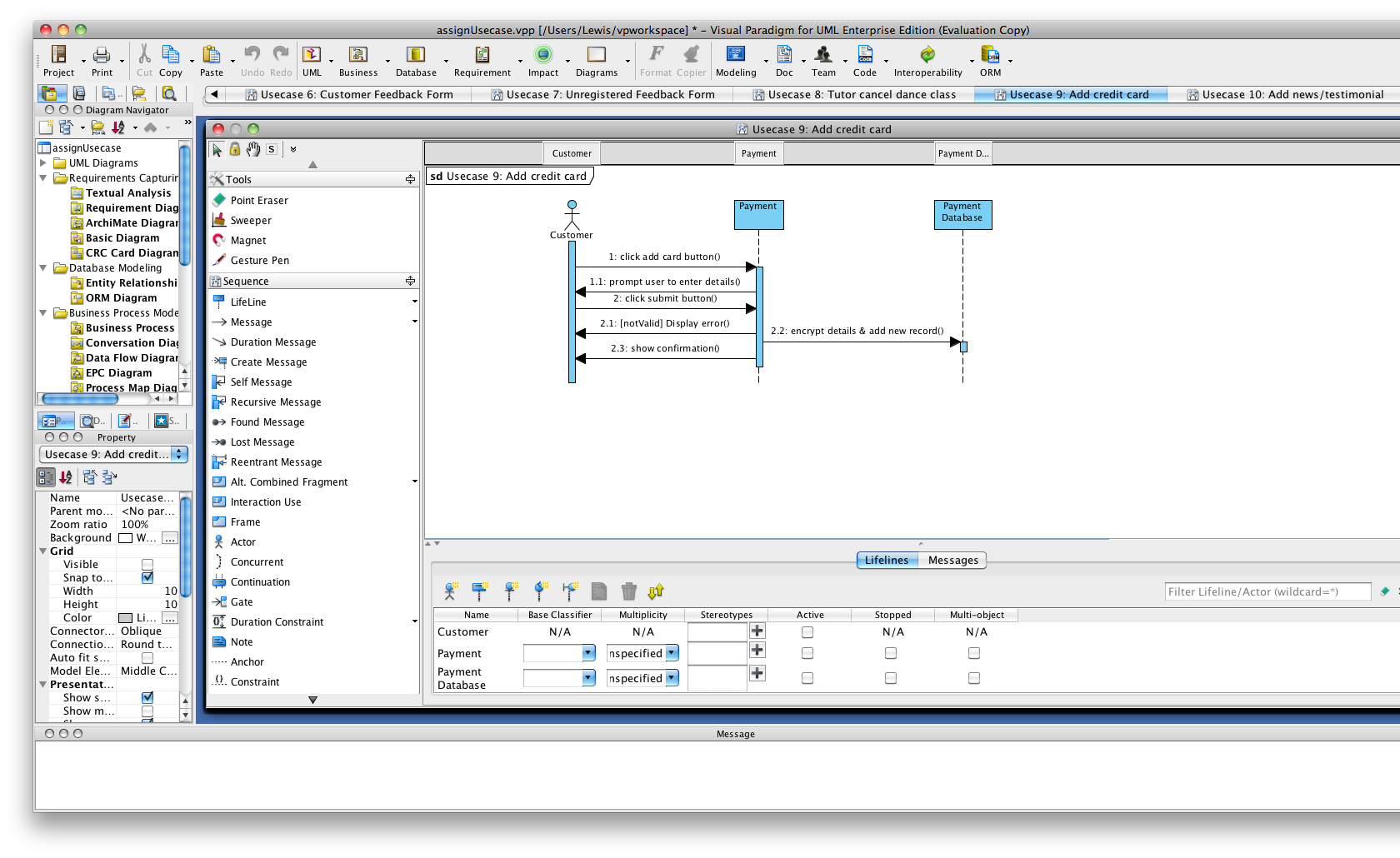
### 3.2.7 Use Case 7: Unregister user feedback form



### 3.2.8 Use Case 8: Tutor cancel dance class



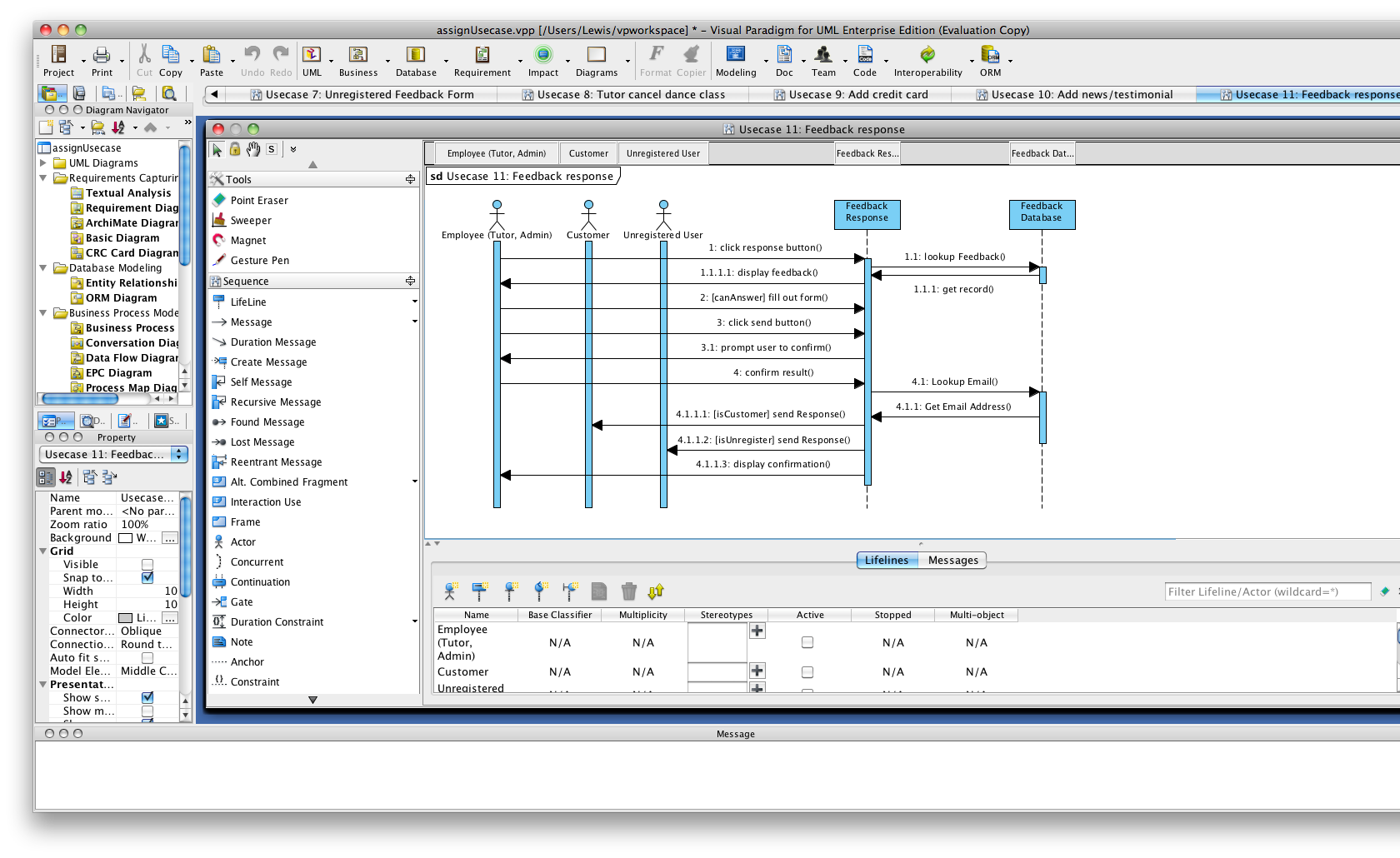
### 3.2.9 Use Case 9: Add credit card



### 3.2.10 Use Case 10: Add news/testimonials



### 3.2.11 Use Case 11: Feedback response



# 4.0 User interface design

## 4.1 Homepage - Overview

## Description: Macintosh HD:Users:Lewis:Pictures:Adv Systems:homepage.jpg

## 4.2 News page - Overview

## Description: Macintosh HD:Users:Lewis:Pictures:Adv Systems:newspage.jpg

## 4.3 Classes page – Monthly overview

## Description: Macintosh HD:Users:Lewis:Pictures:Adv Systems:classesmonthcustomer.jpg

## 4.4 Classes page – Day overview

## Description: Macintosh HD:Users:Lewis:Pictures:Adv Systems:classesdaycustomer.jpg

## 4.5 Classes page – Event overview

## Description: Macintosh HD:Users:Lewis:Pictures:Adv Systems:classeseventcustomer.jpg

## 4.6 Classes page – Event payment overview

## Description: Macintosh HD:Users:Lewis:Pictures:Adv Systems:classespaymentcustomer.jpg

## 4.7 Contact us page – Customer overview

## Description: Macintosh HD:Users:Lewis:Pictures:Adv Systems:contactuscustomer.jpg

## 4.8 Contact us page – Unregistered user overview

## Description: Macintosh HD:Users:Lewis:Pictures:Adv Systems:contactusunregr.jpg

## 4.9 Admin – Add news/testimonial page overview

## Description: Macintosh HD:Users:Lewis:Pictures:Adv Systems:addnewsadim.jpg

# 6.0 References

Zieliński, K., & Szmuc, T. (2006). *Software engineering: evolution and emerging technologies.* Amsterdam: IOS Press.

Moreno, N., Fraternalli, P., & Vallecillo, A. (2006). A UML 2.0 Proﬁle for WebML Modeling. *Proceedings of the 2nd Model-Driven Web Engineering Workshop* (pp. 1-10). New York: ACM Press.

Agarwal, R., De, P., Sinha, A. P., & Tanniru, M. (2000). On the usability of OO representations. *Communications of the ACM* , 83-89.

Birrell, N. D., & Ould, M. A. (1988). *A Practical Handbook for Software Development.* Cambridge: Cambridge University Press.

Geddis, S. (2008). *Generating Websites from Model.* University of Manchester, School of Computer Science. Manchester: University of Manchester.

Hunt, J. (2000). *The unified process for practitioners: object-oriented design, UML and Java.* London: Springer-Verlag London Limited.

Kuruc, J., Dolog, P., & Bielikova. (2004). *Prototyping Navigation in Web-Based Information Systems Using WebML.* Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies. Bratislava: Slovak University of Technology.

Naveda, J. F., & Seidman, S. B. (2006). *IEEE Computer Society real world software engineering problems: a self-study guide for today's software professional.* New Jersey: John Wiley & Sons, Inc.

Schattkowsky, T., & Lohmann, M. (2002). Rapid Development of Modular Dynamic Web Sites using UML. *UML 2002 Conference* (pp. 336-350). Unknown: LNCS 2460.

Spiridopoulos, K., & Widen, E. (2002). *Modelling web applications WebML versus UML.* Blekinge Institue of Technology, Computer Science and Software Engineering. Blekinge: Blekinge Institue of Technology.