**Rationale Design - Decisions**

ClubUML Application

3/26/2014

Addresses use case RM2

# Overview

In Software Engineering, Rationale Decision making is defined as a method for systematically selecting among possible choices that is based on reason and facts. In a rational decision making process, a Software Architect or the Teal Lead will often employ a series of analytical steps to review relevant facts, observations and possible outcomes before choosing a particular course of action. With the addition of the Rationale entity, Software Architect or Team Lead can use the aggregate rationales provided by the team to make more informed decisions while choosing between UML diagrams.

ClubUML Application aims at providing optimal rationale decision making process using the rationales provided in the process of promoting a UML diagram. The features provided by the tool include:

* Review the list of available decisions for a select project.
* Update a decision for a selected project for a select project.
* Create a new Decision for a select project.

# Core Features

The following list offers a brief outline and description of the main features and functionalities of the Decision.

**Display Decision**

* Displays all the following:
  + The Diagram and the Diagram name selected by the team lead for the project.
  + List of aggregated Rationale or Rationales leading up to the decision making.

**Create Decision**

* Create decision functionality allows the team leads to first pick a Diagram from currently uploaded diagrams and select best rationale from the list of rationales.

**Update Decision**

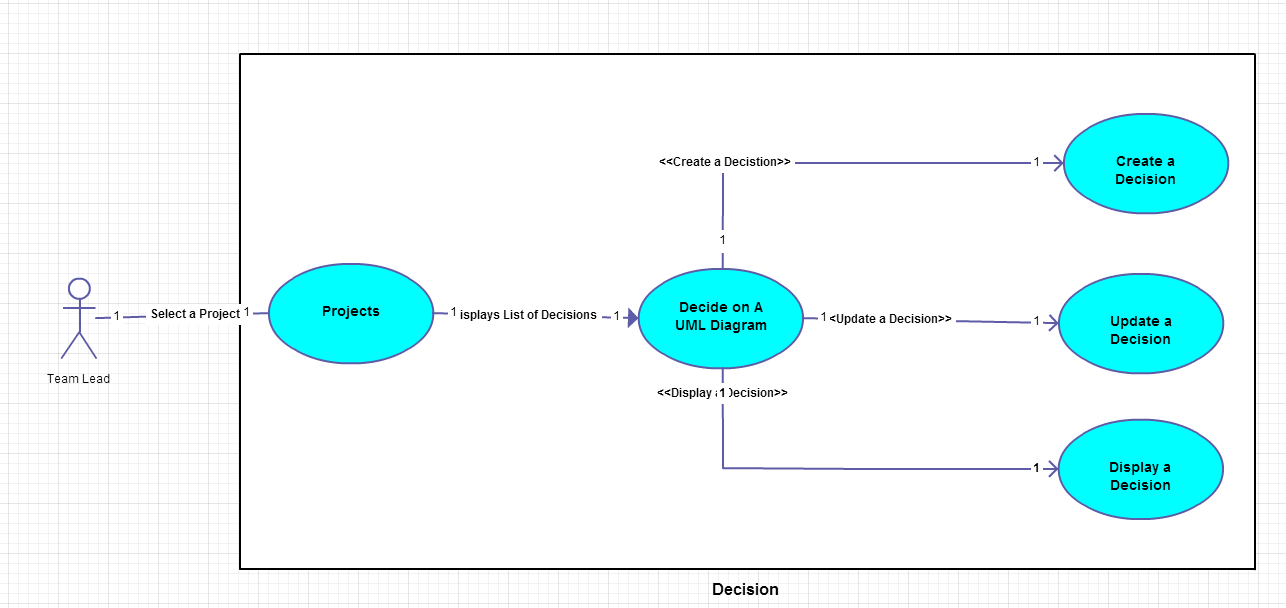
* The update decision allows the team leads to update the following:
  + Change the selected Diagram for the Decision and select a new diagram from the currently uploaded diagrams.
  + Change the rationale or rationales selected for the currently selected Diagram.
* Pick a diagram from the currently uploaded diagrams (excluding the currently selected decision diagram).

# Use Case

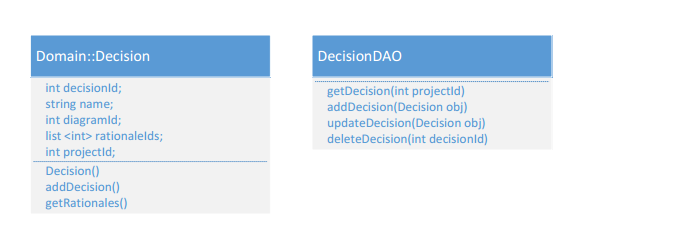
|  |  |
| --- | --- |
| Use Case Element | Description |
| Use Case Number | RM2 |
| Application | ClubUML Application |
| Use Case Name | Select Decision Diagram |
| Use Case Description | The team lead wishes to replace (change decision for) a diagram as the ground truth that will reflect current development goals and intentions. |
| Primary Actor | Team Lead |
| Precondition | There are currently decision diagrams and alternative diagrams uploaded to the system. The team lead has logged in and navigated to the project tab. |
| Trigger | The team lead clicks on a project link to the project for which they wish to select a decision diagram. |
| Basic Flow | 1. The application directs the team lead to the project view[2]. 2. The project view shows a list of current decision diagrams. 3. The user selects one current decision diagram and clicks the UPDATE DECISION button. 4. The application presents an update decision dialog. 5. The update decision dialog prompts the user to pick a diagram from the currently uploaded diagrams (excluding the currently selected decision diagram). 6. The user selects a diagram as the new decision. 7. The decision dialog prompts the user to select [none or more] rationales from the existing rationales associated with the newly selected diagram. 8. The user selects all relevant rationales and clicks OK. 9. The application saves the new decision without deleting the old decision that was selected in step 3. The application instead saves a reference to the predecessor decision. 10. The application redirects to a project view page that shows the updated decision. |
| Alternate Flows | The most significant alternatives and exceptions  If the decision diagram is new and not replacing an old decision diagram,   1. In the project view, the team lead clicks the NEW DECISION button. 2. The application shows a NEW DECISION dialog. 3. The new decision dialog prompts the user to pick a diagram from the currently uploaded diagrams 4. See basic flow steps 6-8 5. The application saves the decision without a predecessor decision and redirects to a project view page that shows the new decision.   If the user clicks the CANCEL button during any time in which a decision dialog is open,   1. The application closes the decision dialog. 2. No changes are made to the project view screen or the underlying database.   If in step 3, the user selects multiple decision diagrams to update,   1. 1. The application opens an error dialog directing the user to select a single decision to update. 2. The user clicks the error dialog OK button 3. The application removes the error dialog and continues to display the project view. |

# Design Details

## Flow Chart

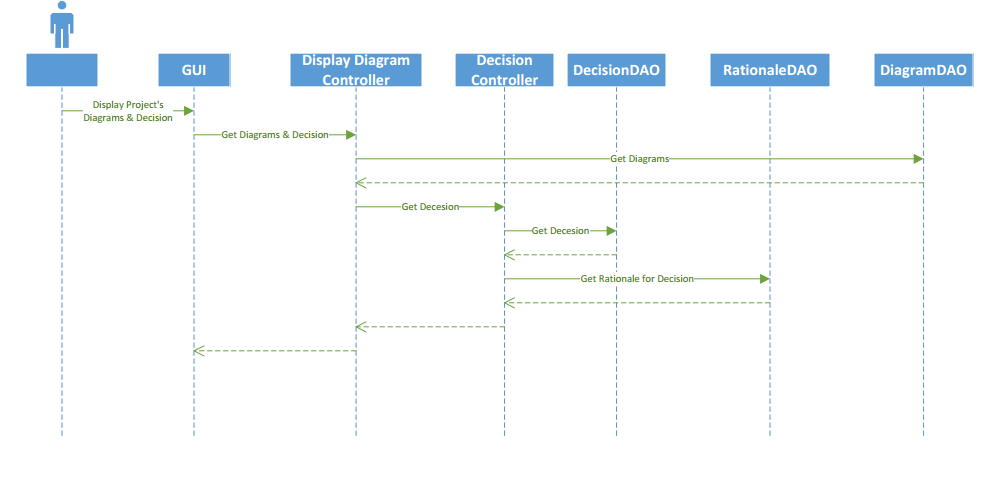


## Class Diagram

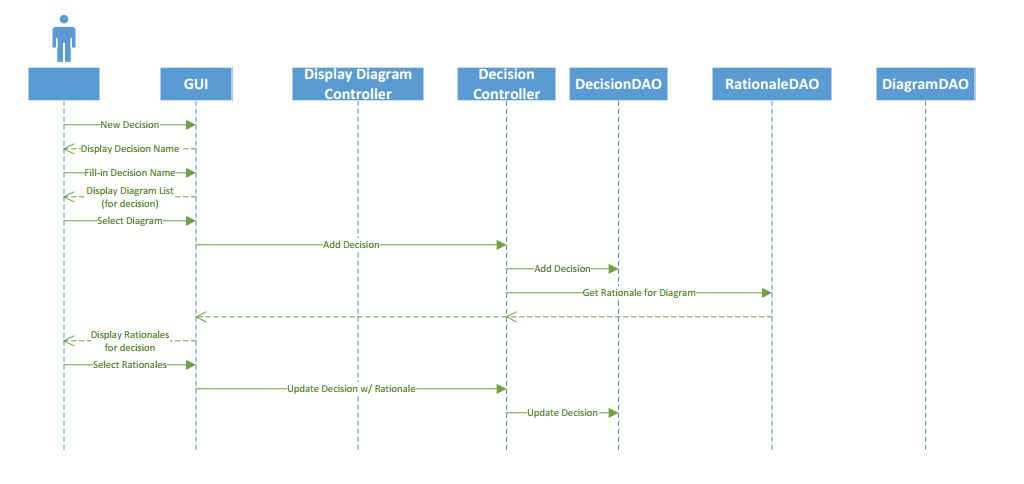


## Sequence Diagram

1. **Display New Diagram and Decision**



1. **Create Diagram and Decision**



## User Interface (UI Design

1. **Create Decision**



1. **Display Diagram and Decision**



## Java Components

## Decision Class

## Attributes

## Methods

## Exceptions

## DecisionDAO

## Attributes

## Methods

## Exceptions

## JavaScript Components

## Cascade Style Sheet Components

* 1. **Database Design**
     1. **Data Design**
     2. **Relationships**
     3. **SQL Scripts**
     4. **Data Access**