

Project Requirements

Felgenhauer, Maximilian Klose, Anthony
Marten, Jameson

September, 24, 2012

1 Introduction

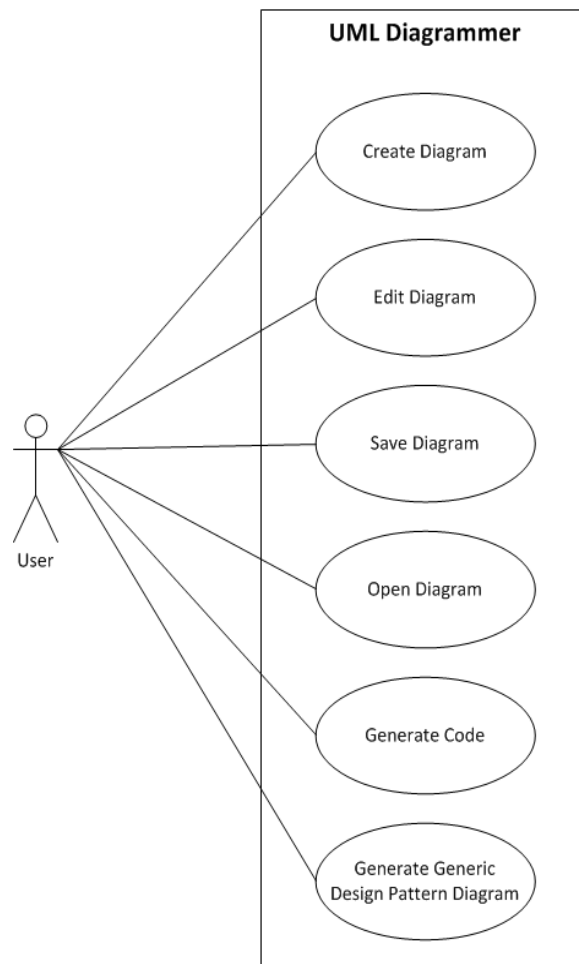
1. Purpose of the System
The purpose of this system is to facilitate writing Java code, by modeling class structures in UML.
2. Scope of the System
This system is for Java programmers and program designers.
3. The success criteria of the project
To create a functioning program which can diagram UML, and translate UML into java code.

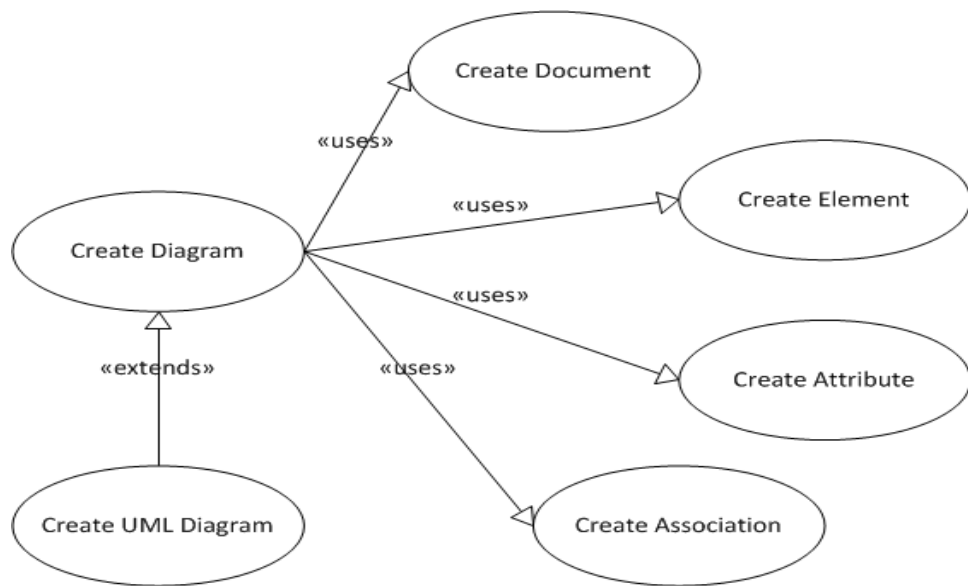
2 Proposed System

1. Functional Requirements
 - A user must be able to create and edit UML diagrams.
 - A user must be able to read and write diagrams to a file.
 - A user must be able to generate Java code from UML diagram.
 - A user must be able to make UML templates.
2. Nonfunctional Requirements
 - (a) Usability
 - This application should look native to the windows or mac based on used operating system.
 - There should be a maximum of 2 step to create any gui element.

- Program will have a Single Document Interface.
- (b) Reliability
- Use report instead of crashing program.
- (c) Performance
- Should take the User no longer than 3 seconds to save a document
 - Adding Gui elements should be instantaneous

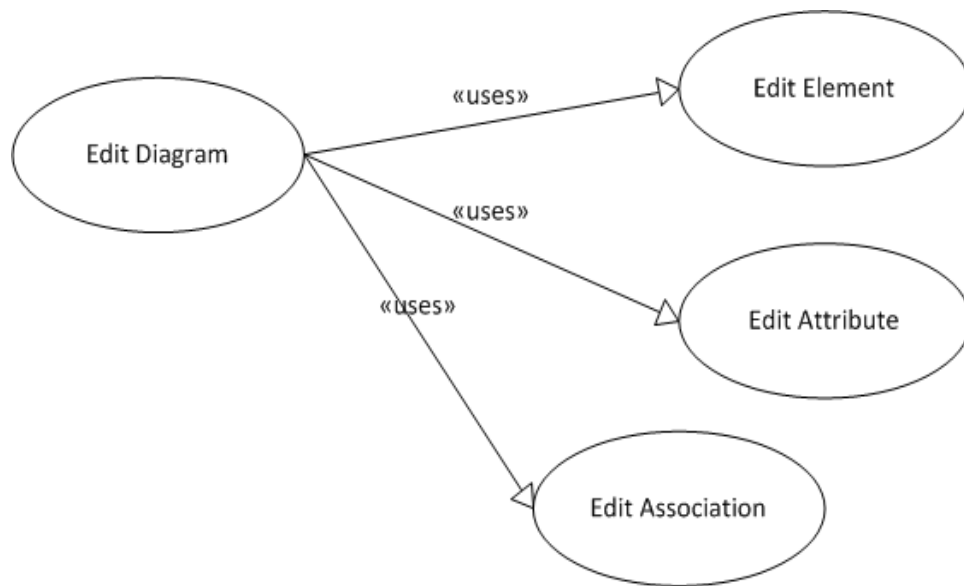
3 Use Case Model





1. Create UML Class Elements

- (a) Priority level High
- (b) Participating Actors
 - User
 - System
- (c) Flow of Events
 - i. User clicks somewhere on the document
 - ii. System creates UML Class at the point on the screen the user clicked
- (d) Entry Conditions
 - User presses Edit/Create/UML Class
 - User presses Right Click/Create/UML Class
- (e) Exit Conditions
 - UML Class is created



2. Edit UML Class Elements

(a) Priority Level Medium

(b) Participating Actors

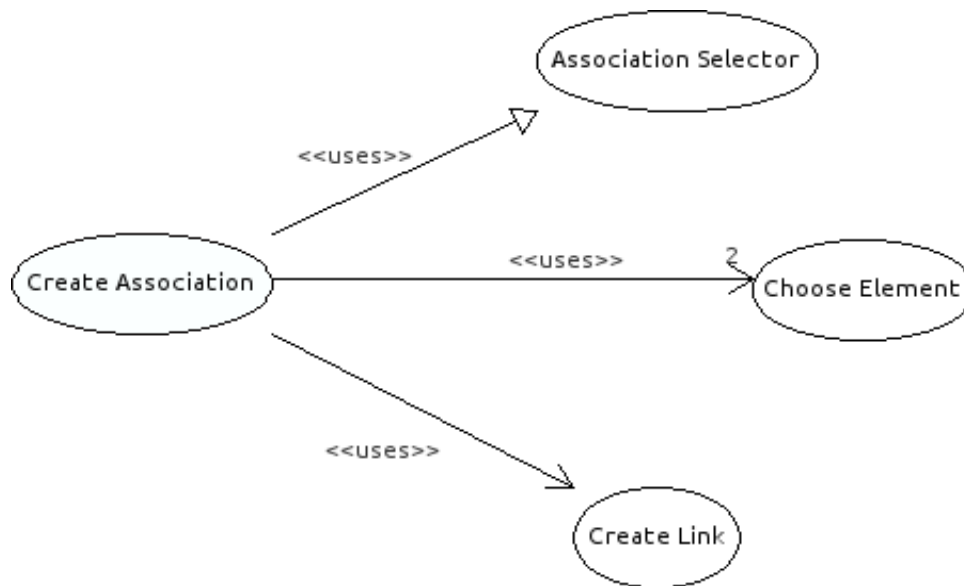
- User
- System

(c) Flow of Events

i. System prompts User with a Edit Element Window

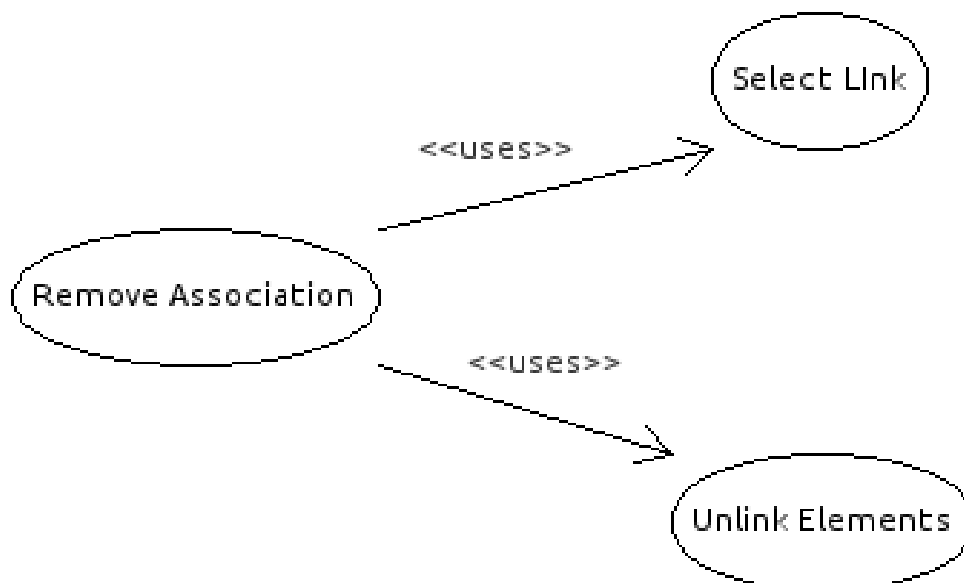
(d) Entry Conditions

- User right click Element/edit/add attribute
- User clicks Edit/Add/attribute



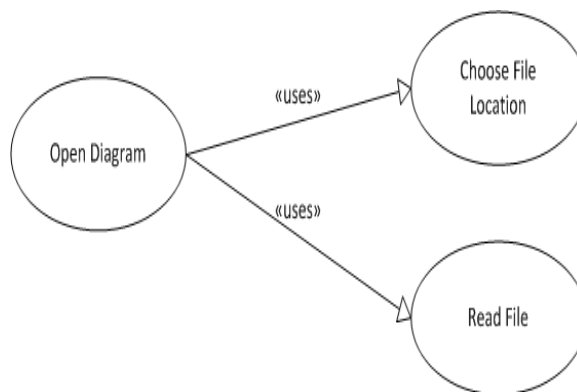
3. Create Association

- (a) Priority Level High
- (b) Participating Actors
 - User
 - System
- (c) Flow of Events
 - i. User will select which type of association
 - ii. User will select one UML Element
 - iii. User will select a second UML Element that is not the first
 - iv. System creates a link between the two Elements
- (d) Entry Conditions
 - User Clicks Edit/create/Association
 - User Right clicks first Element
- (e) Exit Conditions
 - User cancels
 - Link has been created



4. Remove Association

- (a) Priority Level High
- (b) Participating Actors
 - User
 - System
- (c) Flow of Events
 - i. User will select an Association
 - ii. System will unlink the elements
- (d) Entry Conditions
 - User Clicks Edit/Association
- (e) Exit Conditions
 - Link has been created



5. Open File

(a) Priority level Medium

(b) Participating Actors

- User
- System

(c) Flow of Events

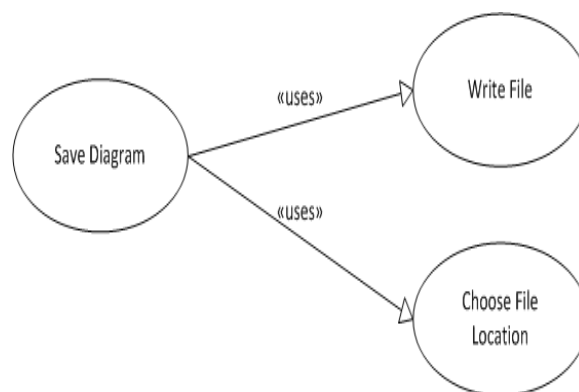
- System prompts User for a filename that exists on the System
- User picks filename
- System opens a new document with the contents of the file

(d) Entry Conditions

- User presses File/Open

(e) Exit Conditions

- User chooses filename
- User cancels



6. Write File

(a) Priority level Medium

(b) Participating Actors

- User
- System

(c) Flow of Events

- System prompts User for new or existing filename
- User enters a filename

iii. System save document into file on system

(d) Entry Conditions

- User presses File/Save
- User presses File/Save as. . .
- User presses Save button

(e) Exit Conditions

- User specifies a file
- User cancels

7. Generate Java Skeletons

(a) Priority level High

(b) Participating Actors

- User
- System

(c) Flow of Events

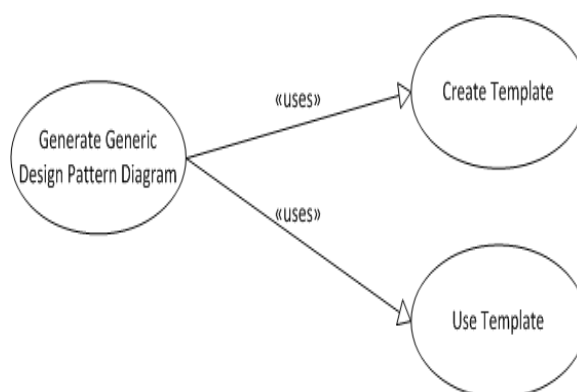
i. System creates java files

(d) Entry conditions

- User presses File/export/Java Project
- User presses Right click/export/Java Porject

(e) Exit conditions

- System is done writing java files



8. Create UML Template

(a) Priority level Low

(b) Participating Actors

- User
- System

(c) Flow of Events

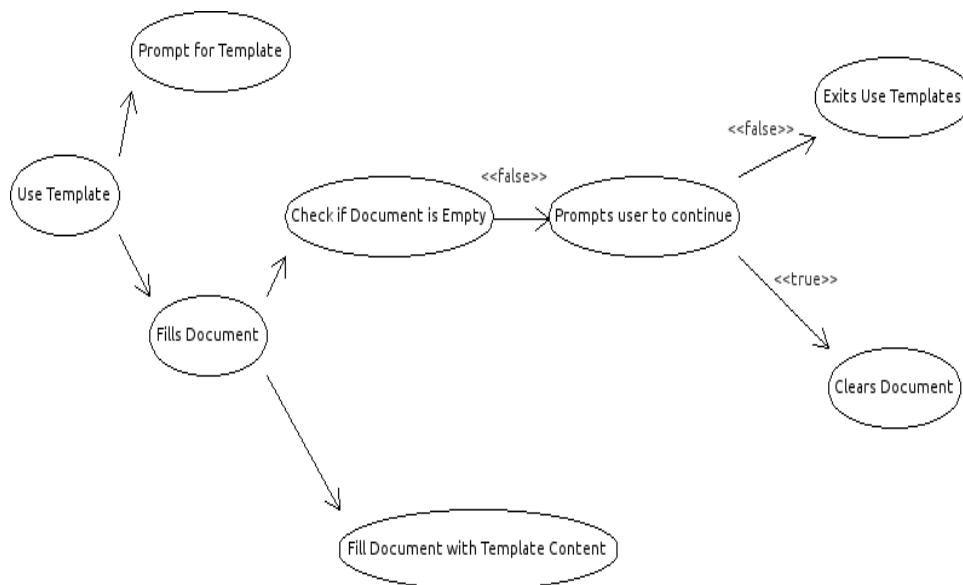
- System prompt user for name of the template
- system saves current document into templates file

(d) Entry conditions

- Document is not empty
- User presses File/templates/create

(e) Exit Conditions

- System finishes saving file



9. Use UML Template

(a) Priority level Low

(b) Participating Actors

- User
- System

(c) Flow of Events

- prompt user for which template to fill the document

- ii. if current document is not empty, system prompts the user if he/she wants to continue
- iii. if current document is not empty, clear it
- iv. fills current document with template

(d) Entry Conditions

- User presses File/templates/open

(e) Exit Conditions

- User decides to not continue
- System finishes loading template