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 modelling class structures in UML.
2. Scope of the System
   This system is intended 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 diagrams.
   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 the operating system used.
   - There should be a maximum of 2 steps to create any UML element.

- Program will have a Single Document Interface.

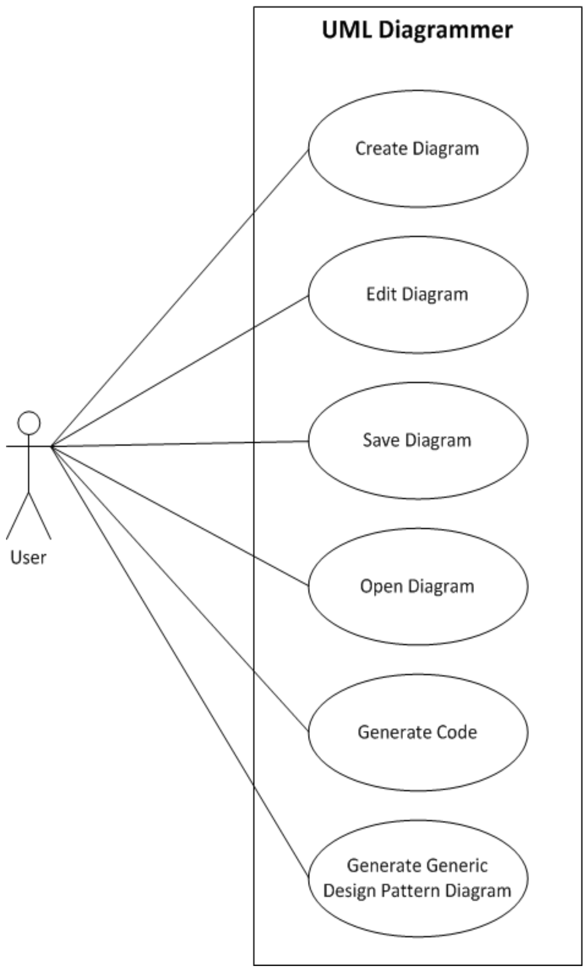
(b) Reliability

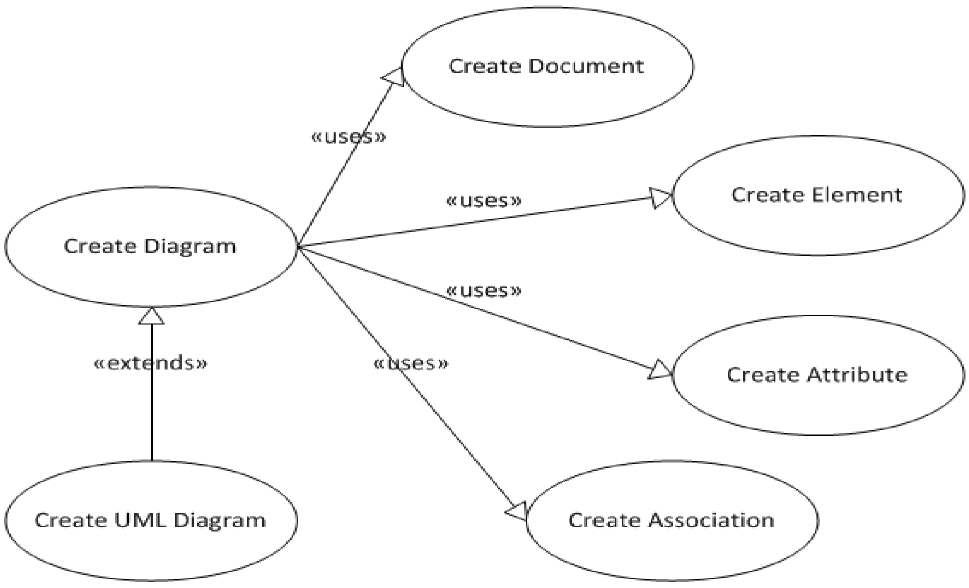
- The program will report any errors to the user instead of crashing, if possible.

(c) Performance

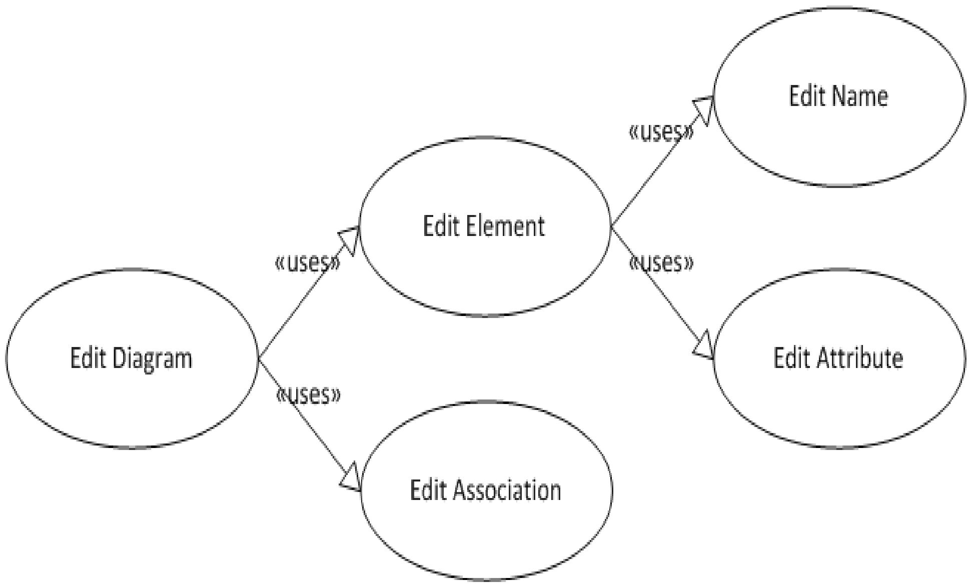
- Should take the User no longer than 3 seconds to save a document.
- Adding GUI elements should be instantaneous.

3 Use Case Model

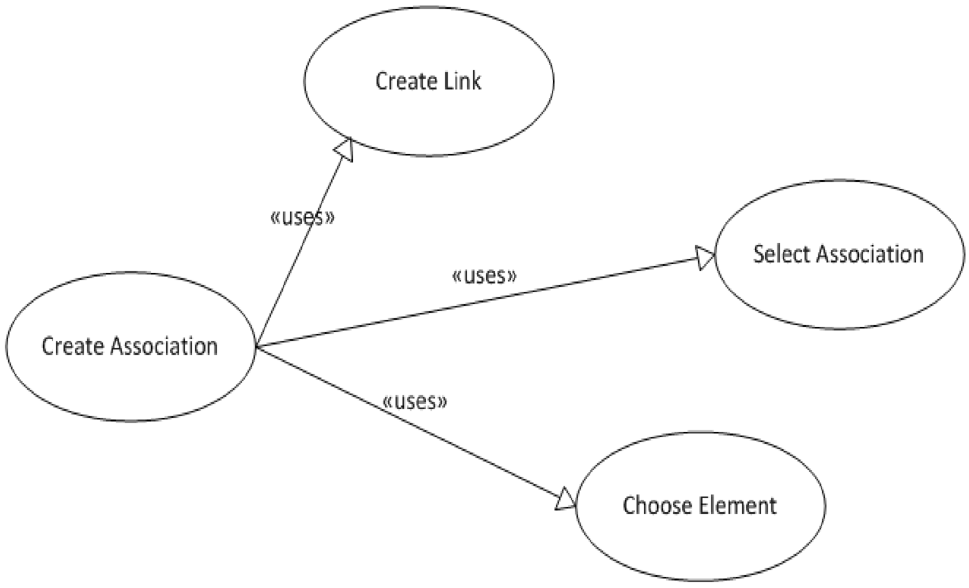




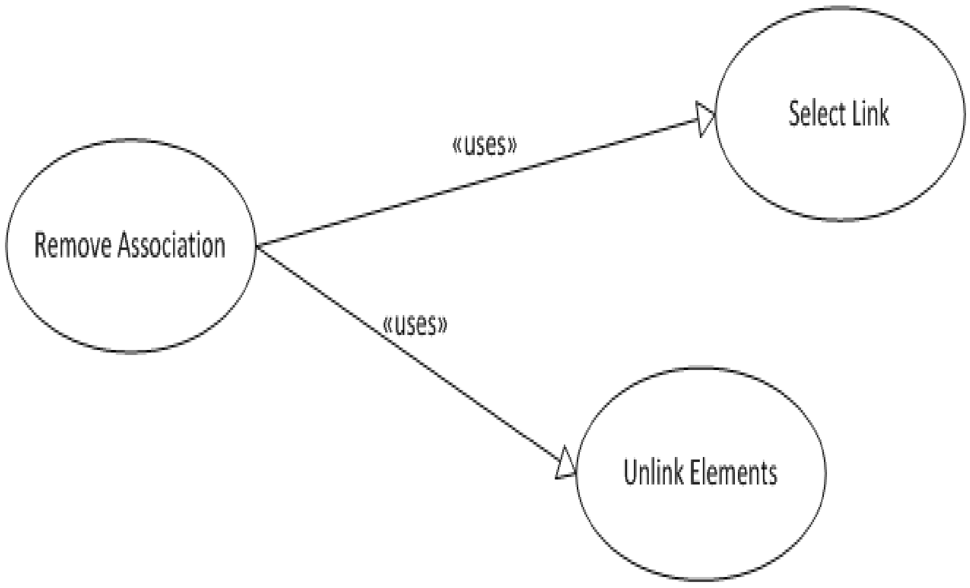
1. Create UML Class Elements
   (a) Priority level: High
   (b) Participating Actors
       - User
       - System
   (c) Flow of Events
       i. User clicks somewhere on the document
       ii. System creates UML Class at the point on the screen the user clicked
   (d) Entry Conditions
       - User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



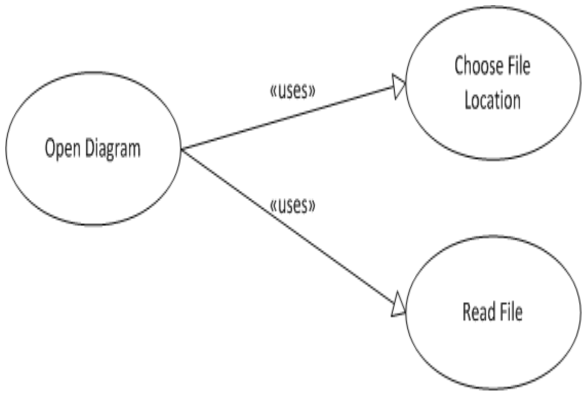
2. Edit UML Class Elements
   (a) Priority Level: Medium
   (b) Participating Actors
       - User
       - System
   (c) Flow of Events
       i. System prompts User with a Edit Element Window
   (d) Entry Conditions
       - User right click Element/Edit/Add attribute
       - User clicks Edit/Add/Attribute



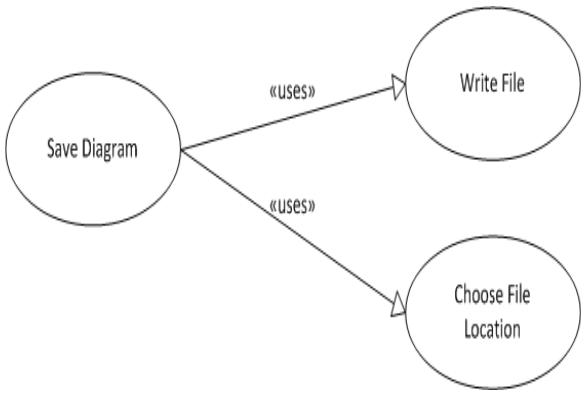
3. Create Association
    (a) Priority Level: High
    (b) Participating Actors
        - User
        - System
    (c) Flow of Events
        i. User will select which type of association
        ii. User will select one UML element
        iii. User will select a second UML element that is not the 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 selects an association
      ii. System unlinks the elements
   (d) Entry Conditions
      - User Clicks Edit/Association
   (e) Exit Conditions
      - Link has been removed



5. Open File
    (a) Priority Priority Level: Medium
    (b) Participating Actors
        - User
        - System
    (c) Flow of Events
        i. System prompts User for a filename that exists on the System
        ii. User picks filename
        iii. 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

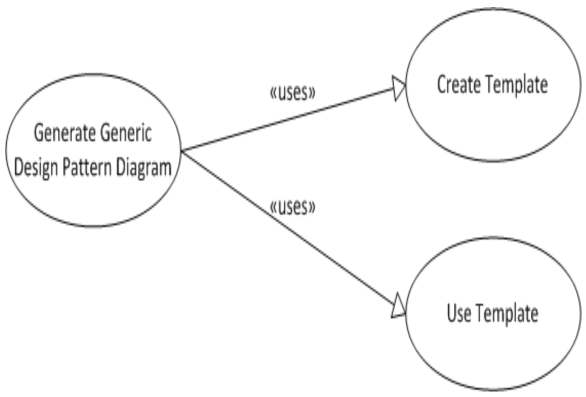


6. Write File
    (a) Priority level: Medium
    (b) Participating Actors
        - User
        - System
    (c) Flow of Events
        i. System prompts User for new or existing filename
        ii. User enters a filename

            iii. System saves document to a file on the 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



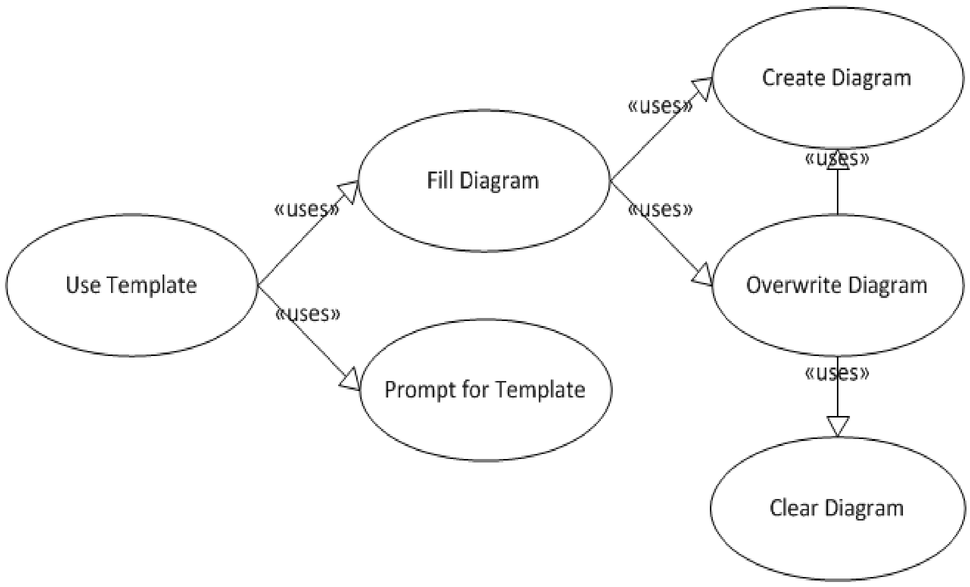
8. Create UML Template
   (a) Priority level: Low

(b) Participating Actors
   - User
   - System

(c) Flow of Events
   i. System prompt user for name of the template
   ii. 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) Piroity level: Low

(b) Participating Actors
   - User
   - System

(c) Flow of Events
   i. 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. Fill current document with template

(d) Entry Conditions

- User presses File/Templates/Open

(e) Exit Conditions

- User decides to not continue
- System finishes loading template