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

3 Use Case Model

1. Creating UML Class Elements
   (a) Priority level Now
   (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. Editing UML Class Elements
   (a) Priority Level red
   (b) Participating Actors
      - User
      - System
   (c) Flow of Events
      i. User right click Element/edit/{add element, remove attribute,edit element}
3. Open File
   (a) Priority level green
   (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

4. Write File

(a) Priority level Cute Bunny!

(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 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

5. Generate Java Skeletons

(a) Priority level OHHHHHH Shit!

(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

6. UML Template Creation

(a) Priority level blah

(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

7. UML Template Instantiation

(a) Piroity level Fuck it

(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. 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