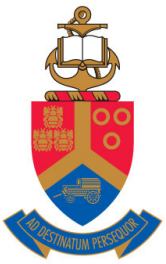


UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

COS 301 Main Project

Requirements and Design Specifications

ThinkTech

Group Members:

Goodness Adegbenro 13046412
Tshepiso Magagula 12274195
Hlavutelo Maluleke 12318109
Xoliswa Ntshingila 13410378
Lelethu Zazaza 13028023

Git repository link:

`https://github.com/COS301-ThinkTech/Flowchart-planning-and-simulation-tool`

Version 0.1

August 26, 2015

Contents

1 Vision and scope

1.1 Project background

The flowchart planning and simulation tool is a visual tool which aims to introduce programming logic to students by providing a platform to construct and execute flowcharts which depict basic programming structures whilst validating whether the flowchart components are valid .

1.2 Project vision

This project sets out to create an application for the planning and simulation of flowcharts, which can be used in an education setting for an introductory program logic course. The objective is to develop an application that can feasibly be used in such a practical setting. The application must be visual in nature, due to the visual nature of flowcharts. The application must facilitate exploration and experimentation, while providing clear feedback to the user as to how the program executes.

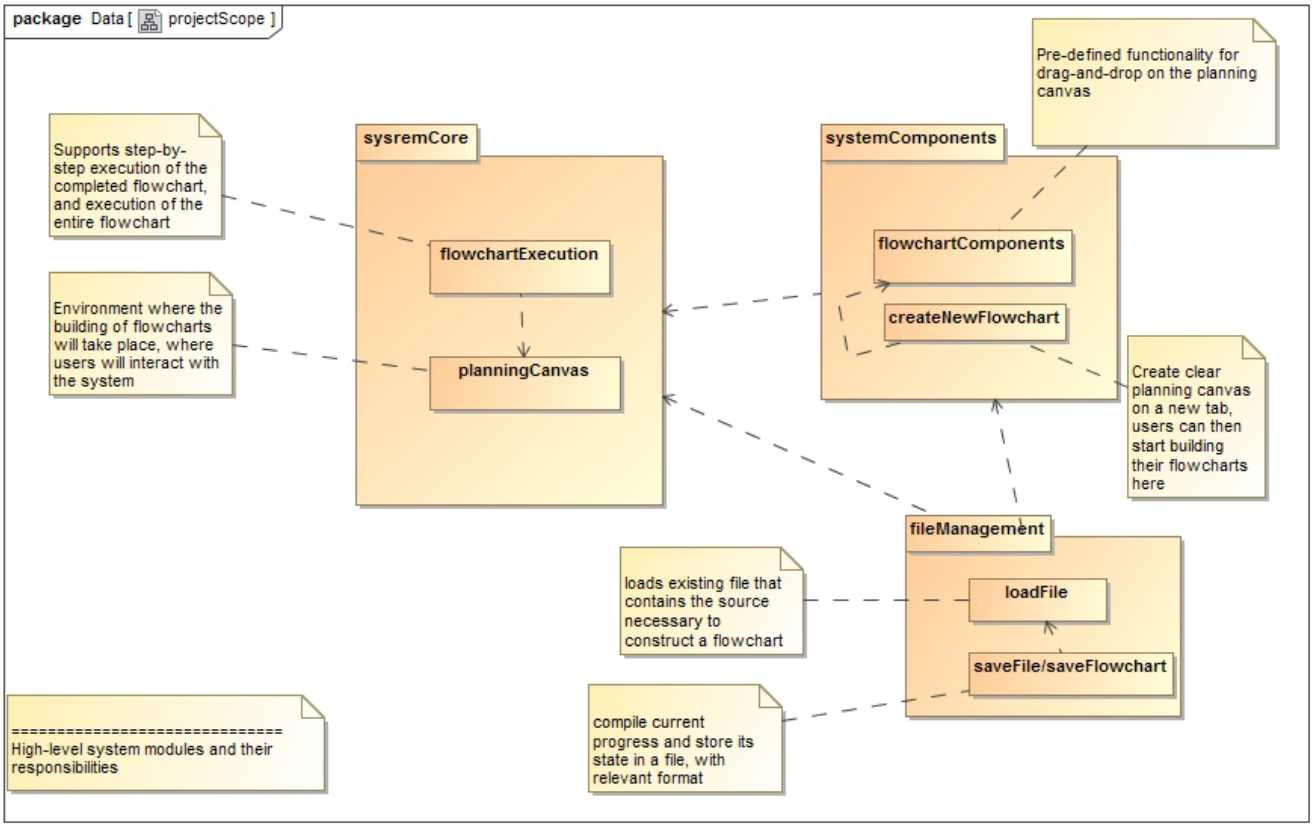
1.3 Project scope

The project consists of the following two components:

- A planning canvas, in which a flowchart can be built using an intuitive drag-and-drop editor. Flowchart components will be selected and dropped into the canvas, and connected into complete flowcharts. The system will also have to do error checking on the constructed flowcharts, so that (for example) multiple entry points into a program or certain flowchart components are not allowed.
- An execution system, in which a flowchart can be run from start to finish. The system should allow for one-click execution of the entire program, as well as step-by-step execution. At all stages during execution, the currently executing component should be highlighted, as well as the connection path being followed. The program's execution should be very visually apparent and appealing. The output of the flowchart's execution should also be apparent.

The following components are specifically excluded from the scope of the project:

- No executable program code generation will be required for this project.
- No complex design elements (such as user-defined component assemblies) are required. Only the basic components of standard flowcharts are necessary.



High level system modules and their responsibilities.

2 Use case prioritization

2.1 Critical

- createFlowchartProject
- deleteFlowchartProject

2.2 Important

- addFlowchartComponent
- editFlowchartComponent
- deleteFlowchartComponent
- saveFlowchart
- loadFlowchart
- runSimulationTool

2.3 Nice to have

- deleteFlowchartComponent(By dragging and dropping into bin)

3 Use cases

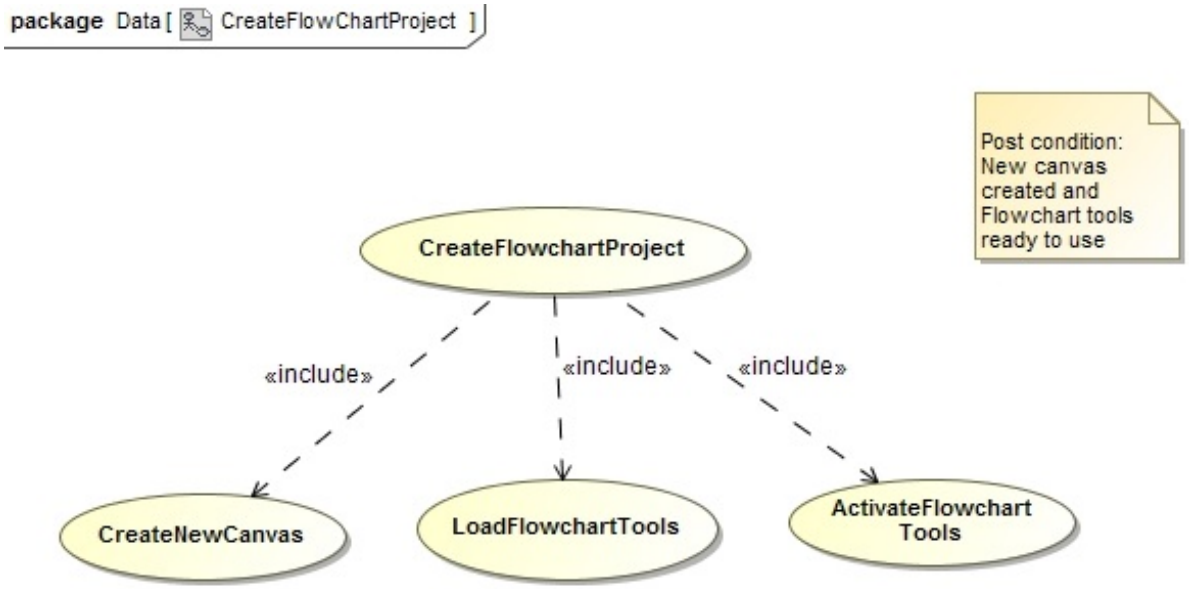
3.1 createFlowchartProject

Creates an environment to enable users to start building flowcharts.

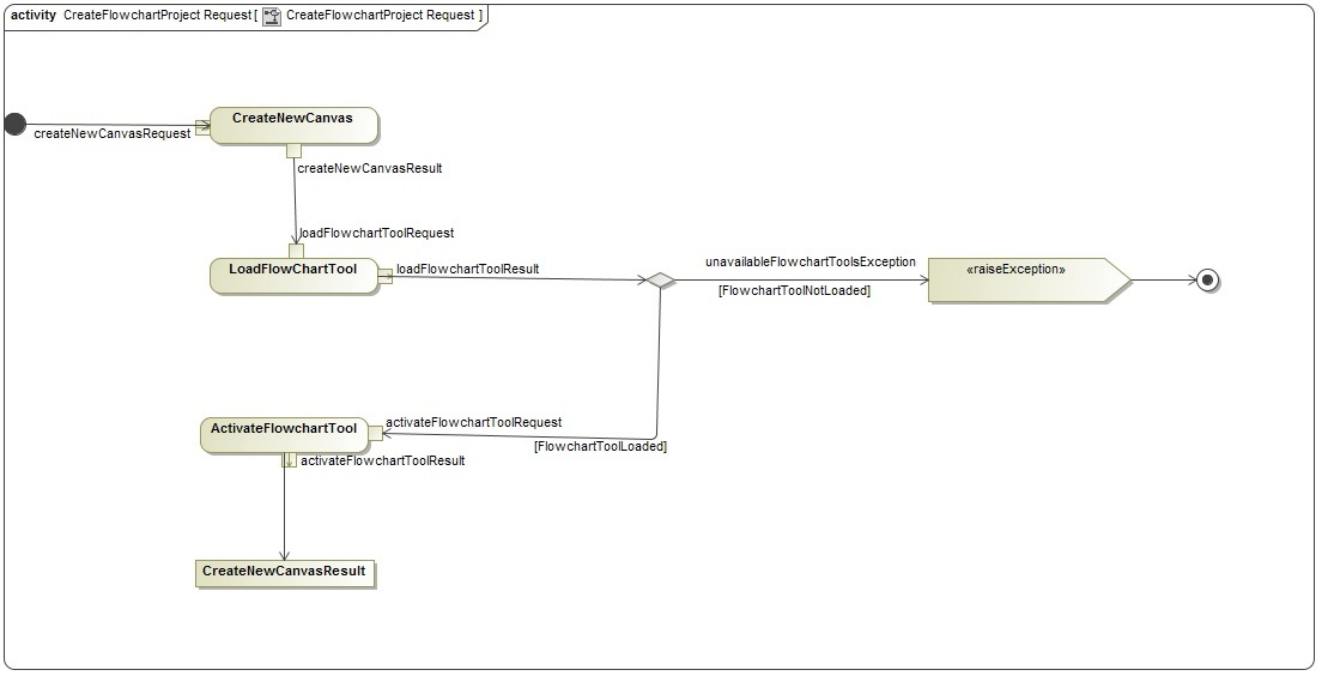
Pre Condition: Planning canvas must be blank.

Post Conditions: New canvas created, flowchart tools ready for use

3.1.1 Use case diagram



3.1.2 Activity diagram



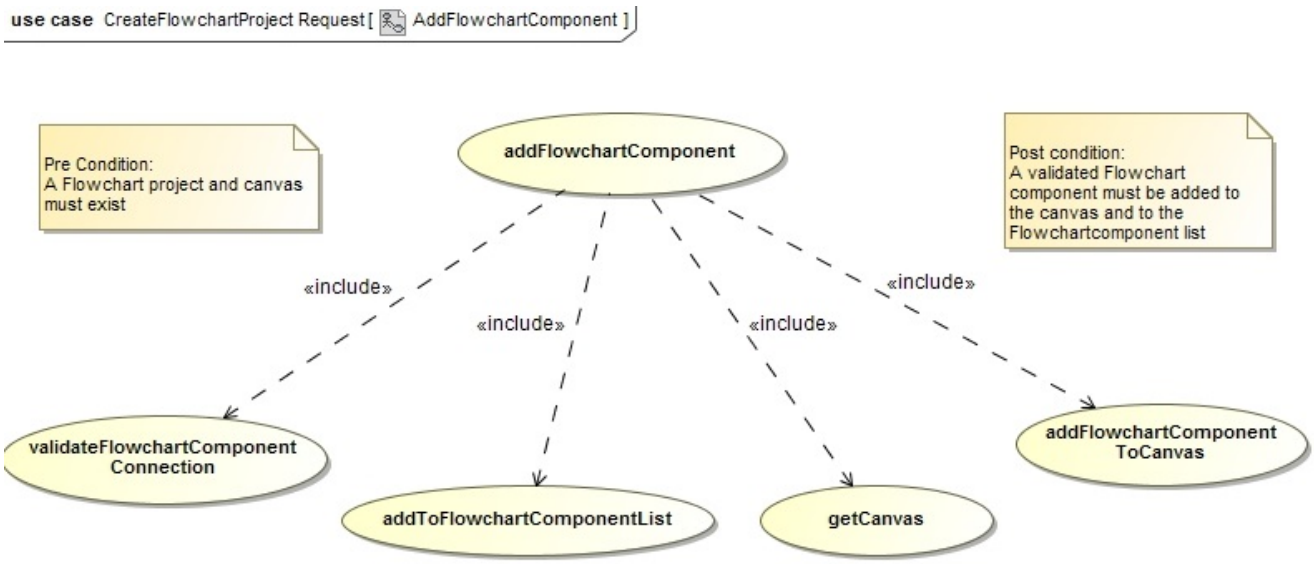
3.2 addFlowchartComponent

Provides users with functionality to drag and drop flowchart components into the planning canvas.

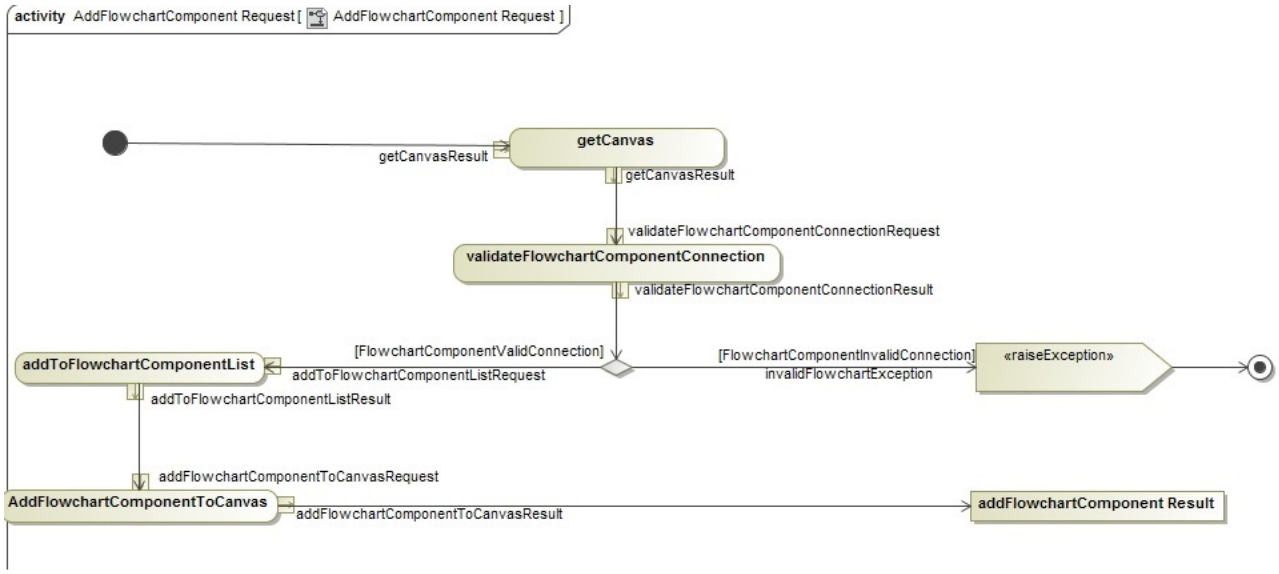
Pre Condition: Canvas is available.

Post Condition: Component has been added to flowchart.

3.2.1 Use case diagram



3.2.2 Activity diagram



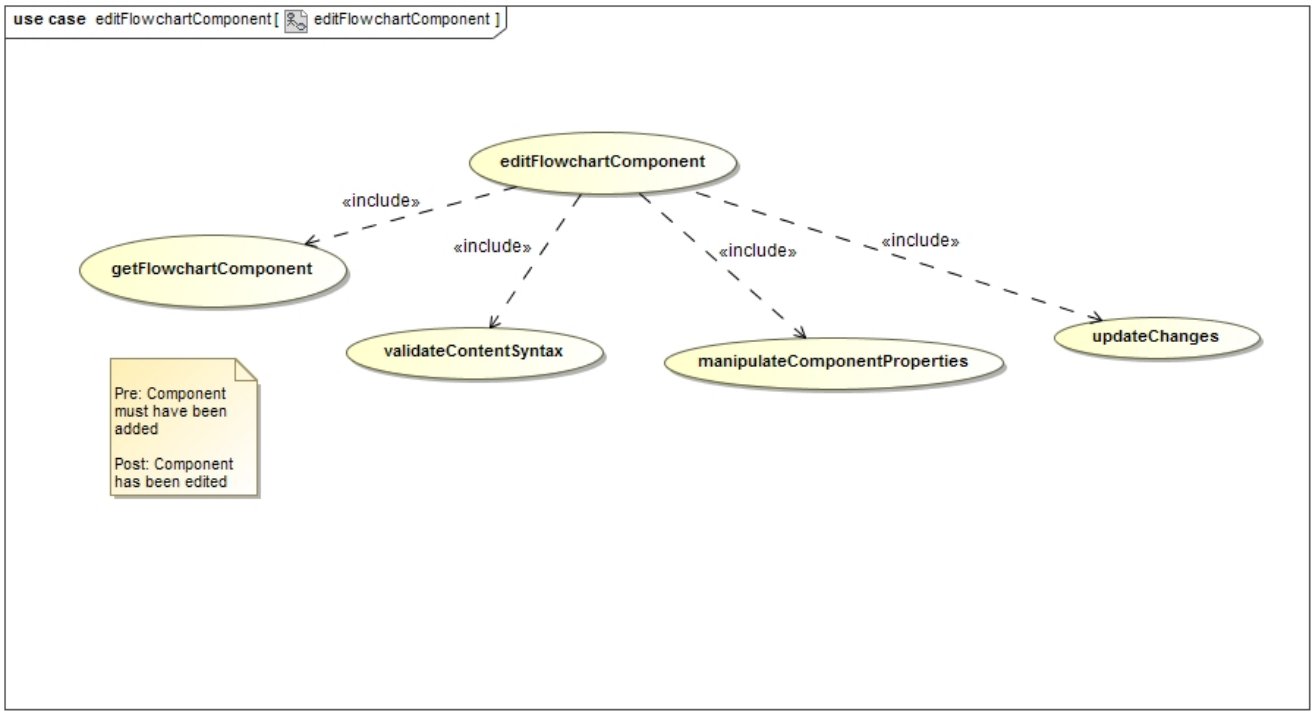
3.3 editFlowchartComponent

The editFlowchartComponent use case provides functionality for the user to edit each component of the flowchart on a canvas.

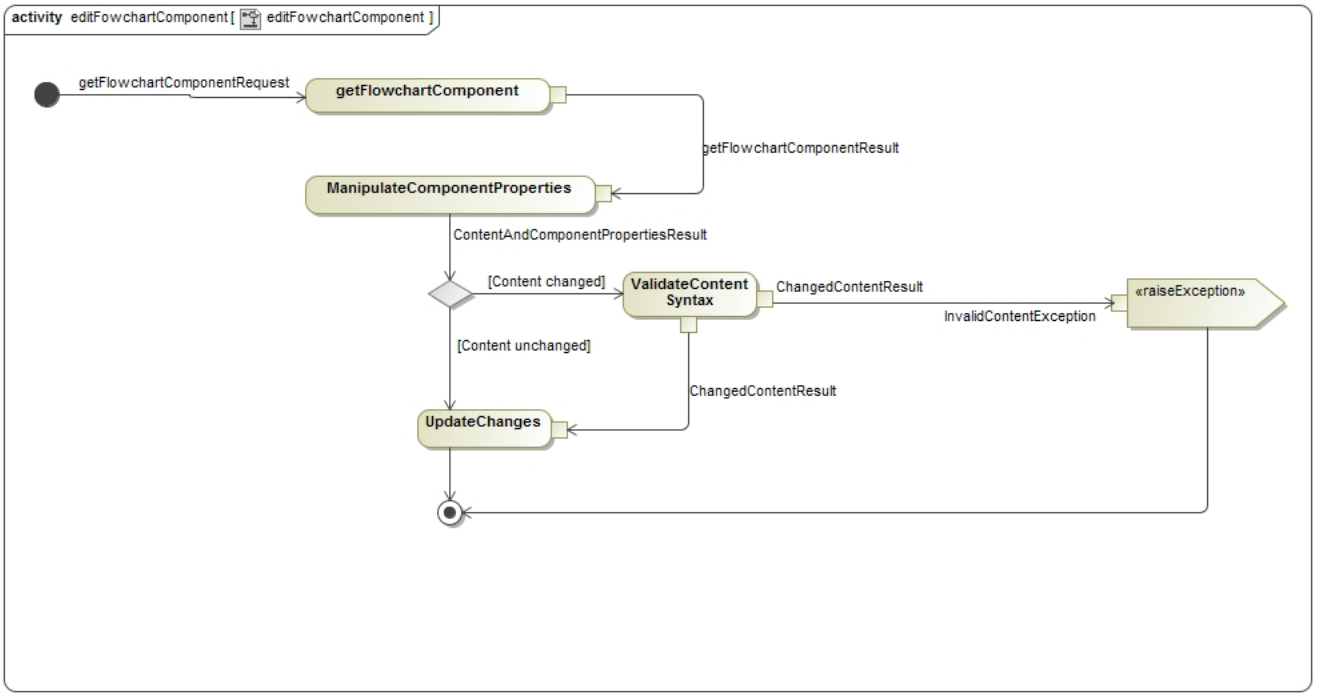
Pre Condition: Component must have been added

Post Condition: Component has been edited

3.3.1 Use case diagram



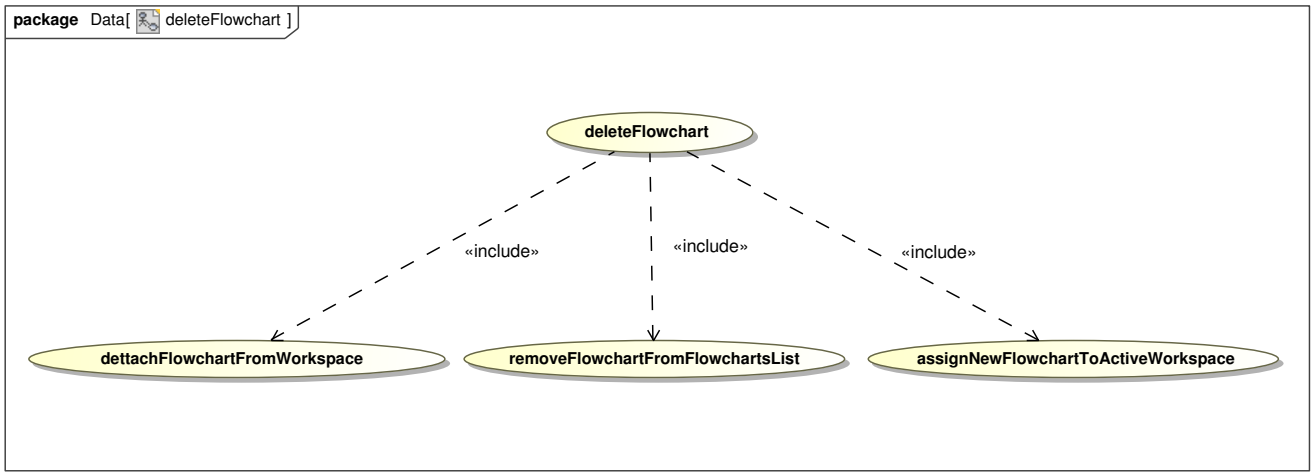
3.3.2 Activity diagram



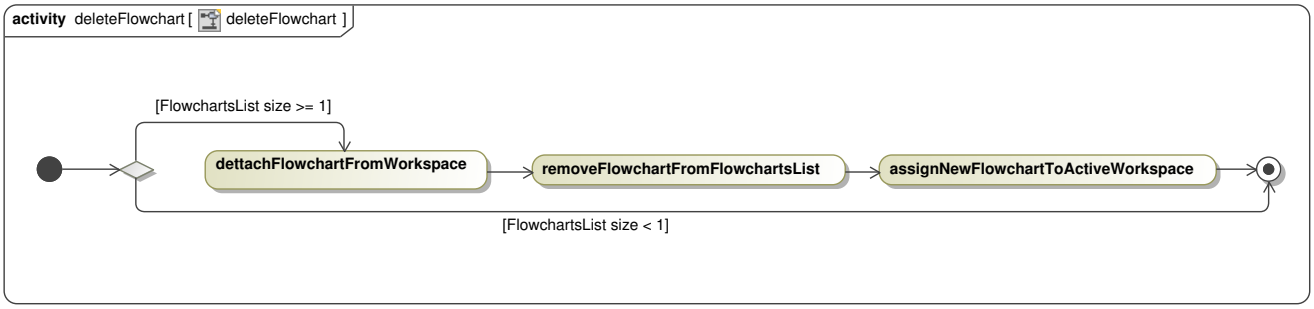
3.4 deleteFlowchartProject

The deleteFlowchartProject use case serves the purpose of removing a flowchart project.

3.4.1 Use case diagram



3.4.2 Activity diagram



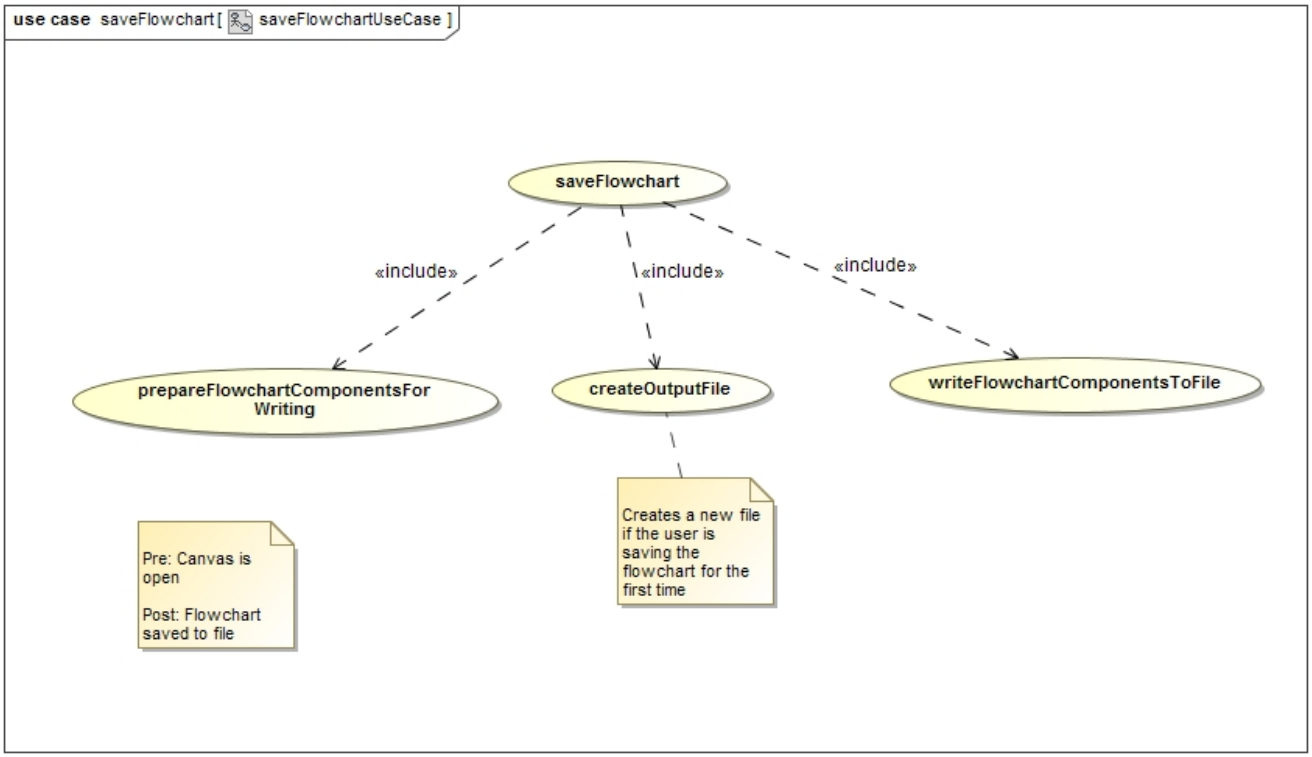
3.5 saveFlowchart

The saveFlowchart use case provides functionality for the user to save a flowchart.

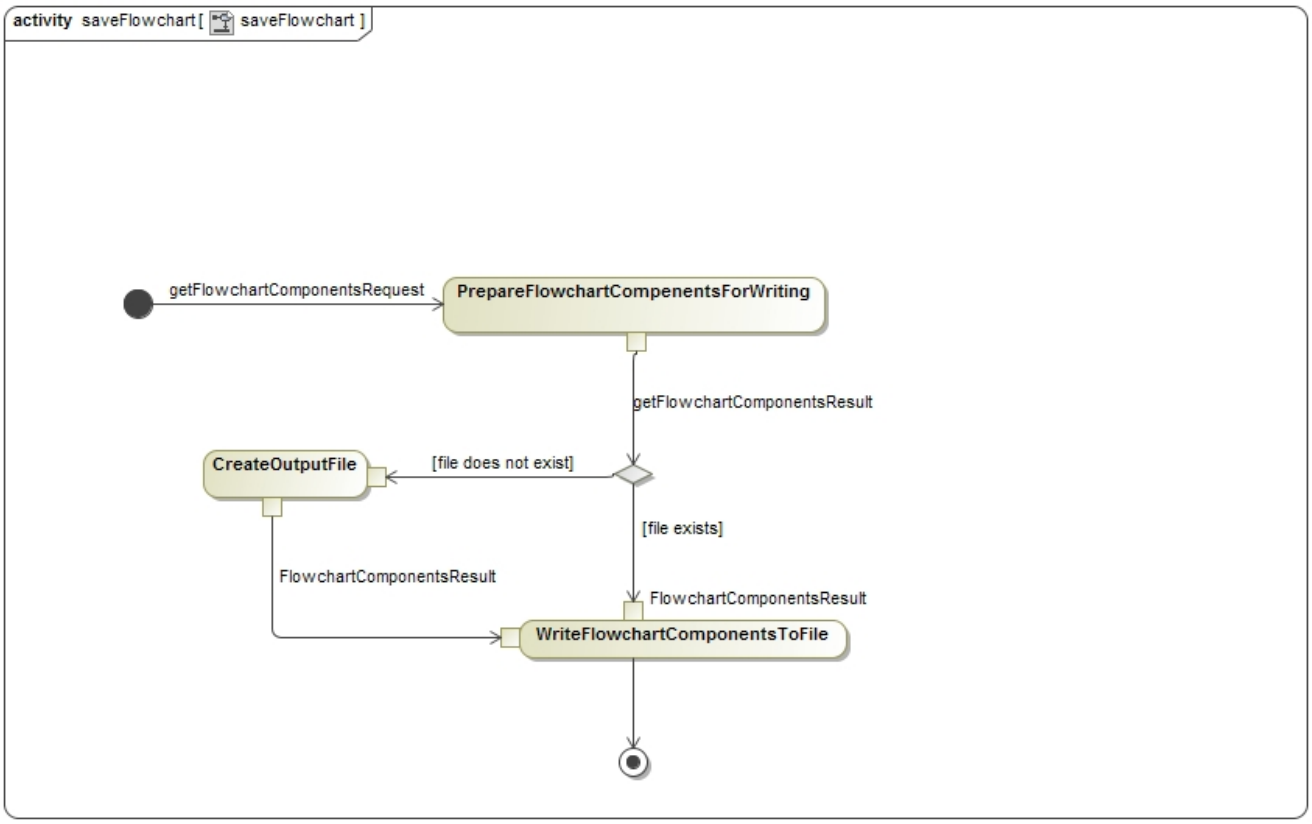
Pre Condition: Canvas is open

Post Condition: Flowchart has been saved to file

3.5.1 Use case diagram



3.5.2 Activity diagram

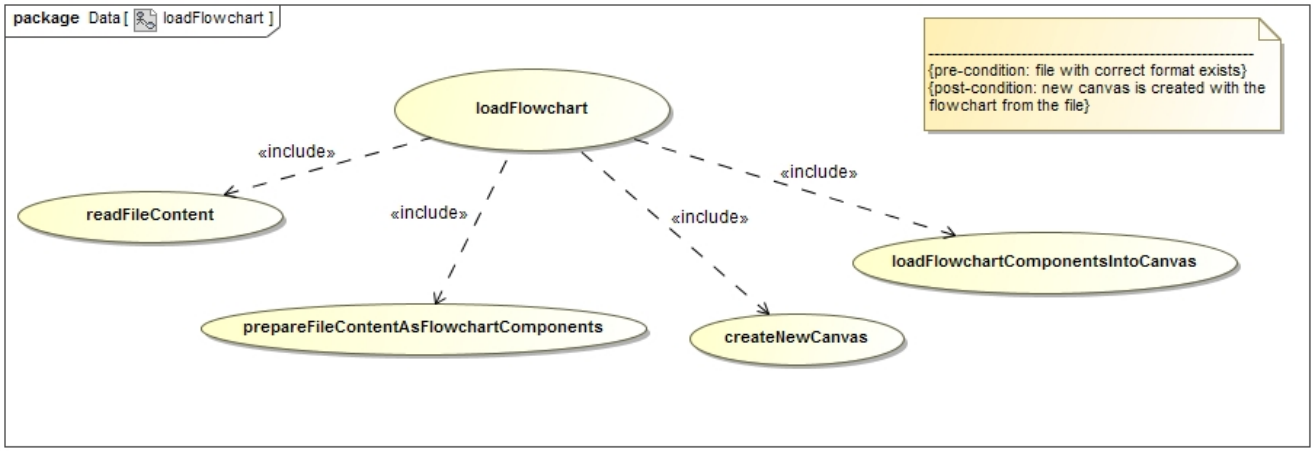


3.6 loadFlowchart

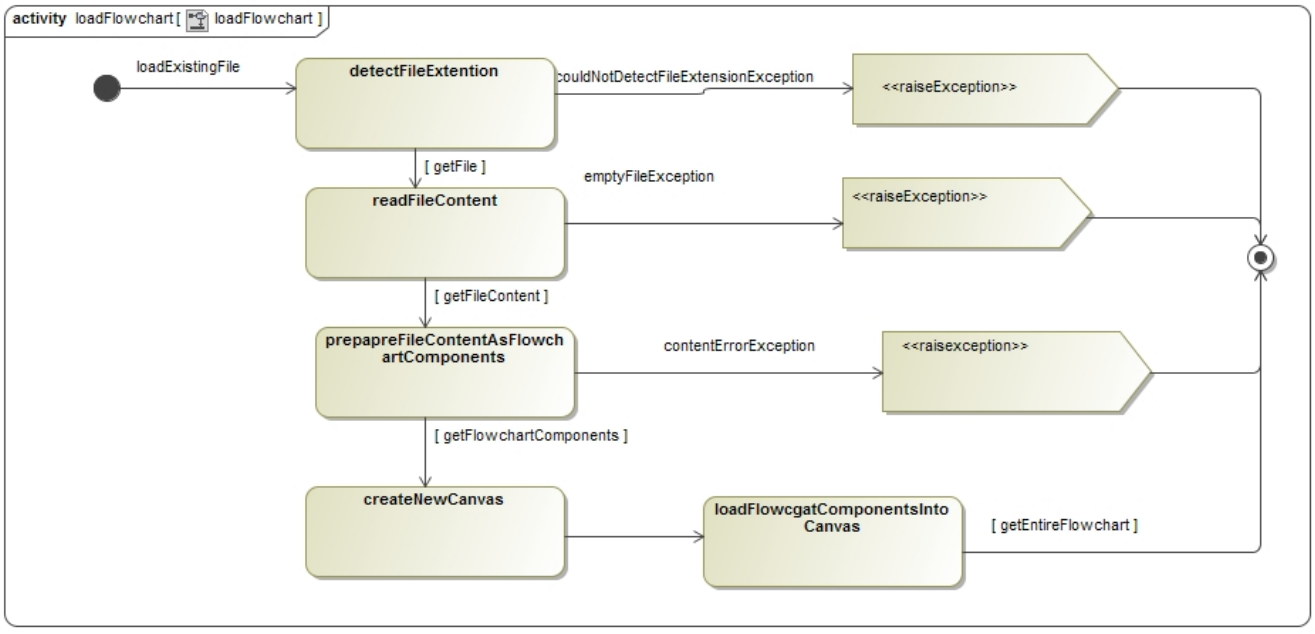
loadFlowchart: users load flowcharts from existing files. The file is editable, can be modified by the user.

Pre Condition: file with correct extension already exists.
Post Condition: open the file and make it editable in the canvas element.

3.6.1 Use case diagram



3.6.2 Activity diagram



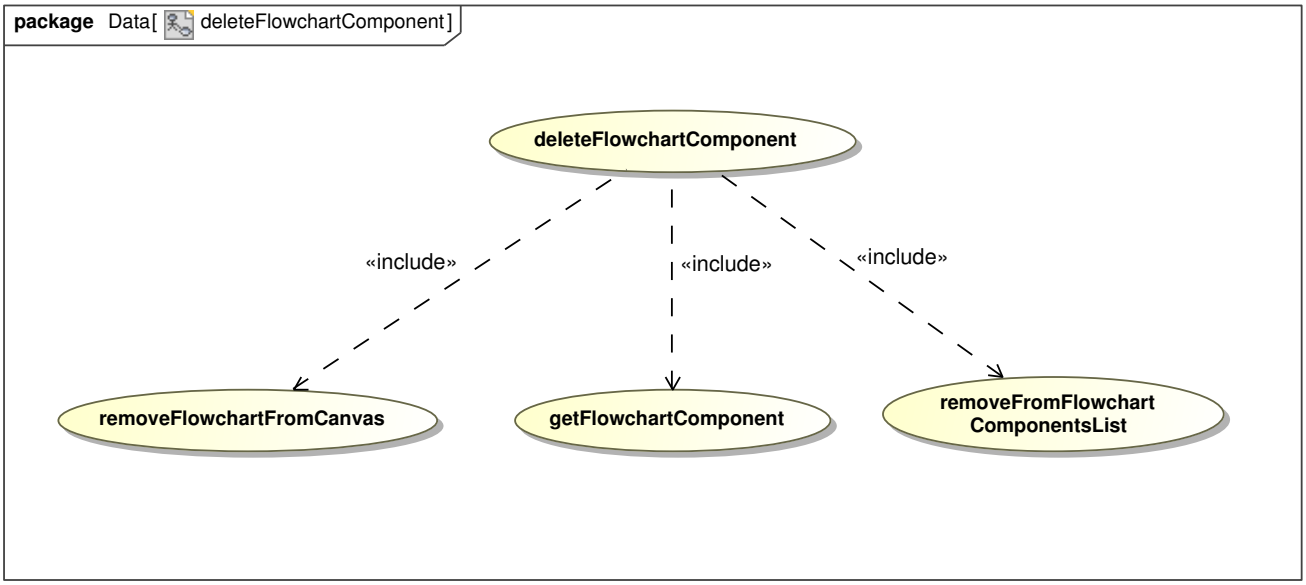
3.7 deleteFlowchartComponent

The deleteFlowchartComponent use case enables the functionality to delete individual components from the canvas.

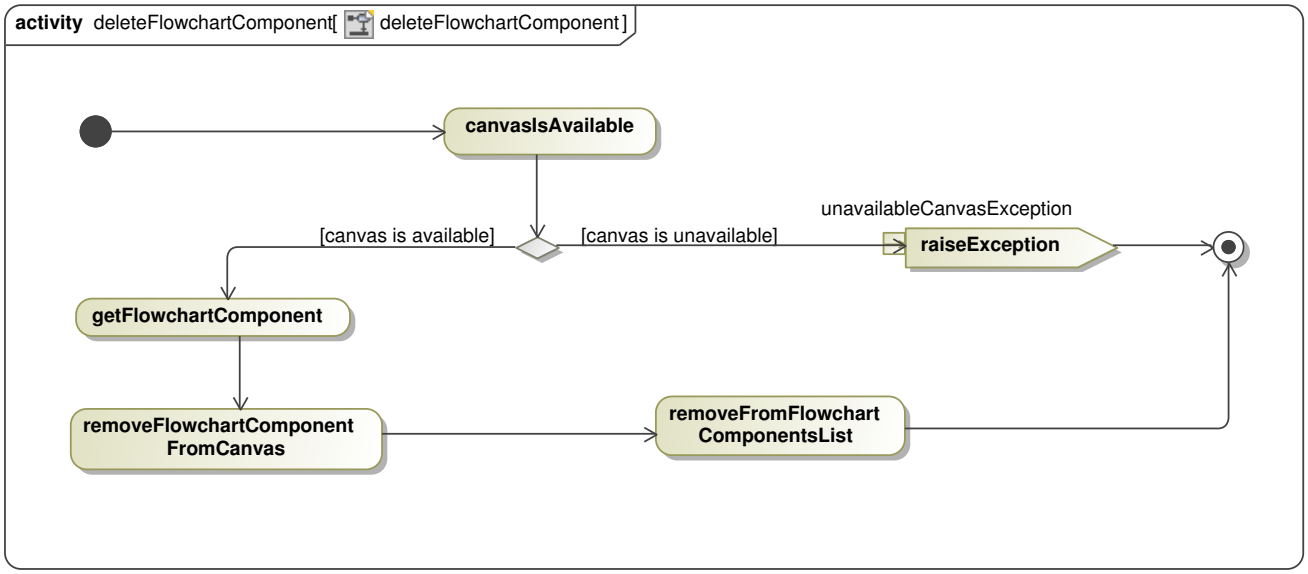
Pre Condition: The canvas has to be available. Component exists in the canvas space and is in the components list.

Post Condition: The canvas is clear of any components that were selected for removal.

3.7.1 Use case diagram



3.7.2 Activity diagram



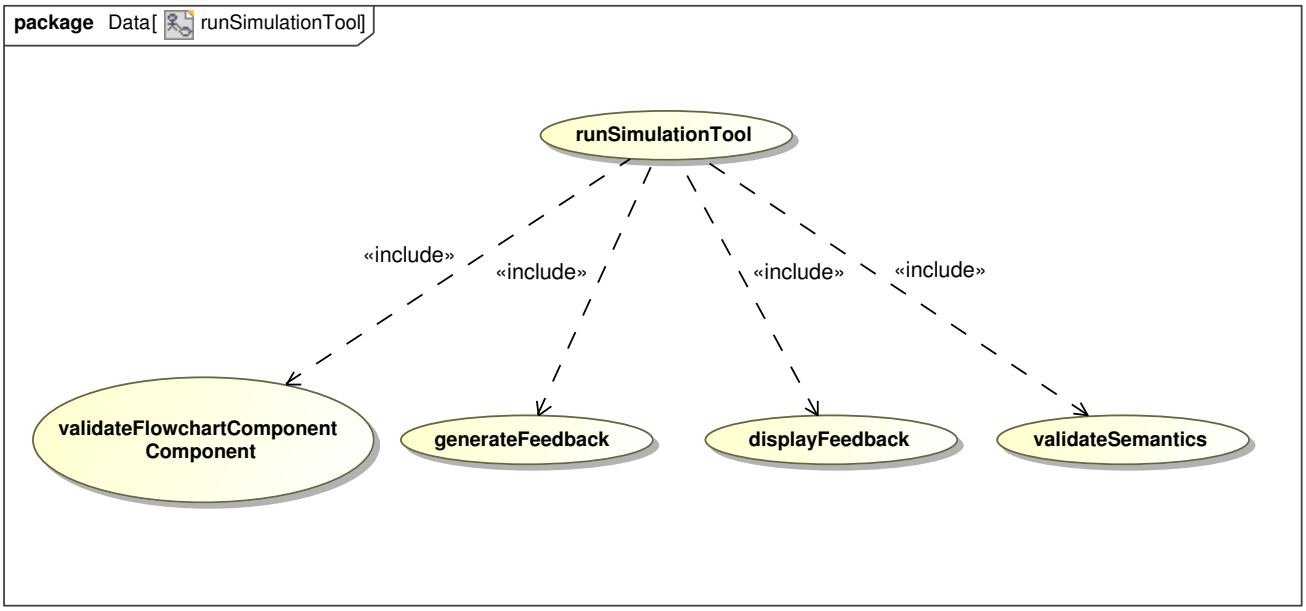
3.8 runSimulationTool

The runSimulation use case enables the functionality to execute the flowchart step-by-step or from start-to-end.

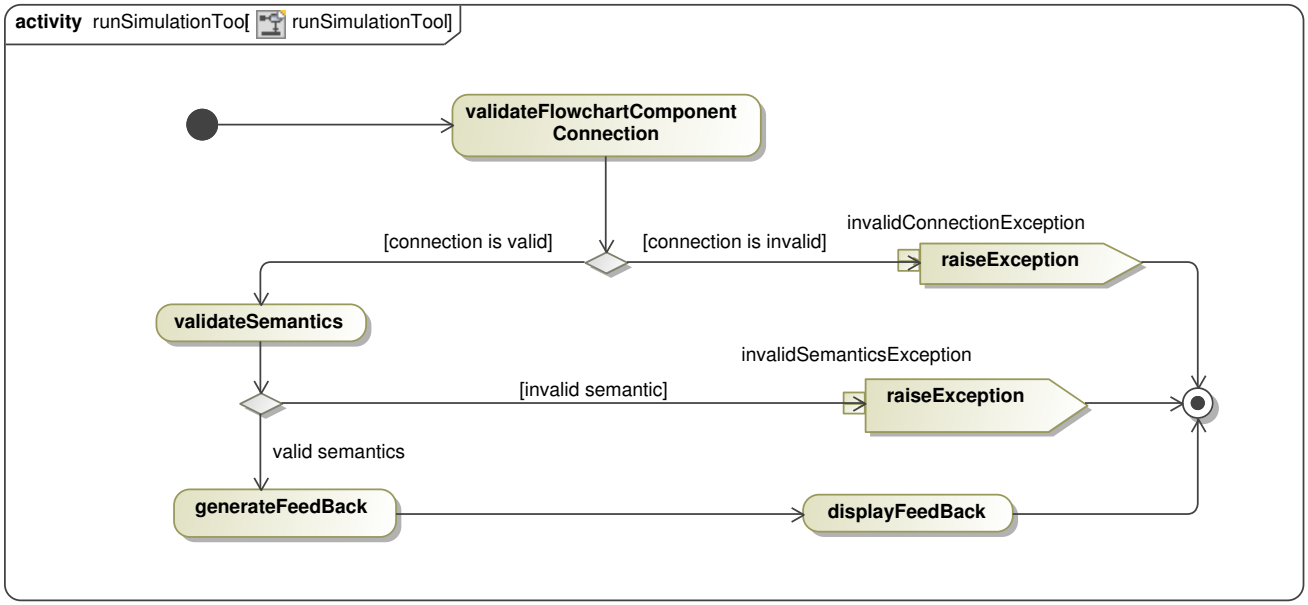
Pre Condition: Canvas has to be available.

Post Condition: Flowchart will return feedback of any errors, warnings or successful execution along with the results of any calculations.

3.8.1 Use case diagram



3.8.2 Activity diagram



4 The Domain Model - High-level