Homework Manual

Ryan Davis

September 2024

Contents

Chapter 1

Team

1.1 About Me

My name is Ryan Davis, and I am a Software Engineering major. I am in my 3rd year of study pursuing my bachelor's degree. I hope to continue my Software Engineering career one day as the founder of a startup company. I am excited to learn more about the modeling and representation of software in this course.

1.2 EPS Figures

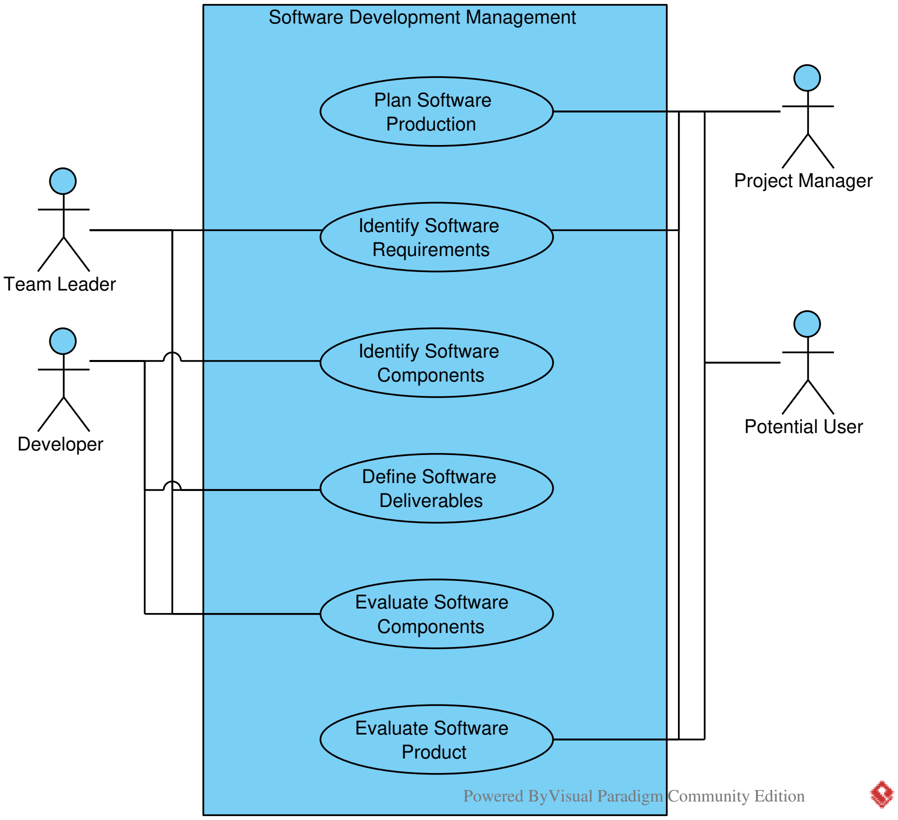

Software Development Management
Plan Software Production
Identify Software Requirements
Identify Software Components
Define Software Deliverables
Evaluate Software Components
Evaluate Software Product
Team Leader
Developer
Project Manager
Potential User
Powered By Visual Paradigm Community Edition

Figure 1.1: Completion Screenshot

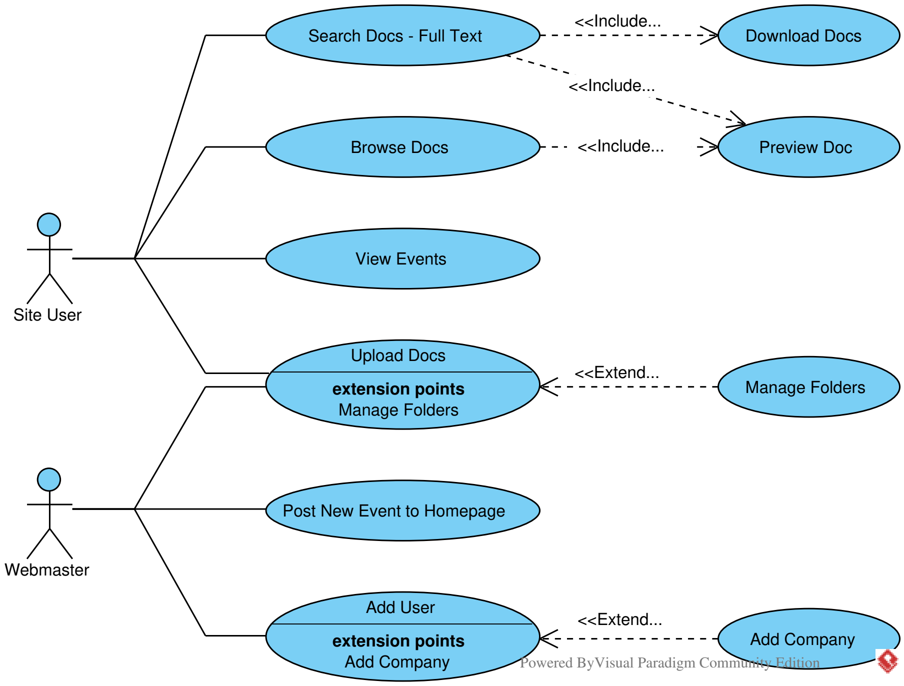
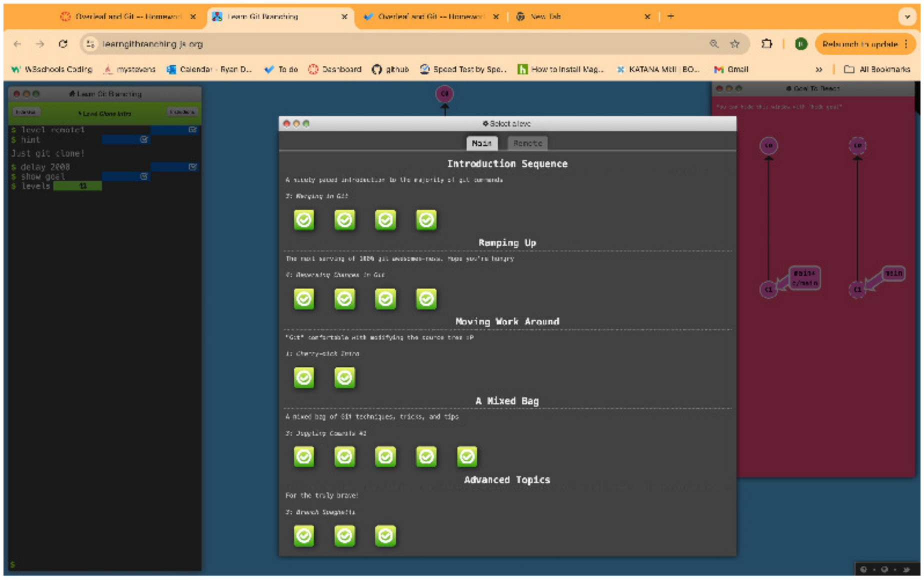


Figure 1.2: Completion Screenshot

Chapter 2

Git Homework

2.1 Learning Git



```
$ level remote1
$ hint
Just git clone!
$ delay 2000
$ show goal
$ levels
```

Introduction Sequence

A nicely paced introduction to the majority of git commands

3: Merging in Git

✓ ✓ ✓ ✓

Ramping Up

The next serving of 100% git awesomes-ness. Hope you're hungry

4: Reversing Changes in Git

✓ ✓ ✓ ✓

Moving Work Around

"Git" comfortable with modifying the source tree :P

1: Cherry-pick Intro

✓ ✓

A Mixed Bag

A mixed bag of Git techniques, tricks, and tips

3: Juggling Commits #2

✓ ✓ ✓ ✓ ✓

Advanced Topics

For the truly brave!

3: Branch Spaghetti

✓ ✓ ✓

Figure 2.1: Learn Git Branching Completion

2.2 Create a GitHub Issue

https://github.com/ryry91021/SSW-345-HW/issues/1

Chapter 3

UML Class Modeling

GitHub Link: https://GitHub.com/ryry91021/SSW-345-HW

3.1 Class Modeling Exercises

3.1.1 Exercise 2.1

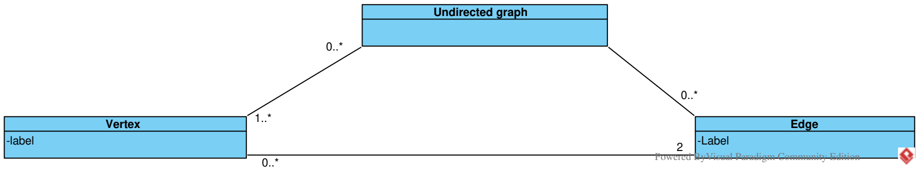


Figure 3.1: Exercise 2.1 UML Diagram

3.1.2 Exercise 2.2

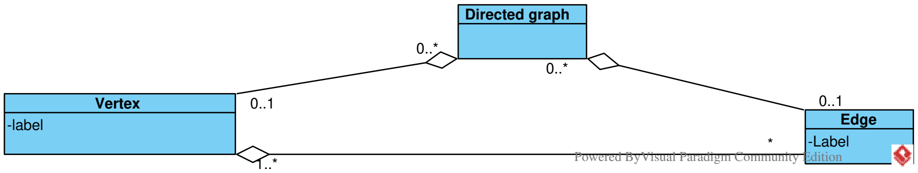


Figure 3.2: Exercise 2.2 UML Diagram

3.1.3 Exercise 2.3

In this windowing system, a window is a generalization used to display content, and has coordinate properties. The window is able to be moved and resized given its methods. A scrolling window is a type of window that is able to be offset by a scroll function. The canvas and the scrolling window utilize the text window and scrolling canvas classes. These two classes are responsible for the purpose of the window and how it is handled, whether it be for text or for reading.

A canvas is another extension of a window that can add and delete elements. A shape is found in a canvas, with attributes color and linewidth. A canvas can have many shape ”elements” and a shape can only have 1 window. There can be two types of shapes, one of which is the line, with attributes coordinate points and a draw method. Another type of shape is a closed shape, which has two attributes fillColor and fillPattern. From closed shapes, a polygon can be created, with a draw method. A polygon can create multiple points, called verticies, with attributes of coordinates (ordered). A point can only be created upon the instance of a polygon. Another closed shape object that can be created is the ellipse, with attributes of size and a method to draw it. The last type of window class is a panel, which allows for a window to accept an item name. A panel can have 0 or 1 panel item, and a panel item must have a panel to be displayed. A panel item is displayed on a panel, and in order for a panel to have a panel item, there must be 1 panel, thus the multiplicity of 1. Furthermore, a panel could have 0 or 1 panel item. A panel item can have 1 action known as a ”notifyEvent.” An event can affect many panel items. It can also have many text items through a keyboard event. A panel item can consist of 3 different types of objects. One of which is the button, with attributes for a label, and if it is pushed or not. Another type of panel item is a choice item. A choice item can either have 1 choice called the currentChoice, or multiple choices. currentChoice is a subset of choices, holding values string and values. A choice entry can have 0 or 1 choice items. The last kind of panel item is a textItem, with attributes of information about the text (or String). A textItem can have 1 keyboard event and an event can have many textItems.

3.1.4 Exercise 2.4

A customer has an attribute, name. This class has a many to many relationship with the class mailing address. A mailing address can have many customers, with the title accountHolder. A mailing address can also have many credit card accounts. Mailing address has the attributes address and phone number. A CreditCardAccount has atrtibutes maximumCredit and currentBalance and can have 1 mailing address. A creditCardAccount via ”statementDate” has 0 or 1 statements. Also a CreditCardAccount can have 1 institution. Via an ”accountNumber” an institution can have 0 or 1 credit card accounts, with attributes name, address and phoneNumber. A statement can only be assigned to one CreditCardAccount, with attributes paymentDueDate, financeCharge, and minimumPayment. 1 Statement via transactionNumber can have 0 or 1 transactions. A Transaction is a generalization with attributes transactionDate, explanation and amount. The types of transactions allowed are: CashAdvance, Interest, Purchase, Fee (with an attribute feeType), and adjustment. Many purchases can have 1 merchant, with the attribute name.

3.2 Class to code modeling

GitHub link to ProductSale.py: `https://github.com/ryry91021/SSW-345-HW/blob/main/class_modeling/PythonClasses/ProductSale.py`

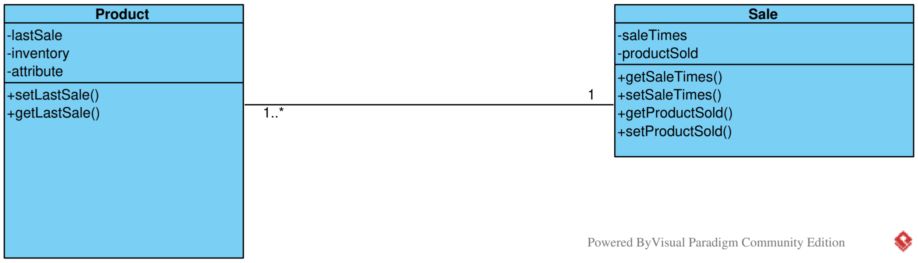


Figure 3.3: Updated UML diagram for ProductSale.py

Chapter 4

Software Execution Models

Github Link: `https://github.com/ryry91021/SSW-345-HW/tree/main/softwareExecutionModeling`

4.1 Exercise 5.1

- Best Demand Time: 21.934
- Worst Demand Time: 101.524
- Average Case: 86.95
- Link to Excel Spreadsheet and Work: `https://stevens0-my.sharepoint.com/:x:/g/personal/rdavis2_stevens_edu/ETH600gdY_9DrfYD2roeC-sB9Zc5XHYFpWN9g3Dq8y_4Tg?e=ZTUlf2`

4.2 Exercise 5.2

- Best Demand Time: 0.825
- Worst Demand Time: 2.425
- Average Case: 1.94225
- Link to Excel Spreadsheet and Work: `https://stevens0-my.sharepoint.com/:x:/g/personal/rdavis2_stevens_edu/ETH600gdY_9DrfYD2roeC-sB9Zc5XHYFpWN9g3Dq8y_4Tg?e=06s47p`

4.3 exercise 5.3

4.3.1 Question 1

Sequence Diagram:

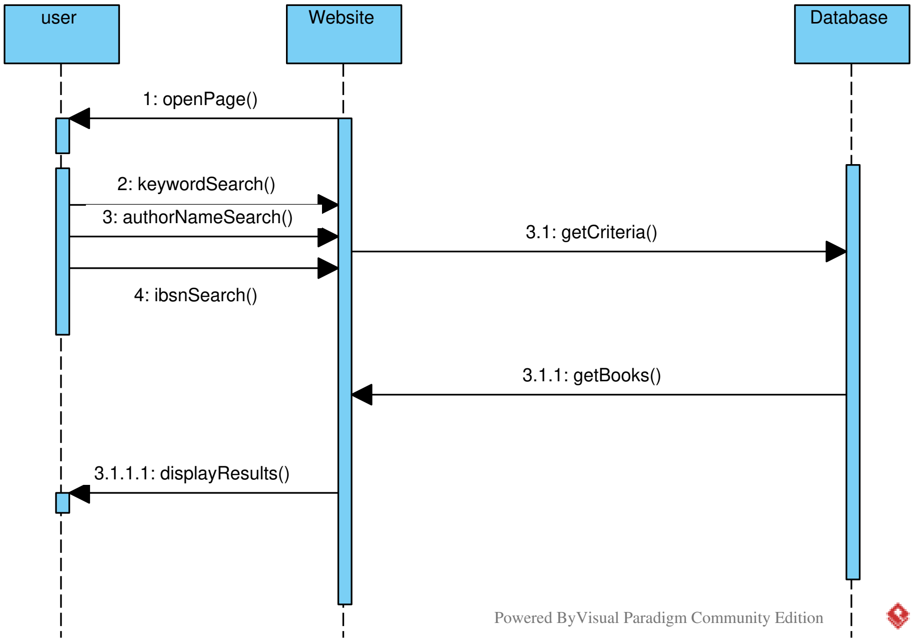


Figure 4.1: eStore Sequence Diagram

4.3.2 Question 2

Execution Model Diagram

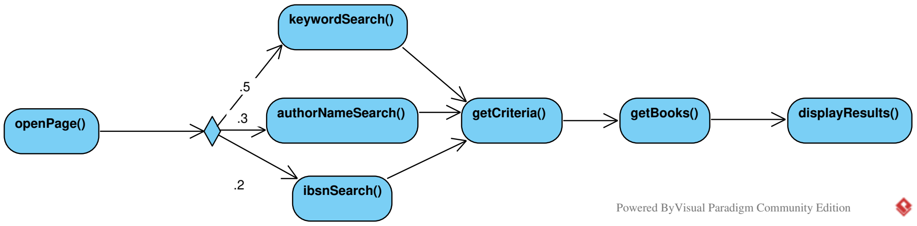


Figure 4.2: eStore search execution model diagram

4.3.3 Question 3

The constraint for the probabilities of the search paths is that the total probability of the searches must be equal to 1. The probabilities were also decided based on the commonality of the type of search.

4.3.4 Question 4

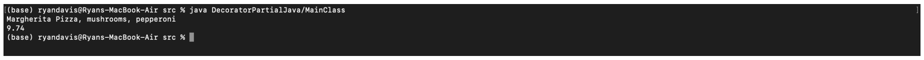
- Best Demand Time: 2.62
- Worst Demand Time: 42.62
- Average Case: 22.62
- Link to Excel Spreadsheet and Work: `https://stevens0-my.sharepoint.com/:x:/g/personal/rdavis2_stevens_edu/ETH600gdY_9DrfYD2roeC-sB9Zc5XHYFpWN9g3Dq8y_4Tg?e=HL94f4`

Chapter 5

Decorator Pattern

GitHub Link: `https://github.com/ryry91021/SSW-345-HW/tree/main/decoratorPattern/src/DecoratorPartialJava`

Output Results:



```
(base) ryandavis@Ryans-MacBook-Air src % java DecoratorPartialJava/MainClass
Margherita Pizza, mushrooms, pepperoni
9.74
(base) ryandavis@Ryans-MacBook-Air src %
```

Figure 5.1: Main Driver Output Results

UML Diagram:

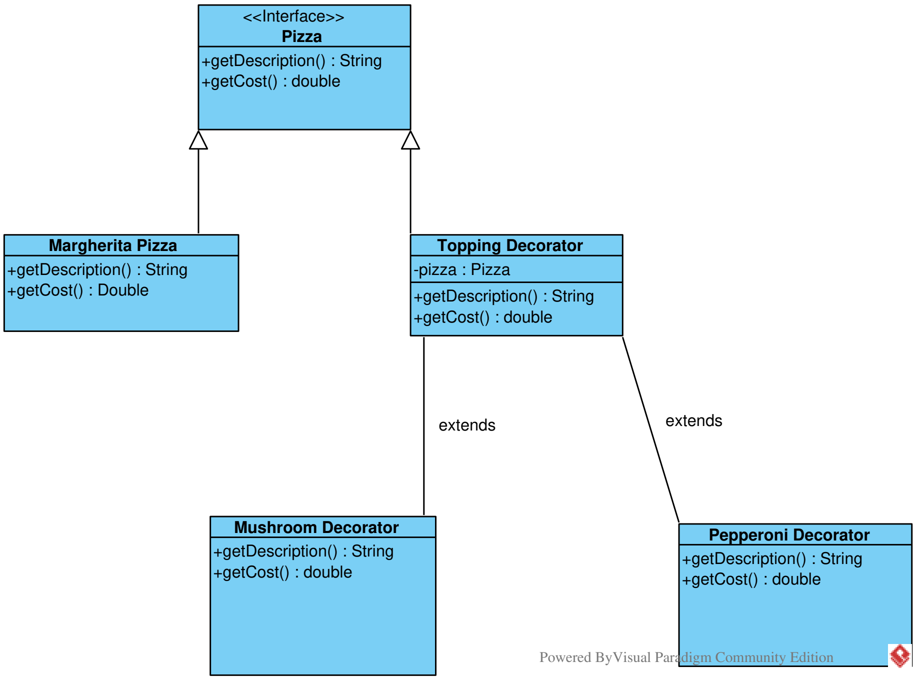


Figure 5.2: UML Diagram