Applied software engineering – Object oriented modelling

Hamza Bhatti (21223241)

Contents

[Introduction 2](#_Toc467578499)

[UML Modelling 2](#_Toc467578500)

[Use Case Requirements 2](#_Toc467578501)

[Use Case Diagram 2](#_Toc467578502)

[Class Diagram 2](#_Toc467578503)

[Sequence Diagram 2](#_Toc467578504)

[Architecture Diagram 2](#_Toc467578505)

[Implementation 2](#_Toc467578506)

[Software Testing 2](#_Toc467578507)

[Conclusion 3](#_Toc467578508)

# Introduction

For this assignment, a case study was provided. A use case was chosen from which a use case diagram, class diagram and sequence diagrams were produced. Along with UML diagrams, an architecture diagram was also produced. Finally, implementation of all these elements were produced and tested using Junit testing.

Could make a better intro

# UML Modelling

## Use Case Requirements

What’s the use case?

The use case that was chosen was “Update site popularity” which also included “Prioritise site for marketing”.

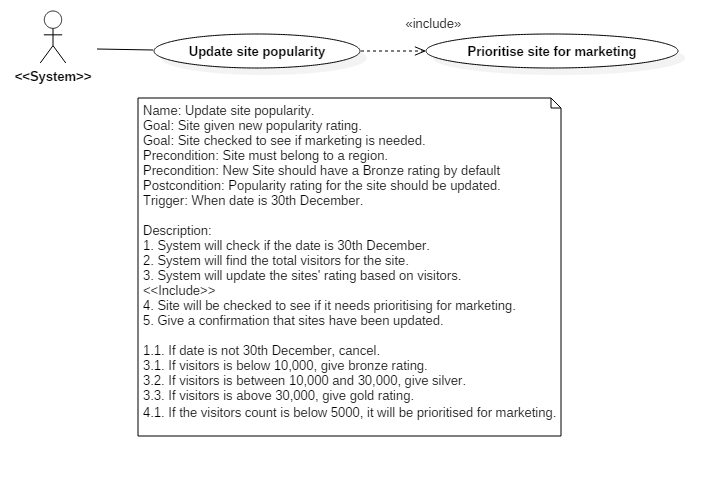
What needs to happen?

For the update “Update Site popularity” use case, certain aspects had been identified in the case study. The case study mentions that there are 6 regions in total, and sites belong to these regions. For the popularity of each site, a rating is given. These ratings are either Gold, Silver or Bronze. These ratings are based on the amount of visitors that the site has annually. If the visitors are less than 10000, the rating will be Bronze, if visitors are between 10000 and 30000 it will be given Silver, and finally if the visitor numbers are above 30000, Gold is given. These ratings change on the 30th of December.

When it come to the “Priorities site for marketing” use case, it is identified that this occurs during the rating process. For this use case, the site will be prioritised based on the amount of visitors that the site receives. If the site gets less than 5000 visitors, it will be marked for marketing.

## Use Case Diagram

A use case diagram was then created to describe the software process. The diagram is provided below.



Use case features, triggers etc. and link to requirements

In the diagram, the System is the actor. This is because the case study does not identify that there is a human or user that triggers this use case. Along with this, the trigger does not identify a user, it only stated that the date must be 30th of December with no user interaction.

The diagram also shows that the “Prioritise site for marketing” is included within the “Update site popularity” use case. This is because it was identified that the process of marketing would occur while the popularity was being updated.

There are two goals identified in the use case diagram. These two goals are to ensure that the site’s rating has been updated and that the site has been checked if marketing is needed. These relate to the identified use cases that has been identified previously.

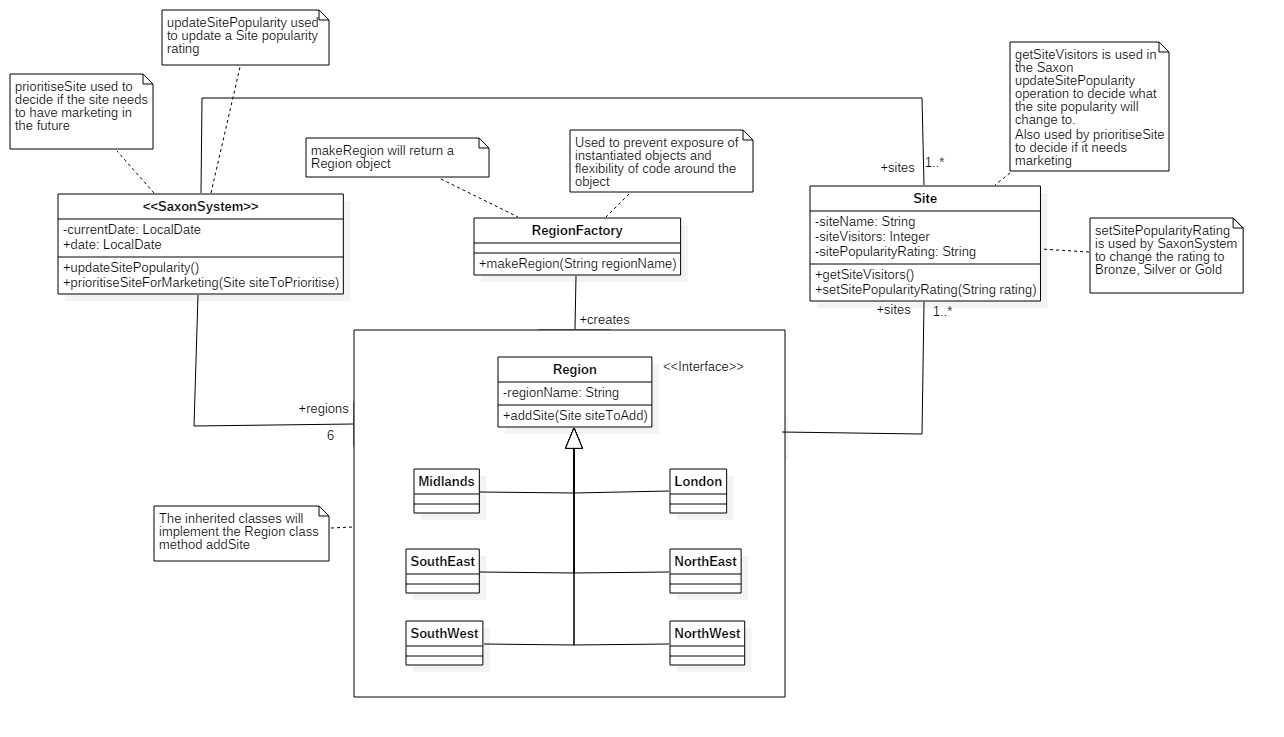
There are also two preconditions provided. One is that the site must belong to a region. This goes back to the case study where it states that each site belongs to one of six regions. Along with this, for implementation purposes, a new site will have a default rating of bronze. This is because Bronze is given to sites that are below 10000 visitors. Having zero visitors would also mean that the rating would be bronze.

The description and related to the requirements

First the system will check to see if the date is 30th December. There is an alternative here, when the date is not 30th December, nothing will happen and no changes to sites will be made. When the date is checked, the Site’s visitor count will be found and the rating for the site will be changed accordingly. Following the requirements, if the site has below 10000 it will be given Bronze, between 10000 and 30000 will be Silver and when above 30000 Gold will be given. Following the update, the site will be checked to see if it needs marketing, this will be decided if the visitors for that site are below 5000. Finally a confirmation is given.

## Class Diagram

A class diagram was created, showing what the different elements where needed so that the use case could be implemented. The diagram is shown below.



Talk about all the elements of the system, how they relate to the description

There is a SaxonSystem class. This deals with updating site popularity ratings and also checking if they need prioritising for future marketing. There is also a Site class which is what the system will be modifying in terms of its popularity. Many Regions are present, all of which will contain many Sites.

How do the classes work together? Their associations and methods

The class diagram shows associations between the classes present. The SaxonSystem has many one to many Sites. This is because it is logical to assume that there are more than one Site handled by Saxon. This also makes it easier to change their popularity ratings when there is a collection of them. SaxonSystem will also have a collection of six regions.

The Regions that are created with the factory will also have one to many sites. This is so because Sites belong to Regions and there must be at least one Site in a Region.

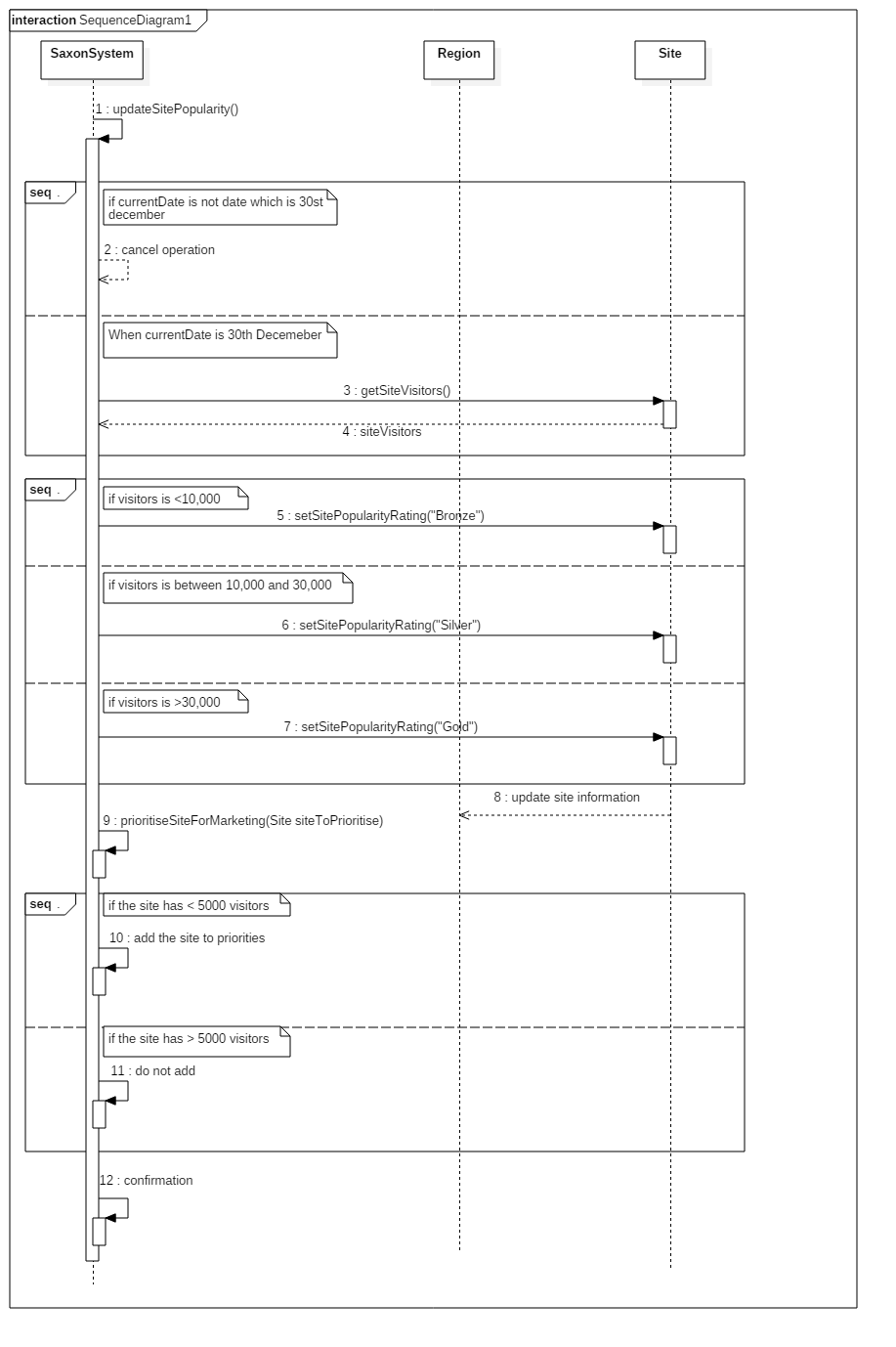
How is the class diagram abstract?

The class diagram itself is abstract. This is so because everything within the diagram is relevant. There are no extra classes present that do not have a purpose or do not follow the use case scenario. Also, there are not extra methods or attributes present that do not need to be there. The methods an attributes present are ones that are useful to reach the goal of the use case.

Refactoring with the design pattern, why this one?

## Sequence Diagram

A sequence diagram was created to show the interaction between the classes that were identified in the class diagram. The diagram is shown below.



What classes are there?

The classes that will interact with each other are the SaxonSystem, Region (which can be either of the six variations of the Region) and the Site class.

Talk about the order of things going on, how they relate to the requirements

The sequence starts off with the SaxonSystem class. As mentioned before, there are no users that are considered actors here, it is strictly the system doing the work on its own. The first method that will be called is the updateSitePopularity(). This clearly falls in line with the use case name. A check will occur to see if the currentDate is the 30th of December. An alternate flow occurs here, where if they are not the same the process will not continue, but will continue if they are the same.

The Site method getSiteVisitors() is used to retrieve siteVisitors value for an instance of a Site. This information is then used to help assign a new popularity rating for the Site.

Another alternate flow is used when the setSitePopularity() method is used. When the siteVistors value is below 10,000, the rating will be set to Bronze, when between 10,000 and 30,000 a Silver and above 30,000 a Gold rating. When the new rating is set, this information will reflect in the Region classes.

Once the new popularity rating has been given to an instance of a Site, the SaxonSystem will call the prioritiseSiteForMarketing() method that will take the instance of the Site. Another alternate flow will occur here. If the sitePopularity is below 5000, it will be added to a list of priorities, otherwise it will not be added.

Finally a confirmation message will be given that the Site has been updated.

## Architecture Diagram

What architecture is used? Give a reason, why is it appropriate for this use case

# Implementation

Describe the software produced and how it meets requirements

Give running instructions for windows terminal and linux shell for the .jar file

Proof of execution

# Software Testing

What tests do you have? How do they relate to the requirements? Based on the sequence diagram

Give instructions on how to run the tests

Proof of execution, which tests failed and passed and why?

# Conclusion

Talk about what your code does,

How it is maintainable etc

Appendix

Code listing