**CS673S16 Software Engineering** 

**Team 4 - Bird’s Eye Statistics**

**Software Design Document**

|  |  |  |  |
| --- | --- | --- | --- |
| Team Member | Role(s) | Signature | Date |
| Matthew Scott | Team Leader | *Matthew Scott* | 04/20/2018 |
| Siwei He | Configuration Lead | *Siwei He* | 04/21/2018 |
| James Christensen | Security Lead | *James Christensen* | 04/23/2018 |
| John Staley | Requirement Lead | *John Staley* | 4/26/2018 |
| Patrick Ryan | Assistant Leader | *Patrick Ryan* | 04/26/2018 |
| Amal Krishna | Quality Assurance | *Amal Krishna* | 04/26/2018 |
| Peter DiMaria | Environment | *Peter DiMaria* | 04/26/2018 |
|  |  |  |  |

**Revision history**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Author** | **Date** | **Change** |
| **1** | **Matthew Scott** | **2018/04/20** | **Initial** |
|  |  |  |  |

[Introduction](#_87t9hln2vjz0)

[Software Architecture](#_buttcq9i221r)

[Design Patterns](#_x18fj36s1121)

[Key Algorithms](#_mtfbusfb0eq3)

[Classes and Methods](#_7ucksmkf6rzx)

[References](#_15tmymhipvdv)

[Glossary](#_8n34lvocupub)

# Introduction

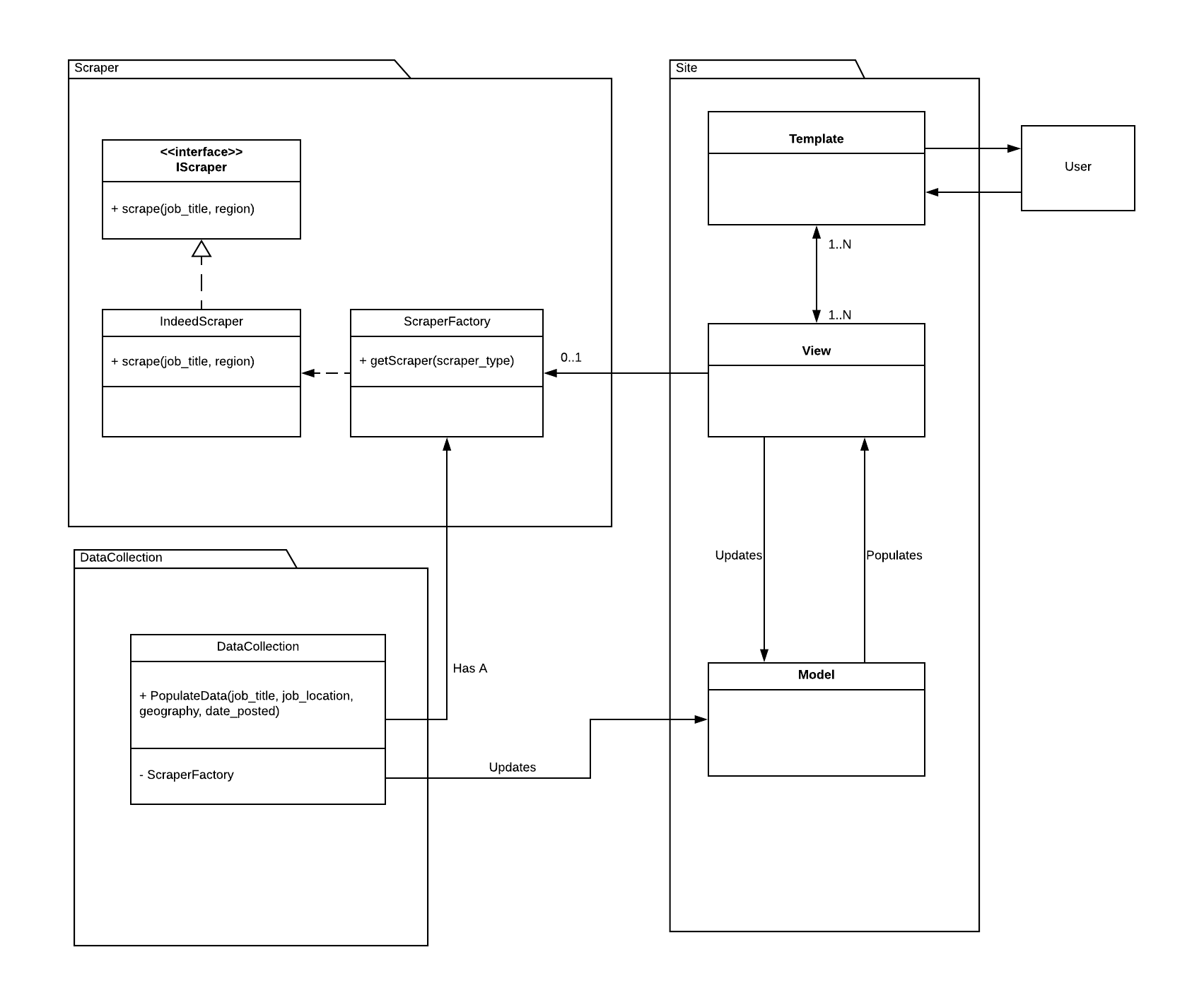
The Job Skills Statistics Website, as its name implies, is meant to present data on skills desired for jobs. The presentation of this data can take a number of forms and a major portion of this project is presenting a variety of graphs that give our users the insight that they desire into the job market. For example, a basic graph that we display is a bar graph showing the relative number of times that skills have been found to be associated to a type of job.

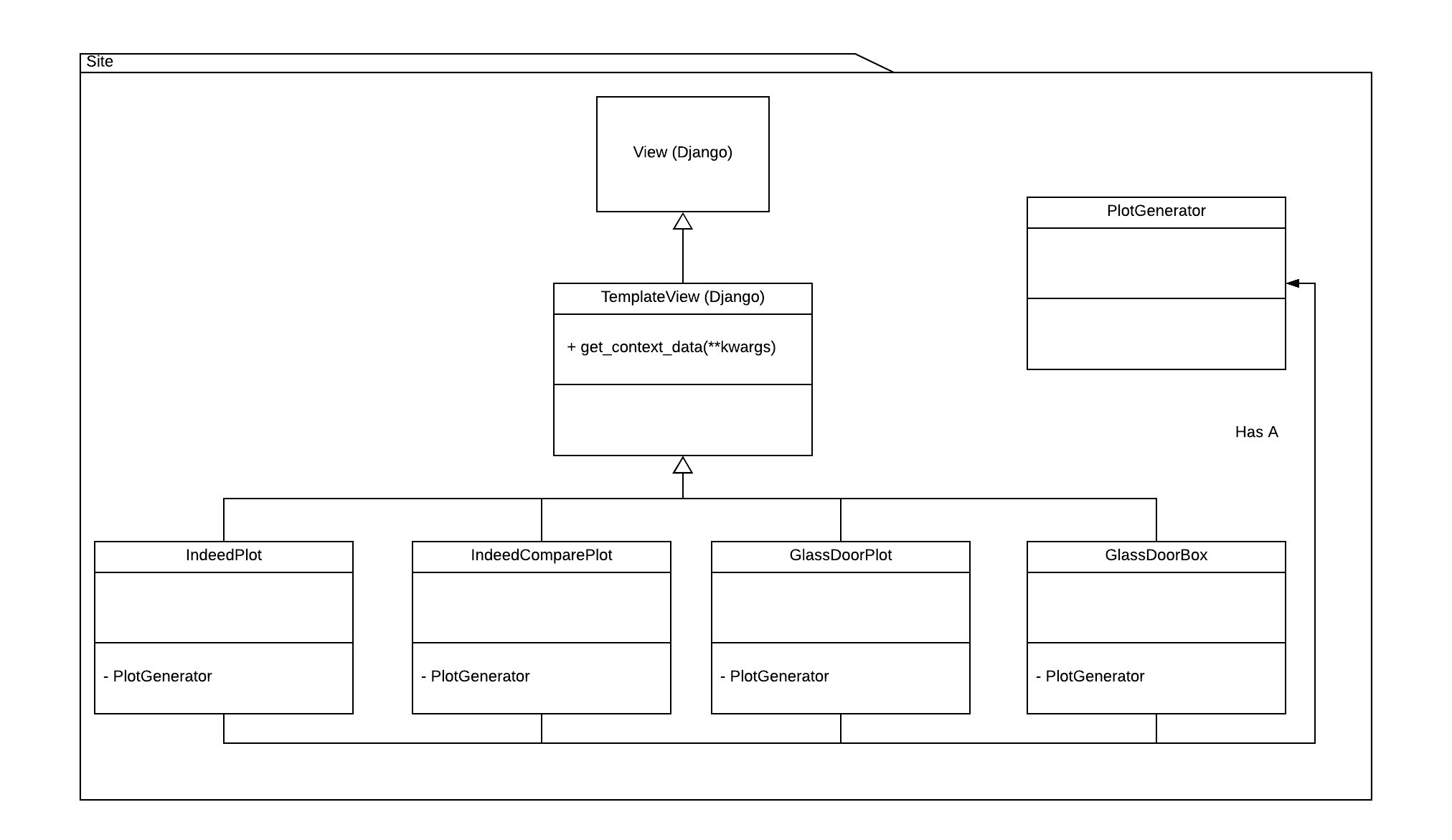
Another major portion of this project is collection of the data used for graphs. Some of the data that we have collected comes direct from providers in a format already consumable by our graphing module. Our major source of data is job postings on the website Indeed.com, which we’ve gathered by crawling to and then scraping and parsing the raw html of individual postings.

Because of our feature goals, our major modules are for collecting and storing data, CollectData, crawling and scraping websites like Indeed.com, Scraper, and presentation of that data which is incorporated into the general presentation of our website. That presentation is done utilizing the Django architecture, in which modules are separated into “apps”. Our only app is the Site app and all of our presentation logic can be found there. Additionally, some of the Django logic for presenting websites in general can be found in the EmploymentSkillsStatisticProject package, which has the same name as our project.

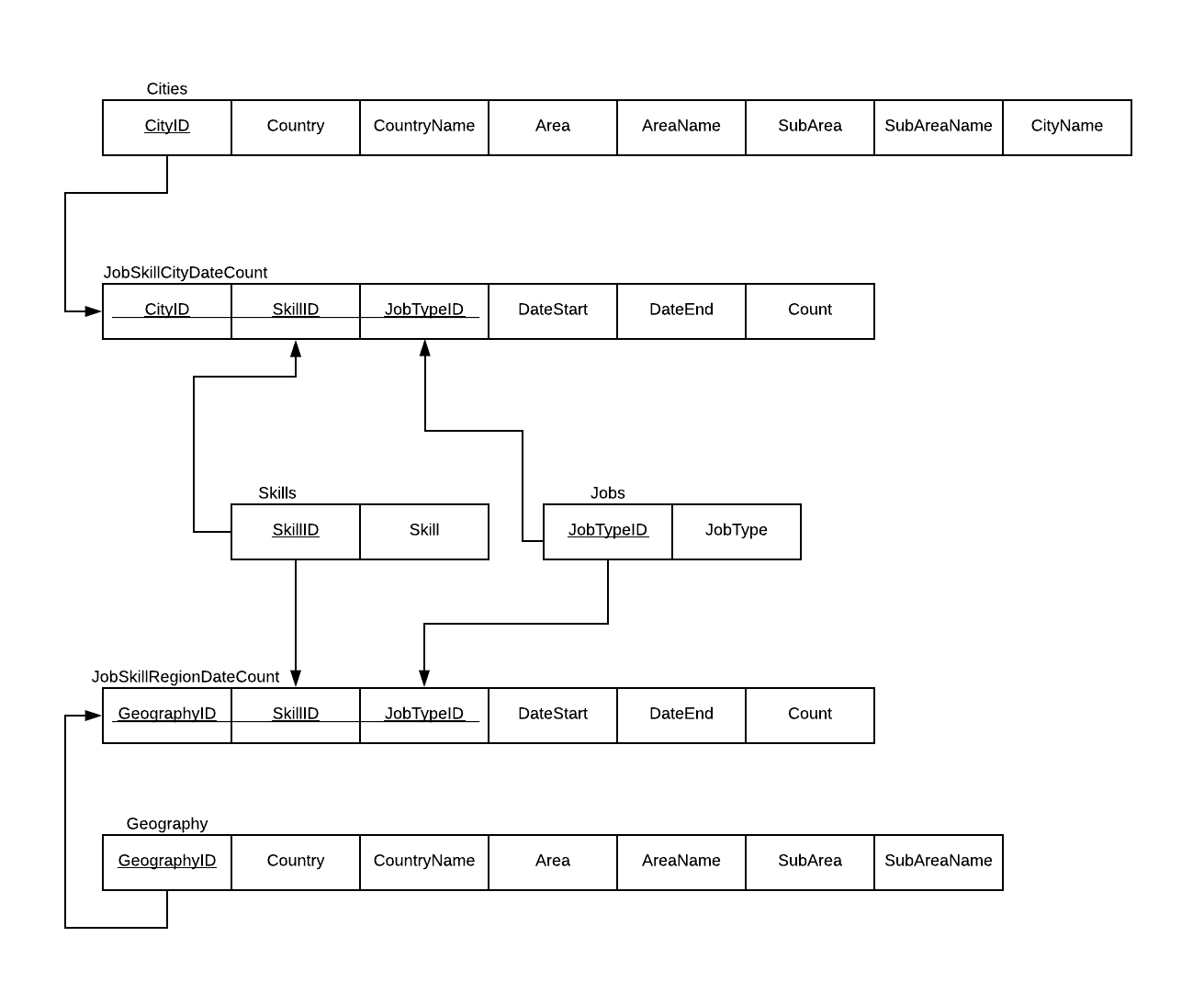
# Software Architecture

* + EmploymentSkillsStatisticsProject
    - This package is necessary to generate a website using the Django architecture. Some information used for presenting our website is contained in this package. The paths for various files files, including our sqlite database is in the settings file. The urls file links website URLs to views in the Site package.
  + Site
    - This package primarily contains our views and our models. The models file describes our code first database. This includes those entity objects needed for storing and displaying data. The views describe both how to render the html files in the templates folder and how and what data to render, while the aforementioned urls file describes on which page to render. The html files themselves along with the Django framework also describe how they should be rendered through keywords referenced in the views and recognized by the Django framework. Some of the views utilize the scraper classes directly to pull and then visualize data, while others display data from persistent storage.
  + Scraper
    - This package is responsible for crawling and scraping web sites for data. IScraper is the interface describing the behavior for scrapers of different websites. The scraper file is currently the implementation of IScraper for the indeed web page.
  + CollectData
    - The CollectData package accepts data directly from a scraper and stores it persistently. This package may also modify the data for better storage. For example, one of its uses is to aggregate the data from a scraper return for quick display of that aggregated data in some of our graphs.
  + Architecture Diagram:





* + Database Design:



# Design Patterns

We are using the standard factory pattern in the scraper module to allow for flexibility in adding new scrapers when and if the need arises. See above whole package diagram for factory pattern class diagram.

# Key Algorithms

One of our key algorithms is in scraping the indeed website and individual job postings. This is primarily accomplished through the BeautifulSoup package, but the logic specific to our project is described below.

Indeed Scraper:

Job listings page:

For all divs (“div” elements where class = “row”) in the html (each job)

jobTitleElement = the “a” element where data-tn-element = “jobTitle”

jobTitle = jobTitleElement.text

postUrl = the “href” element in jobTitleElement

companyElement = the “span” element where class = “company”

If (companyElement == null)

companyElement = the “span” element where class = “result - link - source”

companyName = companyElement.text

Location = the “span” element where class = “location”

If (there is a “nobr” element)

Salary = text in this element

Else if (there is a “div” element where class = “sjcl”

Salary = text in the div in this element

Else

Salary = “No Salary Provided”

# Classes and Methods

# References

# Glossary