**[Project Title]**

**Project Deliverable 1: Requirements Analysis**

**9/22/2022**

Group 03

*Contributors:* Arunabh Bhattacharya,

Kathleen Tiley,

Benjamin Weiss, and

Jonathan Williams

Table of Contents

[Project Overview (1,3) 2](#_Toc114738711)

[Data (2, 5) 2](#_Toc114738712)

[Trend Analysis (4, 6) 2](#_Toc114738713)

[Implementation Method (7) 2](#_Toc114738714)

[Citations 2](#_Toc114738715)

Action Items

* Jon will merge two existing documents, Kathleen will review
* Ben will describe web application, maybe provide a visual
* Ben will flush out tech justification of database
* Ben will focus on implementation plan
* Kathleen will talk about users in project overview
* Kathleen, Neal will consolidate and flush out trend queries/goals, as well as anyone else who wants to jump in

# Project Overview (1,3)

This project will implement a web-based user interface allowing users to analyze changes in book publication rates and popularity by genre, publication language, and timeframe. The web application will have a main page with prepared interactive visualization the represent the key trend queries. Users will be able to create their own visualizations with a filter tool on another page.

* General description of what website will do and how it will look
* Who are users and why are they interested (sort of (2) as well) – publishers considering what kinds of books to publish, authors considering what genres to choose or what languages/countries to write in, book advertising firms looking for information about book buyers, academics interested in when readers begin to list a book as a classic, or how readers think about genres, how genres have changed over the years

**Potential vision for this:** a splash page with some demo interactive charts, a page with instructions on how to use the interactive charts and filter system, and user sign in feature so users can save certain filters like language or region of interest and save charts they’ve generated.

# Data (2, 5)

To implement the web-based user interface, we will use public datasets scraped from goodreads.com for academic purposes in late 2017. The data consists of three groups of datasets which can be merged using the unique identification numbers for book, user, and review. The data can be downloaded as json files from a website established by the data collectors [1]. The data collectors also create a GitHub repository with jupyter notebooks to help users download and read the datasets [2].

The Book Metadata dataset group includes five datasets providing information about around 2.3 million books. The first three datasets provide information about the book itself, it’s author, and it’s genre. An additional dataset called works provides an abstract view of the book that is not dependent on a specific edition. The last dataset in the group associates the book with a series, if it is part of one. Within Goodreads, users create named public or private shelves and add books to those shelves. Our data determines a book’s genre by determining the most popular shelf name on which users choose to place the book. The core of the Shelves dataset is the User-Book Interactions Dataset, which provides over 229 million records of different user interactions with books. These interactions include ratings, leaving reviews, or placing books on shelves. Since the dataset is so large, the data collectors provide csv files modeling the interactions as a series of integers. The Reviews dataset group provides information on more than 15 million user-generated reviews of books. Different datasets in this group provide subsets of the full Reviews dataset which exclude reviews or review text that include spoilers. Each review record includes information on the associated rating for the book and the date the review was posted and updated. The full review dataset also includes the complete multilingual text of the review.

For our project, we will use the Book, Author, Genre, and Review datasets to support the trend analysis queries our website will process. Together these datasets include around 20 million records. Our project requires the use of a database management system to deal efficiently the size of the data and the relationships between the datasets. In addition, because Goodreads no longer issues new API keys since December 2020, our web application cannot directly query the existing Goodreads database. Since the core functionality of our application consists of conducting complex trend queries on a large database with interrelated components, our project requires database support to implement the application.

# Trend Analysis (4, 6)

* More specific info on what kind of questions website will answer, discuss goals and enumerate queries

# Implementation Plan (7)

* JSON, python to get data into Oracle, HTML/CSS/JS website
* If data is too large, will filter things out – only do english language books or books with country code US, for example

# Citations

* We need to site those two sources mentioned on the website

[1] <https://sites.google.com/eng.ucsd.edu/ucsdbookgraph/home>

[2] <https://github.com/MengtingWan/goodreads>

Overview

Historical trend analysis of book reviews and ratings sourced from goodreads.com.

Project Description

Core Functions:

Trend analysis on which genres popular in which languages: query reviews

The project will implement a web-based user-facing interface which will allow user interaction and custom queries.

Requirement for a Database System

# of books and authors is stupid big

As of December 2020, goodreads does not issue new API keys. (data access is not trivial)

Only other method for accessing data is scraping which is expensive(compute + time) and most major websites impede scrapers.

Even if scraping were easy and cheap(time + compute), storing stupid amounts of data in a file and then running analysis on that data would be expensive(compute + time). IO is a huge bottleneck.

Another downside to storing in files is redundancy and fragmentation of data.

Database solves all these problems:

15M API calls is hella scrilla, storage is the answer.

RDB storage eliminates redundancy and fragmentation. TABLE Author, TABLE Book, etc.

DB IO is cheap compared to file read and write.

Update/Deletion is worry free compared to files.

There are more than 10,000,000 books listed on goodreads.com from more than 17,000 authors.

Web-Based User Interface Functionality

Data

Software

Trend Analysis and Queries

Trend Analysis Goals

Trend Queries

Which genre(s) had specific ratings over a particular time period?

E.g. “Which genres had an average rating lower than 3-stars, from January 2007 – January 2009?”

Which genre received the greatest/least number of reviews over a particular time period.

**Phase 1**

**The summarized requirements from Prof:**

propose and understand an appropriate project topic, identify its main data management needs, explore and motivate its potential for interesting queries, and analyze the needed user functionality.

1. What are the main functions that the web-based user interface should provide?
2. How do the different functions work together? Sometimes there are dependencies between different functions.
3. Which real-world data are needed to support the functions identified before?
4. Can such real-world data be found in the Internet?
5. What (colloquial) queries are important for the application?
6. Which public domain and/or proprietary software is needed to perform the task? (The database system used must be CISE Oracle.)

(PDF file) that presents a clear and structured description and motivation of the project topic and requirements that the software solution should later fulfil.

The focus of this project is supposed to be on the database part and not so much on the application part. focus on database queries that evaluate large volumes of stored data. Of course, the application part must be highly functional, and the different user functions must cooperate nicely together. However, a fancy layout design of the user interface is not required but appreciated.

It is important that each group demonstrates in their deliverable that their application would really benefit from database support and that new information (such as trends) can be derived from the stored data. A simple retrieval of data from the database (that is, search) or the pure connection of different tables (that is, joins) are not sufficient. As an example, let us assume that a group selects a sales application as their project topic and stores many daily sales numbers in their database. Of course, one can search for sales data of interest in the database and display them in the user interface. But searching only identifies an interesting subset of all data stored in the database. DBMS are specialized for search tasks, and the respective SQL queries are relatively simply structured. This project aims at more interesting queries that, first, derive new information which is not explicitly stored in the database but can be derived from the data in the database by computations and, second, represent trends (see Section 3).

Arunabh-

Since we are required to do trend analysis and compute the changes in data over a period, based on the datasets, I could think of the following:

For point 5, some of the colloquial queries could be:

* What was the average number of books read during a certain period, ordered by genre?
* Which genres had the highest ratings during a certain period?
* Which genres received more reviews during a certain period?
* What was the percent of shelved books by genre?
* What was the ratio of read and shelved books by genre?