МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»**

Кафедра Математического и компьютерного моделирования

Разработка информационной системы для хранения и обработки информации о научных публикациях

АВТОРЕФЕРАТ МАГИСТЕРСКОЙ РАБОТЫ

студента 2 курса 247 группы

направление 09.04.03 — Прикладная информатика

механико-математического факультета

Ратушного Александра Васильевича

Научный руководитель
доцент, к.т.н. И.А. Панкратов

Зав. кафедрой
зав. каф., д.ф.-м.н., доцент Ю.А. Блинков

Саратов 2020

Введение. В настоящее время важной частью деятельности каждого члена научного сообщества является написание научных работ, и соответственно подготовка списка использованной литературы, для чего используются системы управления библиографической информацией. У каждого автора растет количество опубликованных работ, которые необходимо систематизировать. При написании работы создание списка публикаций может потребовать большого количества времени. Подготовка достоверной и тщательно проверенной библиографии является важнейшим требованием для принятия работы к опубликованию, а также крайне важна для процесса компетентного рецензирования. Работникам научной сферы необходимо иметь сведения о различных наукометрических показателей себя и своих коллег, например количество цитирований. Актуальной проблемой является необходимость создания программного обеспечения, решающего эти задачи. В связи с большим прогрессом в области хранения данных, а именно развитием документоориентированных БД, таких как MongoDB, имеется возможность создать информационную систему для хранения информации о научных публикациях, реализующую больше необходимых функций. В качестве целей данной работы можно выделить:

- Исследование существующих систем управления библиографической информацией;
- Исследование источников библиографической информации;
- Обзор подходящих средств разработки;
- Проектирование информационной системы;
- Разработка информационной системы.

Основная часть. Основная часть работы состоит из трех разделов, а именно:

- Анализ предметной области;
- Обзор технологий разработки;
- Разработка программного обеспечения.

Первый раздел посвящен обзору существующих систем и источников библиографической информации, а также выявлению актуальности реализуемого программного обеспечения.

В настоящий момент существует множество систем управления библиографической информацией. Реализованные системы обладают разнообразным функционалом. Некоторые системы работают локально, в то время как другие расположены на веб-сервере. Одна часть систем направлена на составление библиографий, а другая на управление цитированием. Каждая система имеет свой список поддерживаемых форматов импорта и экспорта данных. Для того, чтобы реализовать актуальную информационную систему необходимо проанализировать существующие программы и определить их недостатки. Для обзора были выбраны системы управления библиографической информацией, имеющие стабильную версию не ранее 2018 года и с действующей поддержкой. В процессе работы было исследовано десять систем, среди которых Mendeley, KBibTex, JabRef, после чего были выявлены их недостатки.

Для наполнения базы данных следует изучить источники библиографической информации. Основными ресурсами, содержащими научные публикации и сопутствующую информацию являются Scopus, Web of Science и Google Scholar. Рассмотрим источники подробнее.

Scopus – библиографическая и реферативная база данных и инструмент для отслеживания цитируемости статей, опубликованных в научных изданиях. Разработчиком и владельцем Scopus является издательская корпорация Elsevier. База данных доступна на условиях подписки через веб-интерфейс. Сервис имеет собственный API, но интерфейс ограничен для пользователей без соответствующего доступа. С использованием бесплатного аккаунта разработчика имеется возможность реализовывать только функцию abstract-search. Сервис предоставляет удовлетворительное количество информации и его можно использовать для пополнения информации о научных публикациях. На странице автора в системе отображается наукометрическая информация, которую можно получить методом парсинга страницы.

Web of Science(WoS) – поисковая интернет-платформа, объединяющая реферативные базы данных публикаций в научных журналах и патентов, в том числе базы, учитывающие взаимное цитирование публикаций. WoS не предоставляет бесплатного доступа к своему программному интерфейсу, однако существует сервис Publons.

Publons – коммерческий веб-сайт, предоставляющий ученым бесплатный сервис для отслеживания, проверки и демонстрации их экспертных оценок и редакционных материалов для академических журналов. Он был запущен в 2012 году и куплен компанией Clarivate Analytics (которая также владеет Web of Science, EndNote и ScholarOne) в 2017 году. Publons предоставляет бесплатный доступ к своему API, через который можно получить информацию как о авторах, так и о публикациях.

Google Scholar – это свободно доступная поисковая система, которая индексирует полный текст или метаданные научной литературы по целому ряду издательских форматов и дисциплин. Google Scholar индексирует большинство рецензируемых онлайн академических журналов и книг, материалы конференций, диссертации, препринты, рефераты, технические отчеты и другую научную литературу, включая судебные заключения и патенты. Google не позволяет автоматизированно получать данные со своих серверов, за исключением платной подписки.

При сборе информации из нескольких источников необходимо обратить внимание на формат и возможное дублирование информации. Для устранения проблем с дублированием записей необходимо предусмотреть не только ручное удаление повторяющейся информации, но и поле, по которому можно однозначно идентифицировать ту или иную научную информацию. С этой целью полностью справляется DOI.

DOI – Digital Object Identifier или Цифровой идентификатор объекта – стандарт обозначения представленной в сети информации об объекте. Фактически, DOI является ссылкой (URL) на постоянное местонахождение объекта или информации о нём (метаданные) в Интернет. DOI может быть присвоен любому объекту. Объект может быть чем угодно: онлайн-публикация или её часть (книга, глава книги, статья) или элемент (рисунок, таблица, формула и т. п.), аудио-видео-контент, наборы данных и базы данных, материальные объекты (DVD, бумажная книга). DOI позволяют однозначно и точно идентифицировать объект и получать доступ к подробной информации о нем (метаданные) либо непосредственно к самому объекту, если доступ к объекту является открытым. С помощью API сайта можно по идентификатору doi получить всю необходимую информацию о научном труде, включая название,

время публикации и авторов, в том числе в формате BibTex. В результате изучения информации об источниках научных трудов для наполнения информационной системы были выбраны сервисы Publons и Scopus.

BibTeX – программное обеспечение для создания форматированных списков библиографии. BibTeX применяется совместно с LaTeX'ом и входит во все известные дистрибутивы TeX и LaTeX. Каждая запись содержит некоторый список стандартных полей, среди которых список авторов, названия статей и журналов.

В результате проведенного анализа предметной области можно выявить ряд функций, которых нет в существующих продуктах, а именно:

- Получение библиографической информации из крупных онлайн-источников;
- Интеграция с форматом BibTex;
- Дополнение неполной информации;
- Оффлайн доступ к данным.

Актуальность реализуемого программного обеспечения состоит в том, что информационная система позволяет реализовать востребованные вышеперечисленные функции. После рассмотрения существующих решений хранения и анализа научных публикаций можно прийти к выводу, что каждая из систем реализует лишь часть необходимого функционала, и необходимо создать информационную систему, которая могла бы хранить, анализировать и структурировать информацию о научных публикациях.

Во втором разделе приведено описание выбранных технологий разработки. В соответствии с необходимым функционалом было составлено общее представление о планируемой программе. Для реализации программы был составлен список этапов разработки информационной системы:

- Выбор метода хранения информации.
- Выбор языка программы.
- Разработка архитектуры программы.
- Разработка структуры базы данных.
- Создание методов получения и обработки. информации
- Создание графического интерфейса

Для хранения информации лучше всего подойдет MongoDB — документо-ориентированная база данных реализующая подход NoSQL. Рассмотрим этот подход и саму базу данных.

История NoSQL начинается со встречи, на которой планировалось обсудить новшества в хранении данных. Вывеска должна была содержать короткий и ясный термин, которым и стал NoSQL. Этот термин чаще всего трактуется как Not Only SQL (не только SQL). Это означало, что с того времени индустрия хранения данных начала двигаться в сторону от реляционных баз данных.

Одним из основных отличий является требования к реляционным и нереляционным СУБД. Традиционные СУБД ориентируются на требования ACID (Атомарность, согласованность, изолированность, надежность), в то же время NoSQL может требоваться набор свойств BASE (базовая доступность, гибкое состояние, согласованность в конечном счете). Для информационной системы хранения информации о публикациях идеальным выбором будет документо-ориентированная СУБД. Самым известным представителем таких систем является MongoDB.

MongoDB – документо-ориентированная система управления базами данных (СУБД) с открытым исходным кодом, не требующая описания схемы таблиц. В основе структуры MongoDB лежат коллекции, в которых содержатся документы. MongoDB имеет ряд преимуществ, среди которых поддержка индексации, репликации и агрегации.

Важнейшим инструментом создания информационной системы является язык программирования. Основными критериями выбора языка для данной информационной системы являются наличие возможности работы с СУБД MongoDB, а также с Bib-файлами и API Elsevier. Язык Python идеально подходит, так как имеет уже написанные библиотеки для всех поставленных задач.

Python – высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода.

Преимуществами языка Python являются наличие множества библиотек, реализующих нужные функции, большое сообщество, помогающее в решении

трудных задач, динамическая типизация, и направленность синтаксиса языка на читаемость кода. Отличительной особенностью языка Python является крайне удобная система управления пакетами pip. Разработчику не требуется ничего скачивать, распаковывать и перемещать,нужно только ввести команду «pip install [название библиотеки]» в консоли, после чего автоматически произойдет скачивание и установка последней версии пакета. Спорными моментами в работе с Python можно считать то, что невозможно стандартными средствами создать исполняемый файл, т. е. для того, чтобы запустить программу на другом компьютере, на нем необходимо установить и настроить среду исполнения. Для этой задачи уже написана сторонняя библиотека PyInstaller, но она имеет свои недостатки и особенности. Еще одна особенность, что без настройки виртуальной среды (virtualenv) все проекты будут иметь зависимости от всех загруженных пакетов.

Важным компонентом информационной системы является GUI – графический интерфейс пользователя. В связи с тем, что существует множество библиотек со схожим функционалом, необходимо определить наиболее подходящую. Основными критериями выбора библиотеки будут являться функционал, кроссплатформенность, условия использования и наличие документации. Вторичными признаками можно считать популярность и простоту использования. Было произведено сравнение следующих библиотек: PyQT, TKinter, Kivy, Flexx, wxPython. Первоначально была выбрана библиотека flexx как перспективная и удовлетворяющая всем параметрам. В процессе использования данной библиотеки были выявлены существенные недостатки, осложняющие дальнейшее использование и повлекшие пересмотр решения о выборе GUI-библиотеки. Чтобы использовать библиотеку так, как задумывали авторы, необходимо реализовывать свои виджеты, наследуя их от класса flx.widget, а это подразумевает работу на языке javascript. В реализуемых классах возникает много спорных моментов, по которым нет пояснений в документации и интернете. Для передачи данных нельзя использовать стандартные поля, необходимо создавать свойства, такие как IntProp или StringProp. Для передачи таких свойств необходимо дополнительно создавать прокси классы. Изменять эти свойства можно только в методе mutate.

Совокупность схожих условностей, созданных авторами, приводит к увеличению объёмов кода и потере его читаемости.

В результате были отобраны PyQT и Kivy. После анализа преимуществ и недостатков для дальнейшей реализации системы была выбрана библиотека kivy, так как она проще в освоении, имеет более подробную документацию и на ней выше скорость разработки.

В третьем разделе была осуществлена поэтапная разработка программного обеспечения. Первоначально для описания структуры реализуемого программного обеспечения были созданы следующие диаграммы:

- IDEF3 диаграмма описывает технологические процессы происходящие в системе;
- UML диаграмма вариантов использования описывает отношения между актерами информационной системы и прецедентами;
- UML диаграмма последовательности показывает поток событий, который происходит в информационной системе;
- UML диаграмма классов демонстрирует структуру иерархии классов системы, их кооперацией, атрибутов, интерфейсов, методов и взаимосвязей между ними.

Следующим шагом разработки было создание структуры базы данных. В соответствии с поставленными задачами, необходимо организовать хранение информации о публикациях и написавших их авторах. Для этого хватит двух коллекций. Коллекция документов будет содержать полный библиографический элемент, количество цитирований документа, время последнего обновления и список авторов со ссылками на вторую коллекцию. В коллекции авторов необходимо содержать как постоянную, так и изменяемую информацию. Коллекция авторов будет содержать несколько идентификаторов автора (ScopusID, ORCID и т.д.), личную информацию (ФИО, альтернатиные написания имени), вычисляемую информацию: количество цитирований, индекс Хирша и т.д. Преимуществом MongoDB является отсутствие необходимости задавать жесткую структуру коллекций, в соответствии с чем в записи можно безболезненно добавлять новые поля по мере необходимости.

После создания структуры БД были написаны методы, отвечающие за загрузку данных из онлайн источников и файлов формата BibTex. В разделе

также описаны методы сохранения информации в файлы форматов PDF и BibTex. Модуль получения информации – важнейший в разрабатываемой информационной системе. Для реализации получения информации необходимо создавать методы в соответствии со структурой информации, полученной из конкретного источника.

При получении данных с сайта Scopus присутствует ограничение на количество выдаваемых статей. Для того чтобы считать всю информацию необходимо реализовать максимально эффективные постраничные запросы. Для получения наукометрических показателей автора используется другой подход. Эта информация бесплатно представлена на странице автора, поэтому с помощью библиотеки BeautifulSoup загружается контент страницы, после чего информация получается из заранее указанных элементов страницы.

При получении данных с сайта Publons ответ сайта выдает только часть результатов и ссылку на следующую страницу. Метод загрузки данных с этого сайта немного отличается от предыдущего, он последовательно считывает возвращаемые данные, в которых получает ссылку на следующую страницу и десять статей, записывает эту информацию и таким образом продвигается до последней страницы.

Для загрузки данных из файлов формата BibTex была использована библиотека BibTexParser. Реализуемый метод на входе получает url папки с файлами, а на выходе - json обьект включающий все считанные публикации. Основной сложностью при реализации метода является создание парсера, который бы делал правильное форматирование и разделение авторов, и добавлял другие необходимые поля. Система считывает все статьи, на которые ссылается документ, переводит их в формат JSON, при этом проводя операции над полями авторов, разделяя их, и затем записывает статьи в БД.

Для добавления полученной информации в базу данных в одинаковом формате был использован сервис DOI. У публикаций из Scopus и Publons присутствует поле doi. В результате правильно структурированного запроса на сайт DOI возвращается информация о статье в необходимом нам формате BibTex, которую мы можем записать в базу данных. В процессе записи информации в базу данных производится проверка на возможное присутствие

данной статьи с целью исключения дублирования и параллельное обновление коллекции авторов.

Метод сохранения отчета является комплекстным и задействует большинство модулей информационной системы. Сначала происходит считывание информации о необходимом авторе. Далее происходит подсчет следующей информации:

- Индекс Хирша (h-индекс): назван по фамилии ученого, предложившего этот показатель для оценки научной деятельности). Это количественная характеристика продуктивности ученого, основанная на количестве публикаций и количестве цитирований этих публикаций: ученый имеет индекс h, если он опубликовал h статей, на каждую из которых сослались как минимум h раз.
- Индекс i10 – количественная характеристика продуктивности ученого, основанная на количестве публикаций имеющих больше 10 цитирований.
- Общее количество цитирований;

Для создания диаграм используется библиотека matplotlib. С использованием указанной библиотеки формируется диаграмма распределения среднего количества цитирований по годам.Далее с помощью функции «savefig» диаграмма сохраняется в файл.

Для создания файла PDF используется библиотека FPDF. Для добавления информации в файл создаются клетки (cells) фиксированного размера, и в них помещается необходимая информация. Добавление диаграммы происходит с помощью указания пути к ранее сгенерированному изображению в функции «pdf.image». В разделе «Разработка графического интерфейса» описан процесс создания интерфейса реализумой системы.

Последним этапом создания информационной системы было написание графического интерфейса пользователя. Для реализации графического интерфейса необходимо определить информацию, которую он будет отображать, и методы, которые он будет реализвывать. В соответствии с задачами информационной системы, в GUI должны присутствовать списки авторов и публикаций и окно отображения авторов. Система должна поддерживать такие взаимодействия, как добавление, изменение и удаление информации об

авторах и статьях, экспорт библиографической информации в файл BibTex и наукометрических данных в файл PDF.

В результате работы был создан графический интерфейс, основанный на вкладках. Основная вкладка содержит список авторов, находящихся в системе, кнопки дополнения списка из разных источников и кнопку поиска публикаций. Дополнительные вкладки могут содержать отсортированные списки публикаций, которые готовы к сохранению в файл формата BibTex, страницу конкретного автора. Страница автора содержит кнопки редактирования информации, перехода к списку публикаций автора и сохранения наукометрической информации в файл.

Заключение. В результате выполнения магистерской работы была реализована информационная система хранения и обработки информации о научных публикациях. В процессе реализации информационной системы были получены практические и теоретические навыки, такие как анализ предметной области, построение диаграмм, написание информационных систем. Получены практические навыки работы с MongoDB, языком программирования Python и множеством библиотек, таких как BibTexParser, PyMongo и requests. Реализованная информационная система выполняет все поставленные перед ней задачи и может быть использована для систематизации, хранения и обработки библиометрической информации и помощи в формировании списка литературы в формате BibTeX.