

ПРОЕКТИРОВАНИЕ МОДУЛЯ ИНТЕГРАЦИИ ОНЛАЙН ПРОСМОТРА И РЕДАКТИРОВАНИЯ ДОКУМЕНТОВ

DESIGNING THE MODULE FOR ONLINE DOCUMENT VIEWING AND EDITING INTEGRATION

V. Popukaylo
M. Hadzhi

Summary: This article discusses the tools that will be used for designing a module for online document viewing and editing in a project management system. The relevance of the topic is justified, and existing tools that can solve the set task are examined. An overview of technologies used for implementation is provided, and the choice of tools is justified. The article describes the general structure of the integration module for an existing project management information system and explains the sequence of actions when events for viewing and editing a file are triggered in the client application. Structural diagrams illustrating the interaction between the client, server, and third-party applications are also presented. Overall, the article emphasizes the importance of using structured programming and algorithm decomposition for efficient software development.

Keywords: information systems, the development of an integration module, the WOPI protocol, online document viewing and editing, and Office Online.

Попукайло Владимир Сергеевич

кандидат технических наук, ГОУ «Приднестровский
государственный университет им. Т.Г. Шевченко»,
Молдова, Тирасполь
vsp.science@gmail.com

Хаджи Марк Сергеевич

аспирант, ГОУ «Приднестровский государственный
университет им. Т.Г. Шевченко», Молдова, Тирасполь

Аннотация: В данной статье рассматриваются инструменты, с помощью которых будет производиться проектирование модуля интеграции онлайн просмотра и редактирования документов для системы управления проектами. Обоснована актуальность рассматриваемой темы, а также рассмотрены существующие инструменты, которые позволяют решить поставленную задачу, представлен обзор технологий, использующихся для реализации и обоснован выбор инструментов. Статья описывает общую структуру модуля интеграции для существующей информационной системы управления проектами и объясняет последовательность действий при вызове событий просмотра и редактирования файла в клиентском приложении. Также представлены структурные схемы, описывающие взаимодействие между клиентом, сервером и сторонними приложениями. В целом, статья подчеркивает важность использования структурного программирования и декомпозиции алгоритмов для эффективной разработки программного обеспечения.

Ключевые слова: информационные системы, разработка модуля интеграции, протокол WOPI, онлайн просмотр и редактирование документов, Office Online.

Введение

В наши дни, практически каждый человек сталкивается с необходимостью работы с документами. Документированная информация составляет основу управления, его эффективность в значительной степени базируется на производстве и потреблении информации. В современном обществе информация стала полноценным ресурсом производства, важным элементом социальной и политической жизни общества.

Говоря об электронном документе, необходимо понимать, что этот документ может проходить все стадии своего жизненного цикла или часть из них в электронно-цифровом формате.

С каждым годом скорость обмена данными в сети интернет, технические характеристики персональных компьютеров и мобильных телефонов возрастают, что позволяет взаимодействовать с документами посредством браузера, без использования дополнительного программного обеспечения.

На данный момент, некоторые веб-приложения предоставляют возможность встраивания в другие продук-

ты. Данная возможность позволяет пользователям взаимодействовать с документами в рамках одной вкладки браузера.

В данной статье будет описано проектирование интеграционного модуля, который позволит пользователям системы управления проектами просматривать и редактировать документы пакета *Microsoft Office*.

Существует большое количество инструментов и способов взаимодействия с электронными документами, но в последние года, с ростом вычислительных способностей персональных компьютеров, телефонов и других гаджетов, а также с увеличением скорости в сети интернет, набирают популярность приложения, позволяющие взаимодействовать с документами стандартного пакета *MS Office* в браузере. Основным плюсом данных инструментов является то, что их использование не подразумевает установки специализированных программ для работы с документами.

Одними из первых в создании подобных веб-приложений была компания *Google*, которая анонсировала *Google Docs* в июне 2006 года. Данное программное обеспечение стало результатом слияния двух продук-

тов — *Writely* и *Google Spreadsheets*. *Writely* — это отдельный текстовый процессор, который был создан компанией *Upstartle*. *Google Docs* — это продукт, который позволял взаимодействовать с файлами формата *Word*, *HTML*, *PDF*, а также с фалами, содержащими простой текст. После тестирования ограниченным количеством пользователей, данный сервис стал доступен всем владельцам *Google Account*. В 2010 году, для файлов формата *PDF* была добавлена функция оптического распознавания символов.

Следующей после *Google* была компания *Microsoft*, анонсировавшая в 2008 году сервис *Office Web Apps*, который преподносился как облегчённая версия полноразмерных приложений *MS Word*, *Excel*, *PowerPoint* и *OneNote*. Данный продукт позволял создавать и редактировать документы, используя только браузер. Официально *Office Web Apps* был представлен в сентябре 2009 года, а финальная версия, доступная для всех пользователей, появилась в июне 2010 года. В конце 2013 года функционал данного продукта был расширен — появилась возможность совместного редактирования и автосохранения файлов *Word*. Данные инструменты значительно облегчили взаимодействие с документами для обычных пользователей.

Для того чтобы выполнить поставленную задачу, необходимо рассмотреть существующие веб-ориентированные программные продукты, которые позволяют работать с электронными документами, имеют открытый *API* (от англ. *Application Programming Interface* — программный интерфейс приложения) и хорошую документацию.

Google Docs Editors — программный продукт, разработанный компанией *Google*. Данный инструмент предоставляет возможность создания и редактирования текстовых документов, электронных таблиц, форм, рисунков и презентаций. При работе с файлами, все изменения сохраняются в облако, что практически исключает возможность потери данных. Рассматриваемое решение предоставляет возможность работы в автономном режиме. При восстановлении интернет-соединения, все изменения сохраняются в автоматическом режиме.

OnlyOffice — программный продукт с открытым исходным кодом, разработанный компанией *Ascensio System SIA*. Данное решение является модульным и включает в себя четыре офисных веб-приложения, которые позволяют работать с документами, презентациями, таблицами и формами, систему управления проектами, корпоративной социальной сети и почтового сервиса. Базовый пакет данного продукта является бесплатным как для личного, так и для корпоративного использования. Модуль, отвечающий за документы, представляет собой централизованную систему управления докумен-

тацией и позволяет управлять доступом к файлам и подключением различных облачных хранилищ.

В отличие от большинства конкурентов, данный продукт написан на *JavaScript* с использованием элемента *Canvas*. Это позволило добиться абсолютно одинакового отображения документов во всех браузерах, не зависимо от операционной системы. Данный продукт может полноценно работать только с документами формата *docx*, *xlsx* и *pptx*. При работе с файлами других, поддерживаемых форматов, документы конвертируются в *docx*, *xlsx* или *pptx*.

Помимо веб-версии *OnlyOffice* имеет коробочную версию, которую можно развернуть в локальной сети.

Zoho Office Suite — офисный онлайн пакет, включающий в себя инструменты, позволяющие создавать и редактировать электронные документы, таблицы, базы данных и презентации. Данный инструмент поддерживает файлы форматов, используемых в *Microsoft Office*, *OpenOffice*, но ввиду того, что данное приложение использует собственные форматы, иногда возможна частичная потеря данных при открытии документа. *Zoho* предоставляет открытый и документированный *API*, для большей части инструментов. *MS Office Online* — аналог *Microsoft Office* для персональных компьютеров. Данный продукт позволяет создавать, редактировать документы, а также предоставлять к ним доступ другим пользователям. Также, данный инструмент позволяет осуществлять совместное редактирование документов двумя и более пользователями. Решение включает в себя веб-версии *Word*, *Excel*, *PowerPoint* и *OneNote*. Интерфейс и горячие клавиши облачной версии схожи с *desktop* приложением, но часть функционала недоступна.

Также, *Microsoft* предоставляет интерфейс прикладного программирования, который позволяет разработчикам других приложений интегрировать офисный пакет для удобства пользователей.

Одним из главных недостатков данного инструмента является то, что достаточно сложные документы могут выглядеть и распечатываться по-разному в разных браузерах.

Выбор методов и средств реализации

В ходе выполнения работы разрабатывается модуль, который будет встраиваться в существующий продукт. Для решения поставленной задачи была выбрана интегрированная среда разработки *IntelliJ IDEA* [1] (комплекс программных средств, использующийся для написания, исполнения и оптимизации кода) от компании *JetBrains*.

Для реализации модуля интеграции необходимо выбрать язык программирования. Учитывая, что данная

работа выполняется на базе существующей системы управления проектами, используемые в данном продукте языки программирования необходимо рассмотреть и сравнить между собой. Серверная часть состоит из двух проектов, один из которых написан на *PHP*, а другой на *Java*.

Java — строго типизированный, высокоуровневый, объектно-ориентированный язык программирования общего назначения [2]. Данный язык был разработан компанией *Sun Microsystems*, которая в последующем была приобретена компанией *Oracle*. Приложения, написанные на *Java*, обычно транслируются в байт код, что позволяет им работать на любой архитектуре, для которой существует реализация *JVM (Java Virtual Machine)*. С середины 1990-х годов язык получил широкое распространение и стал активно использоваться для написания клиентских приложений.

Важной особенностью *Java* является система безопасности, которая контролируется виртуальной машиной. Такой подход позволяет предотвращать операции, которые не входят в список разрешённых операций.

К недостаткам концепции виртуальной машины относят снижение скорости исполнения байт-кода виртуальной машиной. Программы, написанные на *Java*, обычно медленнее, чем написанные на Си и занимают больше оперативной памяти. Однако, если сравнивать данный язык программирования с интерпретируемыми языками, то быстроедействие *Java* обычно заметно выше.

Для серверной части веб-приложения, этот язык программирования обычно используется в крупных приложениях, заказчиками которых является крупный бизнес.

PHP — это высокоуровневый, интерпретируемый язык программирования со слабой динамической типизацией [3]. Интерпретируемым языком программирования называется язык, исходный код которого выполняется методом интерпретации (операторы программы транслируются и выполняются друг за другом). *PHP* является языком программирования общего назначения и сконструирован таким образом, чтобы в его код мог вставляться *HTML*.

Для того, чтобы не увеличивать количество языков программирования и технологий, решать поставленную задачу необходимо с помощью одного из используемых языков.

Учитывая данное сравнение, можно сделать вывод, что для решения поставленной задачи больше подходит язык программирования *Java*. Модуль, написанный на данном языке программирования будет быстрее, безопаснее и надёжнее.

Так как информация о возможных вариантах взаимодействия с *Office Online* хранится в *XML* документе, то необходимо рассмотреть, что такое *XML* и какие инструменты используются для работы с ним.

XML (от англ. *eXtensible Markup Language*) — расширяемый язык разметки — это язык программирования, предназначенный для создания логической структуры данных, хранения и передаче этих данных [4]. Данный язык является рекомендацией сообщества *W3C* [5] в качестве языка разметки общего назначения. Данный язык относится к категории неопределённых, то есть, теги должны быть определены пользователем.

Для того, чтобы работать с *XML* документом, необходимо выбрать готовую библиотеку на *Java*.

Современная версия языка программирования *Java* поставляется с четырьмя методами для работы с языком расширенной разметки:

DOM Parser/Builder — класс, подходящий для работы с небольшими документами. Основным достоинством и недостатком является то, что данный класс загружает весь *XML* документ в оперативную память. Данный парсер загружает весь файл в древовидную структуру и работает с ней как с *DOM* деревом.

Sax Parser — отличается от *DOM Parser* тем, что не загружает полный *XML* в память, а проводит разбор документа построчно и вызывает различные события, соответствующие определённым сущностям и элементам. Этот анализатор рекомендуется использовать при работе с большими документами, при небольшом объёме оперативной памяти.

StAX Reader/Writer — инструмент, работающий с интерфейсом, ориентированным на потоки данных. Основным отличием от *SAX Parser* является то, что данный синтаксический анализатор предоставляет поддержку текущего положения курсора в документе, что позволяет извлекать содержимое, доступное для курсора, тогда как *SAX* выдает события по мере появления определенных данных.

JAXB — является новейшим инструментом для работы с *XML*-документами из представленных. Данный инструмент позволяет переводить сущности *XML*-документов в *Java* классы и наоборот. *JAXB* включает в себя класс *JAXBContext*, который предоставляет *API* для работы с документами.

Из рассмотренных решений был выбран *JAXB*, ввиду того, что этот инструмент имеет высокую производительность, хорошую документацию и позволяет легко переходить к работе с классами, что значительно ускоряет процесс разработки.

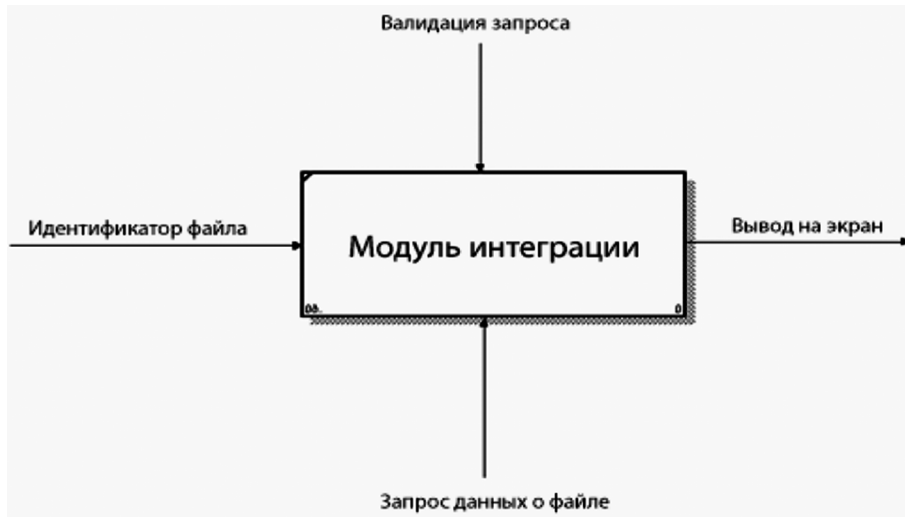


Рис. 1. Контекстная диаграмма верхнего уровня

Для сохранения копии XML-документа на WOPI [6] хосте, необходимо использовать базу данных. В системе управления проектами используется три базы данных, подходящих для решения данной задачи:

MySQL — реляционная система управления базами данных, которая распространяется на бесплатной основе. Данная СУБД обладает достаточно разнообразным набором инструментов, доступных для создания приложений.

Redis — резидентная система управления базами данных класса NoSQL. Данный продукт хранит данные в формате «ключ-значение» и ориентирован на максимальную производительность. По заявлению разработчиков, Redis способен обрабатывать до ста тысяч запросов в секунду. В отличие от большинства нереляционных баз данных, этот продукт поддерживает большее количество типов данных.

MongoDB — это документоориентированная СУБД, которая не требует описания схемы таблиц. Считается одним из классических примеров NoSQL-систем. В основе данной системы управления базой данных лежит документ, являющийся простейшим типом данных в рассматриваемом продукте.

После рассмотрения используемых хранилищ, было принято решение использовать в качестве временного хранилища копии XML-документа нереляционную базу данных Redis. Данная СУБД достаточно быстро обрабатывает запросы и позволяет устанавливать TTL для ключа.

Основная задача, которая состоит перед разработчиком — это создание такого программного продукта, который помог бы пользователю выполнять требуемые задачи. Для реализации такого программного обеспечения вначале требуется определить и точно понять, каким

образом функционирует данный продукт. При создании модели сначала необходимо изобразить самый высокий уровень — действие контекста.



Рис. 2. Блок-схема работы модуля интеграции

Контекстная диаграмма, отображенная на рисунке 1, изображает деятельность самого верхнего уровня и обозначает границу моделирования относительно цели, возможностей и точки зрения.

Блок-схема изображена на рисунке 2. Блок-схема алгоритма — графическое изображение алгоритма в виде связанных между собой с помощью стрелок (линий перехода) и блоков — графических символов, каждый из которых соответствует одному шагу алгоритма. Внутри блока дается описание соответствующего действия.

Блок-схемой называют графическое представление алгоритма, в котором он изображается в виде последовательности связанных между собой функциональных блоков, каждый из которых соответствует выполнению одного или нескольких действий.

Структурная схема — это такая схема, которая отражает состав и взаимодействие по управлению частей разрабатываемого программного приложения. Под элементарным звеном понимают подструктуру, которая реализует ту или иную простейшую, элементарную функцию.

Структурная схема программного продукта изображена на рисунке 3.

Разделение программы на мелкие фрагменты — структурная декомпозиция. При ней отдельные части программы (модули) представляют собой совокупность процедур для решения одной или нескольких задач.

Одна из основных особенностей структурного программирования заключается в том, что оно позволило создавать библиотеки подпрограмм (процедур), которые можно было бы использовать повторно в различных проектах или в рамках одного проекта, при этом не затрачивая времени на создание данного модуля снова.

Общая структура модуля интеграции представлена на рисунке 4.

Когда пользователь вызывает событие просмотра или редактирования файла, клиентское приложение отправляет запроса на сервер, результатом которого является формирование токена доступа к файлу и данные для *iframe*. Далее, клиент *WOPI* обращается к серверу и запрашивает необходимые данные (название, расширение, размер, имя текущего пользователя, его права и т.д.).

Следующий шаг — получение клиентом двоичного содержимого файла. Если в ответе на предыдущий запрос был предоставлен *URL*-адрес файла, то файл будет скачан с *AWS*. В противном случае, клиент *WOPI* отправит ещё один запрос, ответом на который будет контент файла.

Последний этап — отображение файла в браузере пользователя.

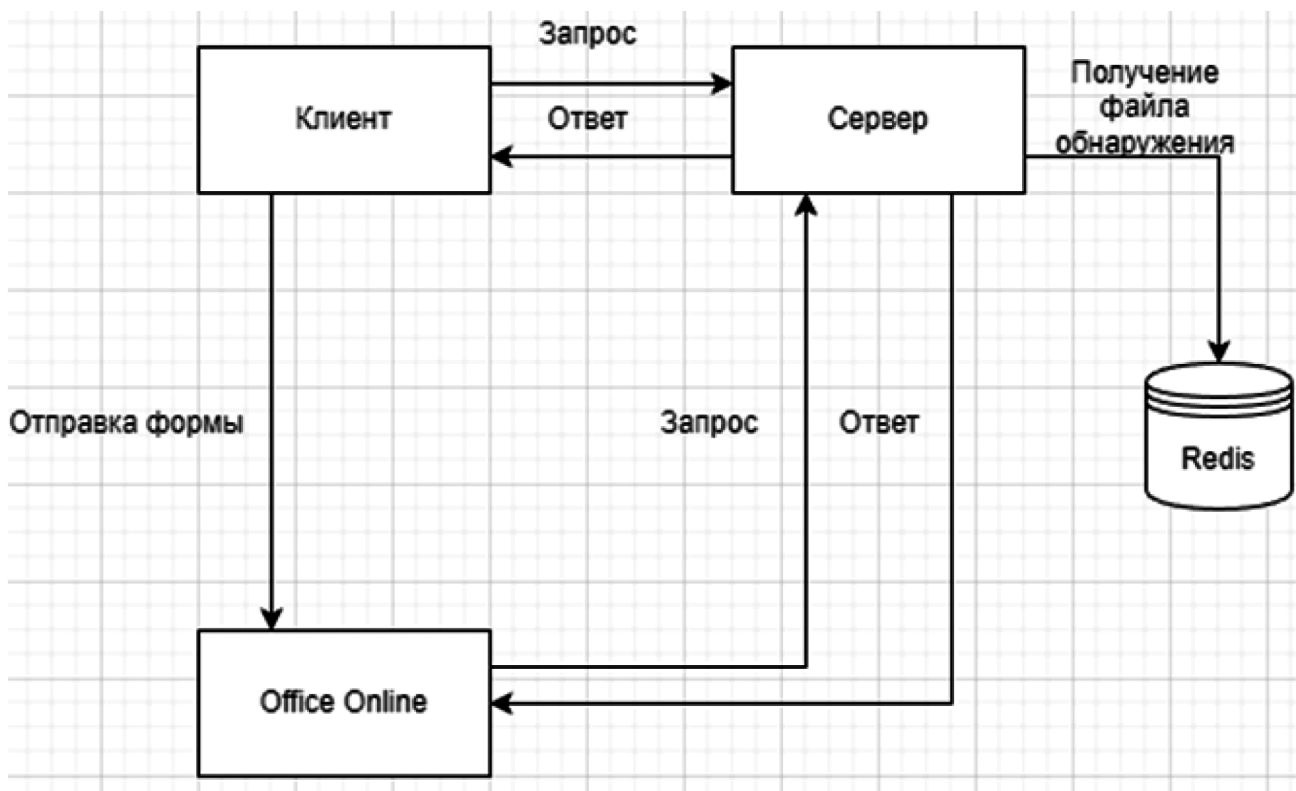


Рис. 3. Структурная схема программного продукта

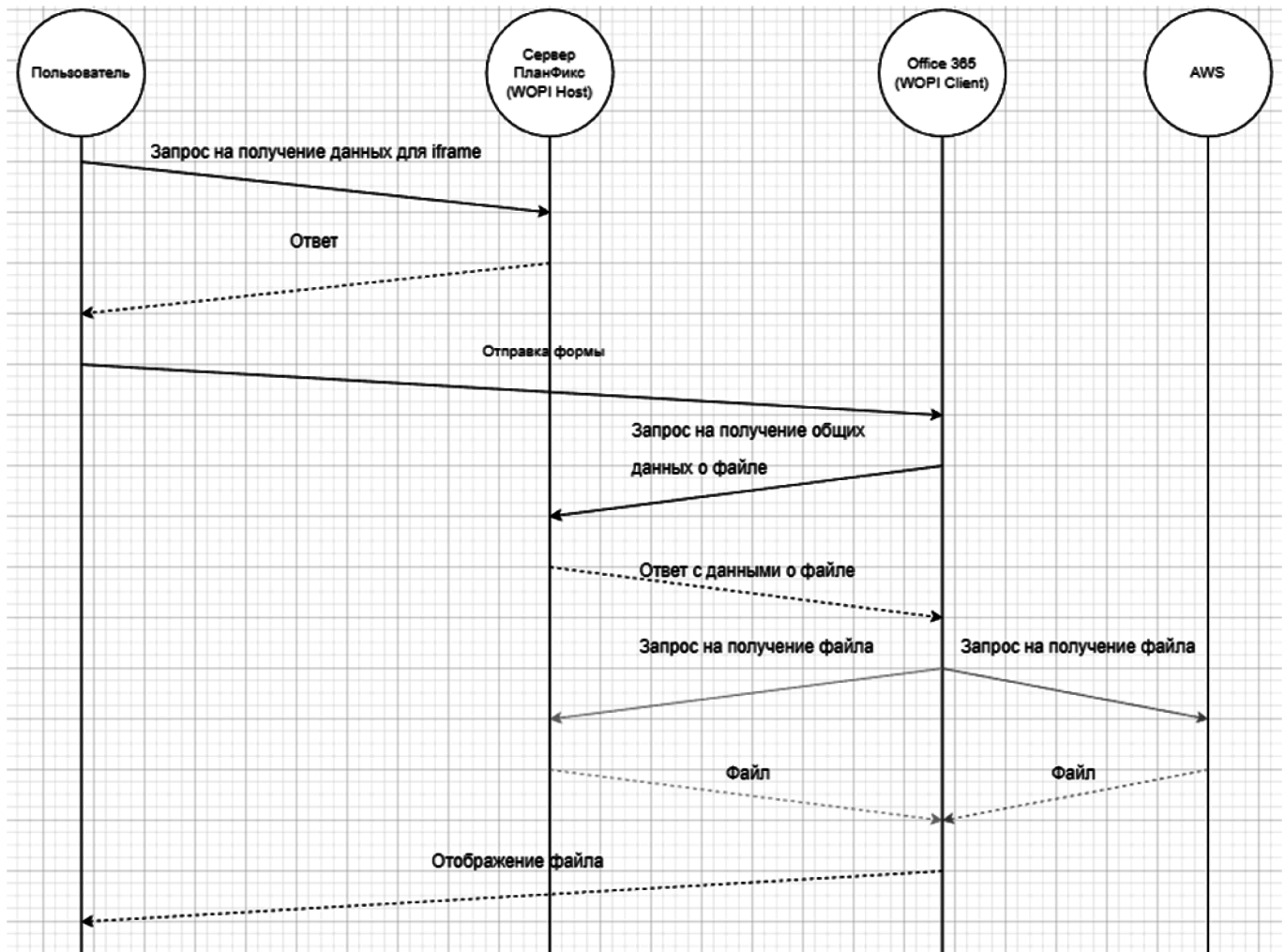


Рис. 4. Структурная схема модуля интеграции

Блок-схема работы разрабатываемого модуля с точки зрения пользователя системы управления проектами представлена на рисунке 5.

При открытии файла, браузер отправляет запрос на сервер за дополнительными данными. Если пользователь открывает файл на редактирование, то ему будет необходимо авторизоваться в *Microsoft Office 365*.

Следующий и последний этап — отображение файла у пользователя на экране. Блок-схема, на которой изображена общая схема обработки запросов с клиентов *WOPI* изображена на рисунке 6.

При поступлении запроса от *Office Online*, необходимо совершить обнаружение *WOPI* и с использованием представленных ключей, верифицировать запрос. В случае, если проверки подлинности подписей завершается неудачей, необходимо вернуть ответ с кодом ошибки 500. Иначе, необходимо по заголовку запроса *X-WOPI-Override* определить тип вызываемой операции. Если данная операция не поддерживается, вернуть ответ

с кодом ошибки 501, а в противном случае — сформировать и вернуть ответ с ожидаемым содержимым и заголовками.

В данной статье были рассмотрены инструменты, необходимые для разработки модуля интеграции онлайн просмотра и редактирования документов. Статья описывает общую структуру модуля интеграции и подробно объясняет последовательность действий при вызове событий просмотра и редактирования файлов в клиентском приложении. Кроме того, были представлены схемы, иллюстрирующие взаимодействие между клиентом, сервером и сторонними приложениями. В целом, подчеркивается важность использования структурного программирования и декомпозиции алгоритмов для эффективной разработки программного обеспечения. Полученные результаты могут быть полезны для разработчиков, занимающихся созданием интегрированных систем онлайн просмотра и редактирования документов, и способствуют более эффективному проектированию и разработке таких модулей.

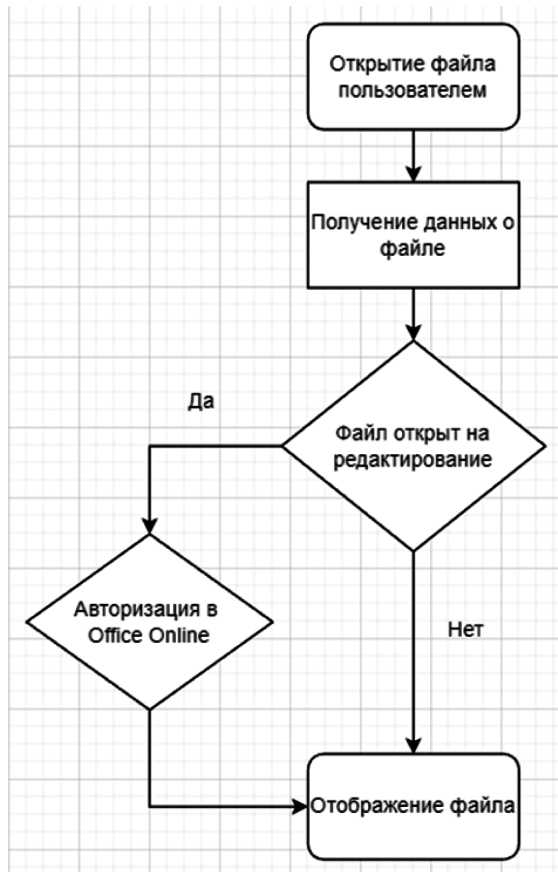


Рис. 5. Блок-схема, отображающая работ модуля глазами пользователя

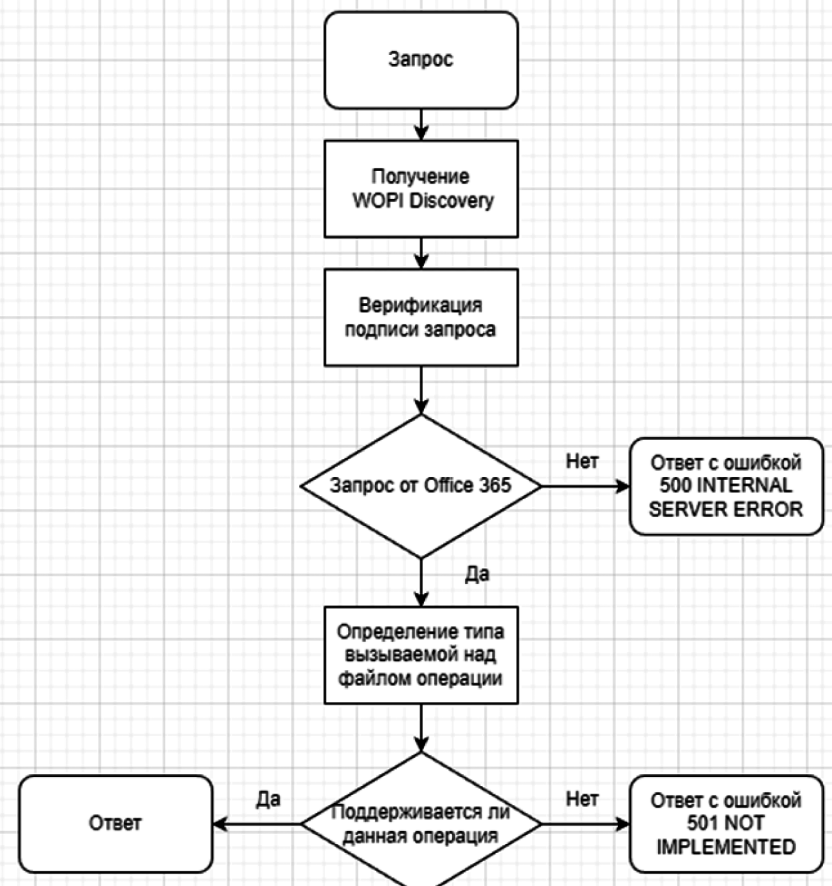


Рис. 6. Блок-схема обработки запросов сервером

ЛИТЕРАТУРА

1. Программный продукт IntelliJ IDEA -Фэлла М. Стильный Java. Код, который работает всегда и везде. СПб.: Питер, 2021. — 352 с.
2. Джош Л. Современный PHP. Новые возможности и передовой опыт. М.: ДМК-Пресс, 2016. — 304 с.
3. Хантер Д., Фаусетт Д., Рафтер Д. XML. Базовый курс. К.: Диалектика, 2018. — 1344 с.
4. W3C — консорциум всемирной паутины. Режим доступа: <https://www.w3.org/> (дата обращения 17.06.2023)
5. Документация по файлу обнаружения WOPI. Режим доступа: <https://docs.microsoft.com/en-us/microsoft-365/cloud-storage-partner-program/online/discovery> (дата обращения 17.06.2023)

© Попукайло Владимир Сергеевич (vsp.science@gmail.com); Хаджи Марк Сергеевич
Журнал «Современная наука: актуальные проблемы теории и практики»