2 Моделирование предметной области и разработка функциональных требований

2.1 Описание функциональности ПС

Для представления функциональной модели была выбрана диаграм-ма вариантов использования UML, которая отражает отношения между актерами и прецедентами и позволяет описать систему на концептуальном уровне. Прецедент соответствует отдельному сервису системы, определяет один из вариантов её использования и описывает типичный способ взаи-модействия пользователя с системой. UML предназначен для определения, визуализации, проектирования и документирования программных систем.

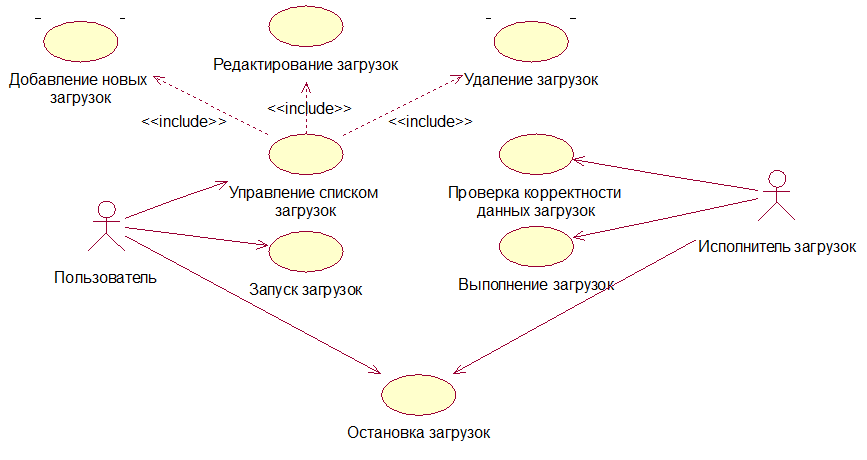


Рисунок 2.1 – Диаграмма вариантов использования

Диаграмма вариантов использования разрабатываемого программ-ного средства представлена на рисунке 2.1. На диаграмме можно выделить два основных составляющих элемента – актер и прецедент. Актер – стили-зованный человек, обозначающий набор ролей пользователя, взаимодей-ствующего с некоторой сущностью. Прецедент – эллипс с надписью, обо-значающий выполняемые системой действия, приводящие к наблюдаемым актером результатам.

На основании представленной диаграммы вариантов использования можно сделать вывод, что в системе будет существовать два основных актера: пользователь, исполнитель загрузок.

Рассмотрим каждый из прецедентов более подробно для каждого ак-тера.

2.1.1 Пользователь

Пользователю предоставляются следующие возможности:

1. управление списком загрузок, что подразумевает добавление новых загрузок в список, удаление загрузок из него, а также редактирование данных загрузок;
2. запуск загрузок: пользователь принимает решение о том, когда необходимо отправить загрузку на исполнение в Исполнитель загрузок;
3. остановка загрузок: пользователь может прекратить исполнение загрузки для того, чтобы освободить вычислительные ресурсы или отложить на время ее исполнение.

2.1.2 Исполнитель загрузок

Исполнитель загрузок – это фоновый процесс программного средства, который отвечает за проверку и исполнение загрузок. Пользователь не имеет доступа к возможностям исполнителя, однако может влить на ход его работы через управление списком загрузок. Исполнитель загрузок выполняет следующие функции:

1. проверка корректности данных загрузок: исполнитель загрузок проверяет возможность осуществления загрузки, и сообщает пользователю о об ошибках в данных загрузки, если они есть;
2. исполнение загрузок: после проверки корректности данных, исполнитель выполняет загрузку и информирует пользователя о ее завершении
3. остановка загрузок: в случае обнаружения ошибки или завершения, исполнитель исключает загрузку из очереди на исполнение.

2.2 Спецификация функциональных требований

На основании анализа исходных данных для проектируемого про-граммного средства можем выделить, что основной целью является созда-ние качественного программного продукта, позволяющего решить суще-ствующие проблемы, такие как:

* эффективное управление расписанием загрузок;
* эффективное использование вычислительных ресурсов при выполнении загрузок.

В ходе разработки будут реализованы следующие возможности:

1. редактирование списка планируемых загрузок;
2. редактирование данных загрузок;
3. проверка возможности осуществления загрузки;
4. формирование очереди исполнения загрузок;
5. исполнении загрузок;
6. прохождение аутентификации и авторизации, если это необходимо для осуществления загрузки;
7. сохранения истории совершенных загрузок;
8. сбор данных о количестве принятых данных.

Программное средство будет доступно всем пользователям компьютера.

3 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА

3.1 Разработка программной архитектуры

Прежде чем приступить к разработке программного средства, необходимо определить основные программные модули, которые будут отвечать за функциональные возможности приложения.

Исходя из функциональных требований, можно описать модули, которые будут реализованы в программном средстве:

1. база данных: база данных необходима для долговременного хранения данных загрузок, включая данные о ходе их выполнения исполнителем загрузок;
2. графический интерфейс пользователя, который необходим для эффективного взаимодействия пользователя с программным средством и позволит уме в полной мере использовать все функции программного средства
3. сетевой модуль, который будет осуществлять взаимодействие с удаленными узлами сети;
4. модуль бизнес логики, который будет отвечать за обработку команд пользователя, а также управлять работой сетевого модуля.

Поскольку разрабатываемое программное средство является программой для персонального использования, то нет никаких особых требований к его процессу его развертывания.

3.2 Разработка модели базы данных

Неотъемлемой частью конечного программного средства является база данных, используемая системой в процессе работы. Информационную модель предметной области можно представить на языке IDEF1X.

В модели использованы три типа связей: не идентифицирующая ноль-или-один-ко-многим, не идентифицирующая многие-ко-многим и идентифицирующая один-ко-многим. Первая обозначена штриховой линией с ромбом со стороны родительской сущности и кружком со стороны дочерней сущности, вторая – штриховой линией с ромбом со стороны родительской сущности и кружком со стороны дочерней сущности, третья – линией с кружком со стороны дочерней сущности. При наличии связи «один ко многим» одна запись в одной таблице связана с множеством записей в другой таблице. Связь между таблицами организуется на основе общего поля. На стороне «один» должно выступать ключевое поле, содержащее уникальные значения – такое поле называют внешним ключом. Значения на стороне «многие» могут повторяться.

Организация связей между таблицами обеспечивает целостность данных. Система не допустит, чтобы одноименные поля в разных таблицах имели разные значения. Ввод данных автоматически контролируется. Рассмотрим сущности по-отдельности.

3.2.1 Сущность «Загрузка»

Представляет данные о загрузке, планируемой к выполнения, выполняющейся или уже выполненной. Имеет следующие поля

* Идентификатор загрузки – идентификатор загрузки(первичный ключ);
* Название – название данные загрузке пользователем, для того, чтобы пользователю было легче найти загрузку в списке;
* Папка сохранения – путь к папке, куда будут сохранятся все загруженные файлы;
* Приоритет – приоритет загрузки при исполнении;
* Статус – отображает статус загрузки в текущий момент;
* Дата создания – дата внесения пользователем загрузки в список загрузок;
* Логин – логин для авторизации на удаленном сервере, если это потребуется при выполнении загрузки;
* Пароль – пароль для авторизации на удаленном сервере, если это потребуется для выполнения загрузки;
* Ошибка – описание ошибки выявленной при попытке выполнения загрузки.

Сущность имеет связь «один ко многим» с сущностью «Файл».

3.2.2 Сущность «Файл»

Представляет данные о конкретном файле который будет скачан в ходе выполнения загрузки. Имеет следующие поля:

* Идентификатор файла – идентификатор файла(первичный ключ);
* Идентификатор загрузки – внешний ключ на сущность «Загрузка»;
* Ссылка – ссылка по которой можно загрузить файл;
* Статус – текущий статус файла;
* Размер – размер файла на удаленном сервере;
* Количество загруженных байт – количество информации, загруженной из файла на удаленно компьютере на данный момент;
* Ошибка – описание ошибки выявленной при попытке загрузить файл.

Сущность имеет связь «Многие к одному» с сущностью «Загрузка».

3.3 Разработка алгоритма выполнения загрузки

Выполнение загрузки состоит из проверки загрузки и последовательной загрузки всех указанных файлов. Проверка загрузки состоит из проверки того, что что папка назначения существует, что программа может записывать файлы в нее, и проверки каждого файла перед скачиванием.

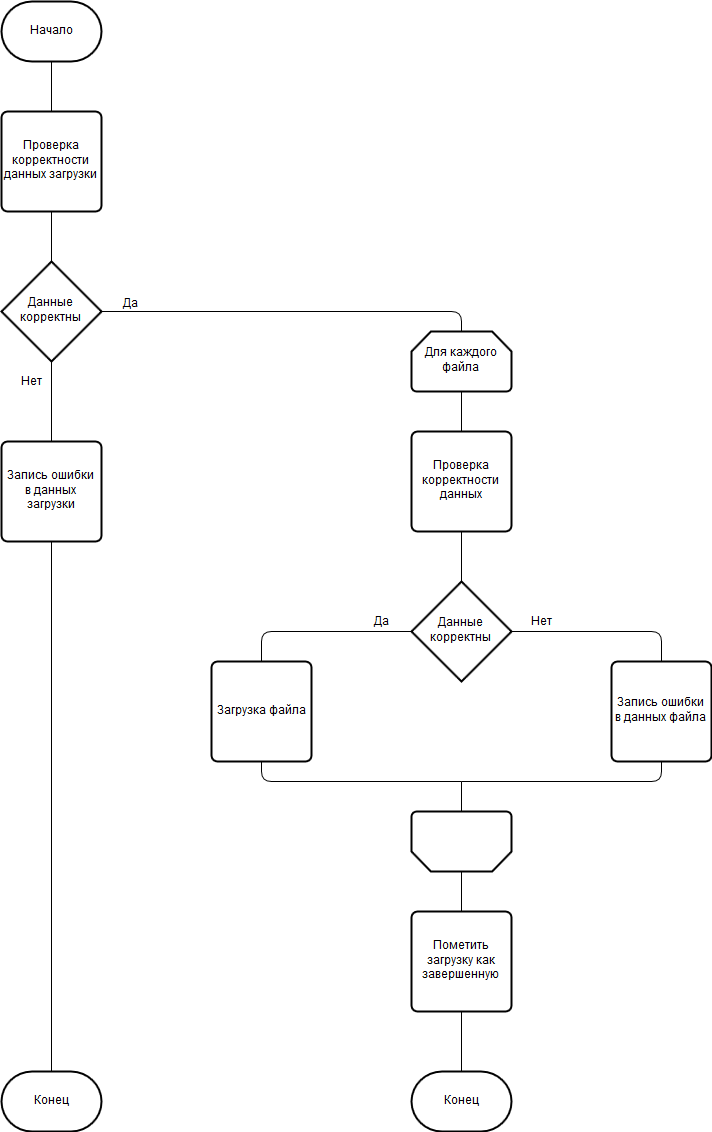


Рисунок 3.1 – Алгоритм выполнения загрузки.

Обнаружение ошибок при проверке загрузки ведет к тому, что загрузка будет удалена из очереди на выполнение. Обнаружение ошибки при скачивании файла приведет к тому, что файл будет помечен как имеющий ошибки, а исполнитель загрузок перейдет к скачиванию следующего файла.

Проверка файла представляет собой проверку доступа к серверу с файлом, наличия файла на удаленном сервере, разрешения на загрузку этого файла, а также того, что файл поместится на жестком диске компьютера.

По завершению загрузки файла, файл будет помечен как загруженный, а после загрузки всех файлов, загрузка будет отмечена как завершенная.