# **2** СИСТЕМНОЕ ПРОЕКТИРОВАНИЕ

Любая программа обладает определенным функционалом. Как правило, выделяются основные возможности программы и дополнительные.

Основные возможности разрабатываемого программного модуля заключаются в следующем:

* авторизация, перед тем как начать пользоваться приложение пользователю необходимо пройти процедуру авторизации, после чего будет создан его учетная запись. Подтверждение авторизации происходит, используя почтовый адрес пользователя.
* выбор начальной папки для работы. Эта папка будет использоваться для построения первоначального дерева физический папок, контент из нее будет загружен на сервер.
* создание виртуального каталога. Пользовать может создать виртуальный каталог и добавить его как подкаталог к любому физическому или виртуальному каталогу. Виртуальные каталоги отличны от физических тем что не создаются в файловой системе и не могут быть просмотрены вне приложения.
* создание физического каталога. Пользователь может создать физический каталог и добавить его к любому физическому каталогу в качестве подкаталога.
* удаление физического или виртуального каталога. Пользователь может удалить любой каталог, который не имеет подкаталогов. Весь контент каталога перемещается в корзину.
* перенос каталога из его текущего каталога в другой каталог. При этом виртуальные каталоги можно переносить без ограничений, физические же можно переносить только в физический каталог.
* изменение физического каталога в виртуальный. Изменить можно только физический каталог, у которого нет ни одного физического подкаталога. После изменения папка удаляется из файловой системы, а ее контент переноситься в кэш приложения или удаляется.
* изменение виртуального каталога на физический. Изменить можно только виртуальный каталог, у которого паррент-каталог является физическим. В файловой системе создается папка с именем каталога, а затем туда копируется файлы каталога.
* сортировка по дате контента. Пользователь может указать начальную и конечную дату для фильтрации и каталог, в котором нужно фильтровать данные.
* добавление тегов к контенту. Пользователь может добавить тэги для удобного поиска и фильтрации в дальнейшем.
* обработка контента. Пользователь может применять различные фильтры для работы с фотографиями, а также может изменять их разрешение. Возможно редактирование информации и имени файлов.
* добавление друзей. Пользователь может отправить запрос на добавление в друзья человека из списка людей.
* отправление сообщений и контента друзьям. Пользователь может отправлять друзьям контент и сообщения.
* добавление комментариев к контенту. Пользователь может добавить комментарий к контенту, пометив важную информацию.
* создание общего альбома. Пользователь может создать альбом и поделить его с несколькими друзьями. Этот альбом будет доступен всем подключенным пользователям, они смогут добавлять туда свой контент.
* отображение содержимого физический и виртуальных каталогов. Пользователь может просматривать содержимое созданных им каталогов через интерфейс приложения. Физические каталоги можно также просмотреть файловым мэнеджером. Виртуальные можно просмотреть только с помощью приложения.

На каждый логически выделенный блок программы возлагаются определенные задачи. Кроме того, каждый блок программы, так или иначе связан с остальными блоками, чтобы обеспечить работоспособность всего приложения в целом. Связь, как правило, реализуется посредством обмена данными между блоками.

*Блок объектной модели базы данных*  представляет собой набор классов. Каждый класс по сути будет представлять таблицу в базе данных а поля класса являются столбцами в таблице. В каждом классе помечается какое поле будет являться первичным ключом в таблице также внешние ключи к другим таблицам. Также помечаются связи между классами они же будут являться связями в таблице. Также он отвечает за структуру классов, на основе которых строится база данных, а также классов, с которыми работают другие блоки. Также в его входит классы для работы с базой данных в виде промежуточного слоя. В блоке также есть классы для рефлексионного преобразования моделей базы данных в модели для внешнего доступа

*Блок обеспечения доступа к базе данных*  отвечает за обработку запросов к базе данных. Запросы ко всем таблицам проходят через данный блок. На вход поступают данные от блока веб сервиса. Этот блок поддерживает все необходимые операции, которые необходимы для быстрой и удобной работы с базой данных, такие как:

* запросы на добавление строки в таблицу
* запросы на удаление строки таблицы
* запросы на обновление строки таблицы
* запросы на получение выборки данных из одной или нескольких таблиц

*Блок веб сервиса* отвечает за работу Wcf сервиса. В нем находиться запущенный Wcf сервис, который обеспечивает доступ к функционалу блоку доступа к базе данных с внешних устройств по сети. Запросы, которые он получает из блока работы с веб сервисом, передаются в блок доступа к базе данных. После обработки и формирования ответа, в виде модели, она проходит серелизацию и отправляется обратно к блоку работы с веб сервисом, где происходит её анализ.

Также блок отвечает за загрузку контента в потоковом режиме на сервер, в котором контент помещается в специальное хранилище, а также из него к клиенту, где блок работы с файловой системой сохраняет его в требуемый каталог системы.

*Блок работы с файловой системой на стороне клиента* отвечает за создание, удаление каталогов, находящихся в коренной папке. Также в случае преобразование виртуального каталога в физический, этот блок отвечает за корректное создание и перемещение контента. Блок при первом запуске собирает информацию о дереве каталогов в коренной папке и на основе этой информации пересылает ее на сервер для сохранения в базе данных. При переносе каталога, создается папка в новом паррент каталоге, затем весь контент переноситься в новую папку, после чего исходная папка очищается. При преобразовании физического каталога в виртуальный, файлы из каталога переносятся в кэш приложения, а затем исходная папка удаляется. При преобразовании виртуального каталога в физический, в паррент папке создается новый каталог с именем виртуального каталога, затем контент копируется туда из кэша приложения, в случаи если в кэше нет необходимого контента, он подкачивается с сервера с использованием блока работы с веб сервисом.

*Блок аутентификации пользователя* отвечает за регистрацию новый пользователей в сети, а также за выполнение входа уже существующими пользователями. Процесс регистрации проходит следующим образом данные: логин, пароль и почтовый ящик регистрирующего приходят от блока пользовательского интерфейса, затем они попадают в блок работы с веб-сервисом, где происходит запрос к веб-сервису, затем проходит их обработка и результат работы возвращается к блоку работы с веб-сервисом, откуда попадает к блоку аутентификации, где в зависимости от результатов обработки пользователь показывается соответствующий результат. В случаи удачной регистрации пользователю отправляется письмо на почту с кодом подтверждения, получив который и введя в модальное окно он завершит регистрацию. Для пользователей уже зарегистрировавшихся, достаточно лишь ввести свои данные и авторизоваться.

*Блок работы с виртуальными каталогами* отвечает за работу с каталогами, которые существуют только виртуально, т. е. физически в файловой системе они не существуют, однако благодоря сохранению всей необходимой информации в базе данных, могут быть представлены в приложении. В данном блоке, используя информацию из блока аутентификации и блока работы с веб сервисом, с сервера загружается вся необходимая информация о текущем пользователе, в том числе и о его физических и виртуальных каталогах. Затем в блоке пользовательского интерфейса строиться дерево каталогов, исходя из данных полученных с сервера. Также этот блок отвечает за создание виртуальных каталогов. Это реализуется с помощью блока пользовательского интерфейса, где пользователь выбирает паррент каталог и имя нового каталога, после чего используя блок работы с веб сервисом данные пересылаются на сервер, где они заносятся в базу данных. При создании физического каталога процесс проходит аналогично, за исключением того то с помощью блока работы с файловой системой на клиентской стороне в паррент папке также создается физическая папка с нужным именем

*Блок работы с файловой системой на серверной стороне* представляет собой иерархию классов, в которых находиться логика работы с файлами файловой системы. Если клиентское приложение начнем синхронизацию файлов между клиентом и сервером, то сначала блок для работы с файловой системой на клиентской стороне сформирует модель с информацией о текущем состоянии клиентской файловой системы, затем с помощью блока работы с веб сервисом передаст его на серверную сторону. Далее делается анализ пришедших данных, создается лист изменений запросы на сервер, он взаимодействует с блоком пользовательского интерфейса.

На серверной стороне иерархия папок отличается от структуры на клиентской стороне. Здесь файлы хранятся в нескольких системных папках, в место их имени используется их идентификатор, что позволяет избежать конфликта в файловой системе.

*Интерфейс* *пользователя*, он же пользовательский интерфейс – разновидность интерфейсов, в котором одна сторона представлена человеком (пользователем), другая – машиной/устройством. Представляет собой совокупность средств и методов, при помощи которых пользователь взаимодействует с различными, чаще всего сложными, машинами, устройствами и аппаратурой, в данном случае с программой.

Интерфейс в случае с программой представляет собой то, как выглядит программа на экране и то, как пользователь взаимодействует с логикой программы. Блок интерфейса пользователя отвечает именно за отображение программы на экране и обеспечение визуальных реакций на действия пользователя. В данном случае за графическое отображение информации выступает WPF приложение. А за обработку пользовательских действий отвечает код С#, который подготавливает данные и передает и в блок работы с веб-сервисом. Например, после нажатия кнопки зарегистрироваться введенные данные на форме регистрации группируются в объект, после чего он передается блоку работы с веб-сервисом.