

Финальный отчет по ППС

выполнил студент ПК-12-2 Бекленищев Владислав

О проекте

В курсе “Проектирование программных систем” я работал с первым вариантом, который был связан с темой “Магазин видеопроката”. Задание лабораторной работы было таким:

“Видеомагазин нуждается в компьютерной системе. Ассортимент магазина составляет около тысячи видеокассет и пятьсот видео-дисков. Запас уже заказан у одного поставщика, однако, для будущих заказов директор намерен прибегать к услугам большего числа поставщиков. Все видеокассеты и диски снабжены штрих-кодом, так что сканер, интегрированный в систему, может поддерживать операции выдачи напрокат и возврата видеофильмов. Членские карточки клиентов также снабжены штрих-кодом.

Клиенты имеют возможность резервировать видео таким образом, чтобы комплект видеофильмов был собран к определенной дате. Система должна обладать поисковым механизмом для ответов на запросы клиентов, включая вопросы, касающиеся фильмов, которых нет в ассортименте магазина (но которые он может заказать по просьбе клиента).

Для каждого фильма установлен конкретный период проката (исчисляемый в днях) с соответствующей платой за прокат за этот период.

Видеомагазин должен быть в состоянии немедленно дать ответ на любой запрос по наличию фильмов в запасе, а также количеству кассет или дисков (текущие условия по каждой ленте и диску должны быть известны и зафиксированы).

Плата за прокат отличается в зависимости от видеоносителя: кассета или диск. Хотя магазин держит в запасе видеодиски только одного формата – DVD, пользователи желали бы расширить в будущем систему проката и на другие форматы дисков.

Работники видеомагазина стремятся запомнить коды наиболее популярных лент. Зачастую при идентификации фильма они используют именно код фильма, а не его название (поскольку фильм с одним названием мог выпускаться разными режиссерами).

Дополнительные требования:

За кассеты и диски, возвращенные позже срока, взимается дополнительная плата за период, превышающий срок проката. Каждый видеоноситель обладает уникальным идентификационным номером.

Фильмы заказываются у поставщика, который, в общем случае, может поставить кассеты и диски в течение одной недели. Обычно один заказ делается на несколько фильмов.

Забронировать можно те фильмы, которые заказаны у поставщика и/или все копии которых находятся в прокате. Можно также забронировать те фильмы, которых нет в запасе и которые не заказаны у поставщика; при этом с клиента требуется задаток за один период проката.

Клиент может также сделать несколько предварительных заказов, однако для каждого забронированного фильма готовится отдельный запрос на бронирование. Бронирование может быть отменено. Из-за отсутствия реакции со стороны клиента, более точно, в течение одной недели с момента, когда клиенту было сообщено о возможности взять фильм напрокат. Если за фильм был уплачен задаток, он записывается на счет клиента.

База данных хранит обычную информацию о поставщиках и клиентах, т.е. адреса, телефонные номера и т.д. В каждом заказе поставщику указываются заказываемые фильмы, их количество, форматы кассеты/диска, а также дата ожидаемой доставки, отпускная цена, возможные скидки и т. д. Когда кассета возвращается клиентом или поступает от поставщика, вначале удовлетворяются предварительные заказы. Работники магазина устанавливают контакт с клиентами, сделавшими предварительный заказ. Для правильной обработки бронирования фильмов информация, связанная с бронированием, обновляется дважды: после установления контакта с клиентом, когда ему сообщается, что «забронированный фильм пришел», и после сдачи фильма клиенту напрокат. Эти шаги гарантируют правильное проведение операции бронирования.

Клиент может взять несколько кассет или дисков, однако каждому взятому видеона носителю ставится в соответствие отдельная запись. Для каждого выдаваемого напрокат фильма фиксируются дата и время выдачи, установленный и фактический срок возврата. Позже запись о просмотре обновляется, чтобы отразить факт возврата видеофильма и факт окончательного платежа (или возврата денег). Кроме того, запись хранит информацию о продавце, отвечающем за прокат фильма. Детальная информация о клиенте и по прокату хранится в течение года, чтобы можно было легко определить уровень доверия к клиенту. Старая информация по прокату сохраняется в течение года в целях проведения аудита.

Все операции выполняются с использованием наличности, электронного перевода денег или кредитных карточек. От клиентов требуется внести плату за прокат при выдаче кассет/дисков. Если кассета/диск возвращены позже установленного срока (или не могут быть возвращены по каким-либо причинам), плата снимается либо со счета клиента, либо принимается непосредственно от клиента.

Если кассета/диск задержаны более чем на два дня, клиенту отправляется уведомление о задержке. После отправки двух уведомлений о задержке одной и той же кассеты/диска, клиент предупреждается о том, что он является «нарушителем» и при следующем его обращении в магазин руководство рассматривает вопрос о снятии с него статуса «нарушителя».

Собственно, нужно было создать 6 диаграмм для этой информационной системы. Обо всех диаграммах мы поговорим в этом отчете. Кроме того, структуру отчета будет состоять из пунктов (фаз) методологии RAD. Мы выясним на каком этапе лучше проектировать ту или иную диаграмму, сколько людей в команде разработчиков должно быть и на каком оборудовании должна развёртываться вся система.

RAD (rapid application development) я выбрал потому, что данная методология предусматривает небольшую команду разработчиков, короткий, но проработанный

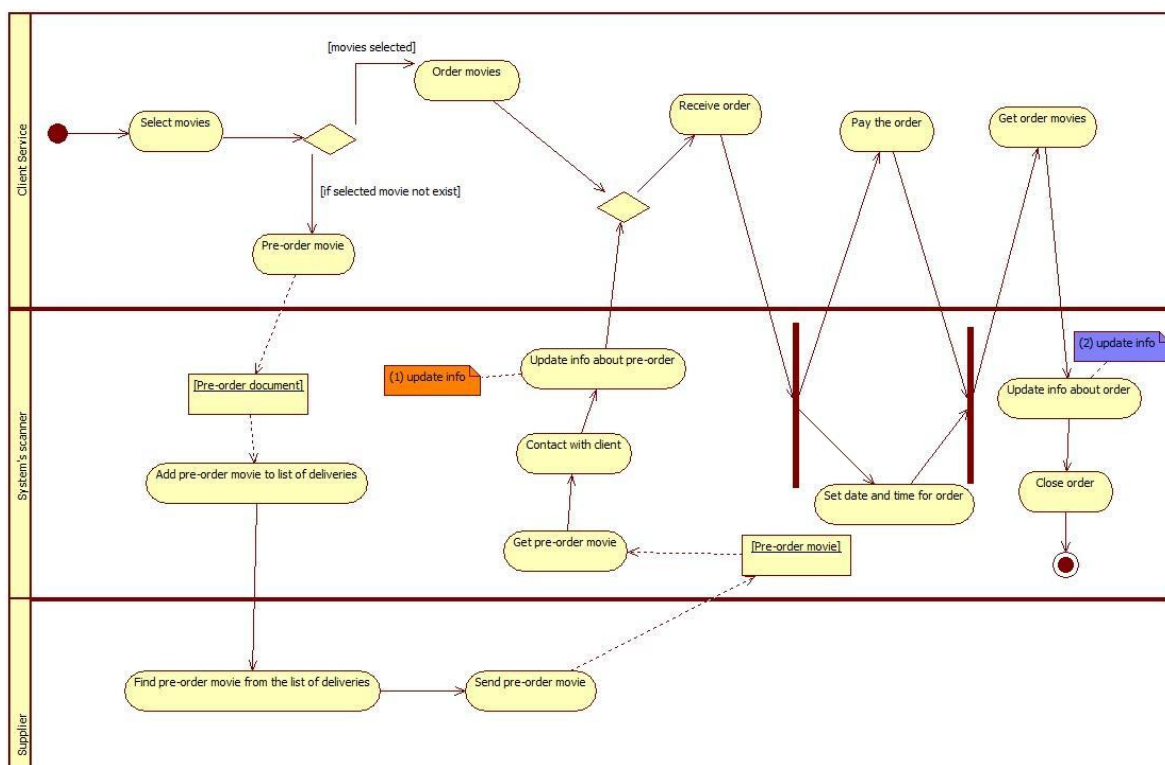
график, и повторяющийся цикл разработки, что позволит постоянно общаться с заказчиком, так как первоначальные требования поставлены очень нечетко.

Фаза анализа и планирования

На этом этапе проводится анализ требований, составление технического задания и планирование графика. Также на этом этапе подбираются оптимальные решения проблем, определяются наиболее актуальные бизнес-процессы для автоматизации, определяются сроки и стоимость.

В нашем случае я бы не сказал, что постановка задачи лабораторной работы является техническим заданием. ТЗ - документ, в котором более лаконично отражены требования заказчика благодаря работе бизнес-аналитика. Причем скорей всего, в документе далеко не все требования отражены (особенно если мы имеем дело с RAD, спиральной или похожими моделями SDLC).

Как я уже сказал, на этой фазе также подбирается команда разработчиков. По диаграмме деятельности и по диаграмме состояний мы определяем количество разработчиков:



Как видим, по диаграмме деятельности у нас есть три дорожки: сторона клиента, сторона системного сканера и сторона поставщика. В итоге, для создания такого проекта может понадобится 3 разработчика. То есть, количество разработчиков соответствует количеству дорожек на диаграмме деятельности.

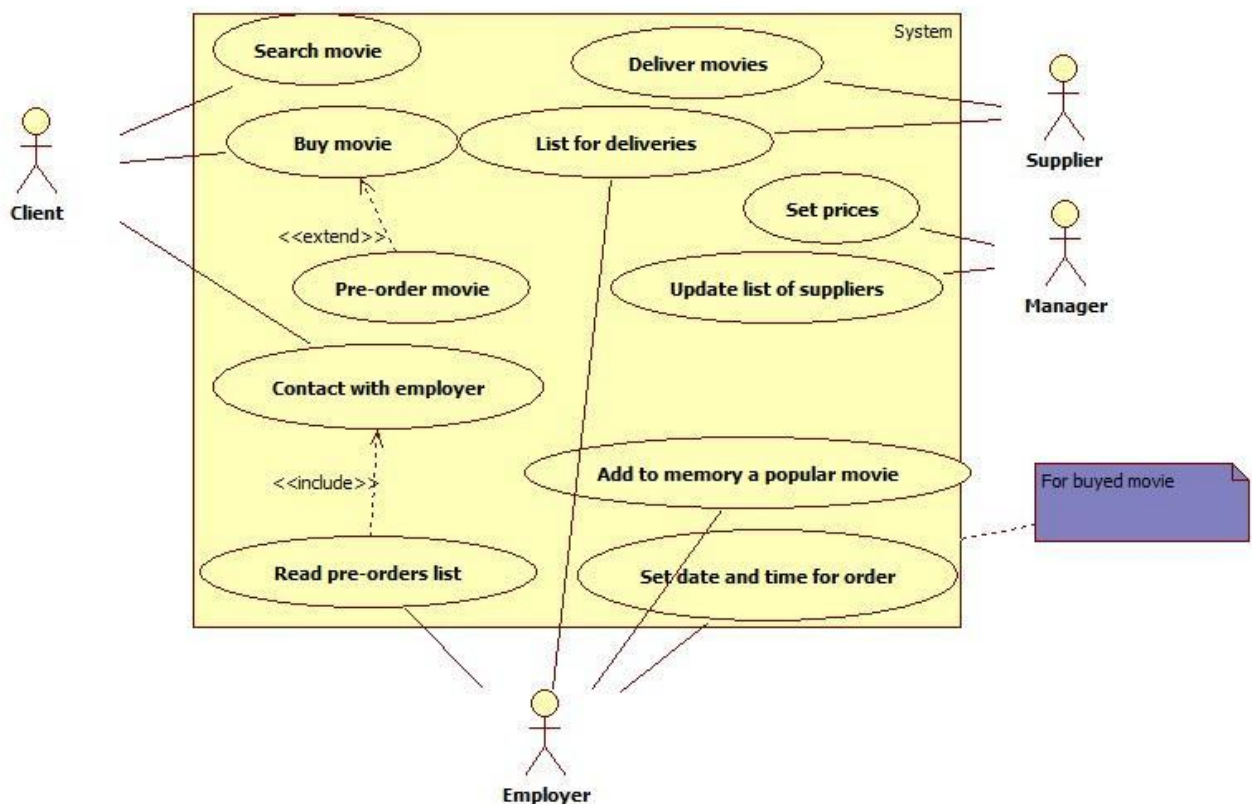
Фаза проектирования

На основе предыдущего этапа проводится **проектирование системы**. Эта методология проектирования соединяет в себе объектную декомпозицию, приемы представления физической, логической, а также динамической и статической моделей системы.

Во время проектирования разрабатываются проектные решения по выбору платформы, где будет функционировать система языка или языков реализации, назначаются требования к пользовательскому интерфейсу, определяется наиболее подходящая СУБД. Разрабатывается функциональная спецификация ПО: выбирается архитектура системы, оговариваются требования к аппаратному обеспечению, определяется набор орг. мероприятий, которые необходимы для внедрения ПО, а также перечень документов, регламентирующих его использование.

Проектирование выполняется с помощью UML. Для создания UML диаграмм использую я воспользовался таким case-средством как StarUML.

Учитывая сложность задания, в первую очередь я попытался отсортировать предложенную информацию относительно участвующих лиц в системе. То есть, я определил кто будет пользоваться данной информационной системой. Для документирования данной информации я составил диаграмму прецедентов (use case diagram). Именно эта диаграмма позволяет визуальное представить взаимодействие пользователей и системы:



Самой важной частью диаграммы являются сами прецеденты. Именно они подробно описываются в документе "Use cases" проекта, который является набором алгоритмов

(сценарием), в которых исполнителями считаются актёры (или роли), а действие происходит над системой. Самим созданием use cases может заниматься бизнес-аналитик, так как именно он общается с заказчиком. Однако в маленьких проектах его роль может выполнять другой сотрудник, например тим лидер. Я специально не включал бизнес-аналитика в список, так как мы работаем именно с небольшим проектом.

Далее, мы занимаемся проектированием самой архитектуры системы. Таким в больших проектах может заниматься архитектор ПО, но в нашем случае этим может заняться сам программист. Таким образом, мы строим следующие диаграммы:

- Диаграмму состояний (state diagram);
- Диаграмму деятельности (activity diagram);
- Диаграмму классов (class diagram).

Сразу стоит отметить, что диаграмма состояний и диаграмма деятельности далеко не являются обязательными как и многие другие диаграммы. Они просто помогают графически представить разные бизнес-процессы, состояния, в которых может быть система.

Диаграмма состояний позволила мне определить в каких состояниях может находиться система, определить динамическое поведение сущностей, что помогло в дальнейшем, при построении диаграммы классов:

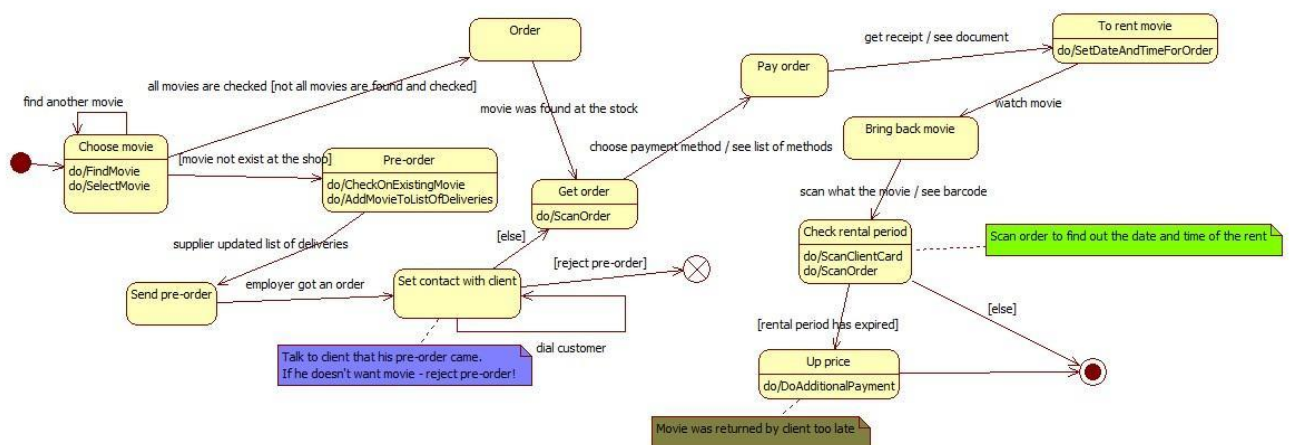
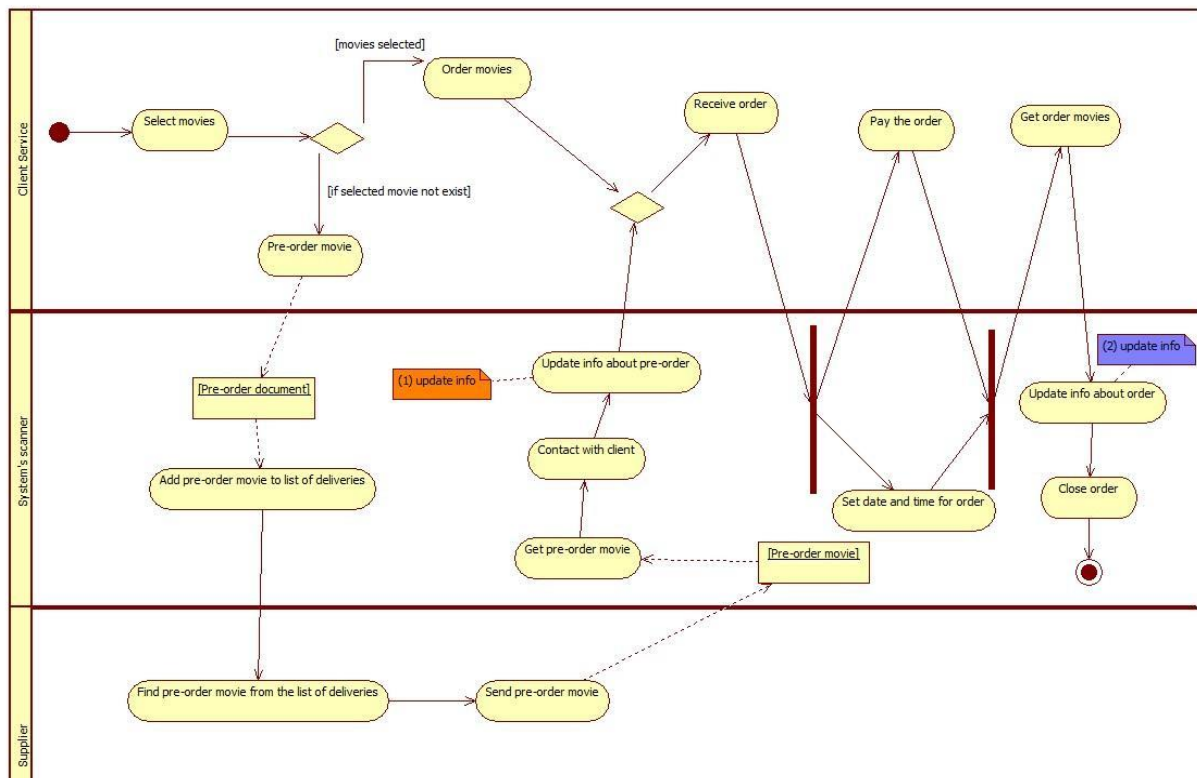
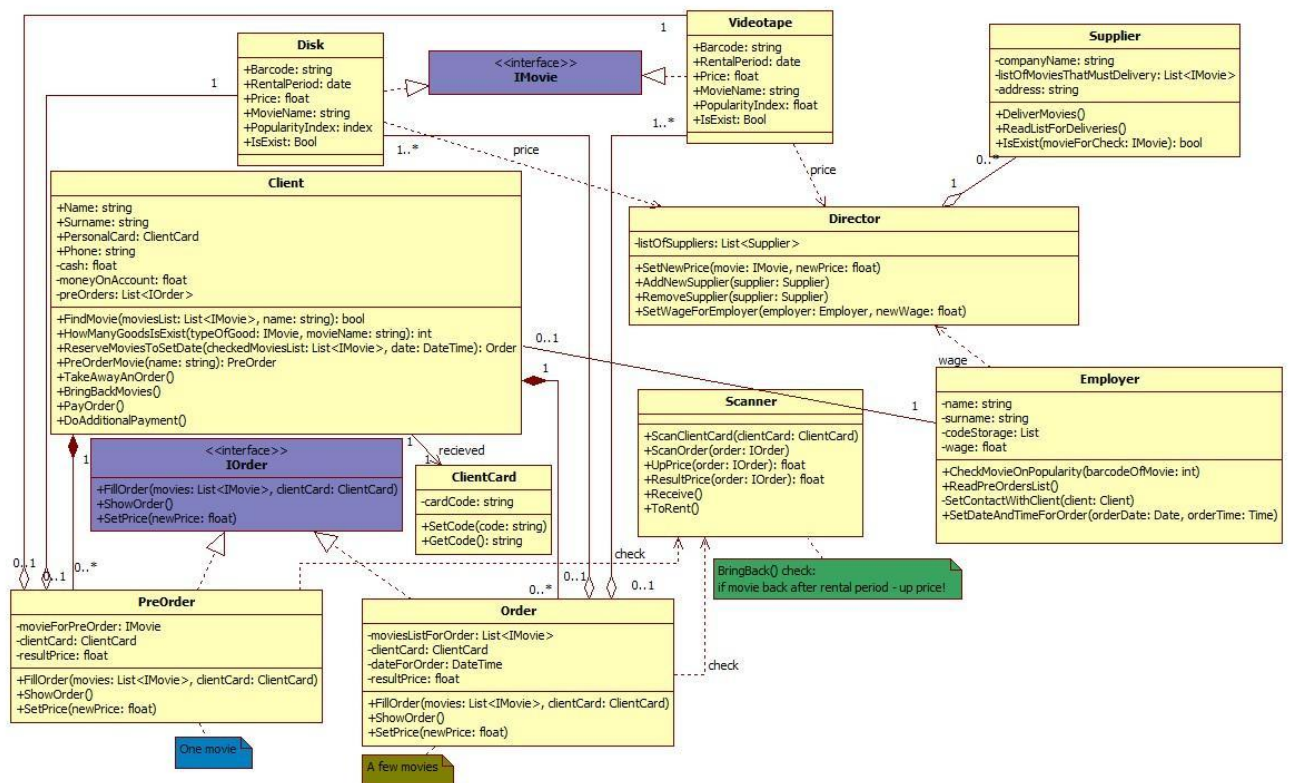


Диаграмма деятельности помогла понять какие бизнес-процессы протекают в системе и каким образом сущности взаимодействуют между собой в системе, например при составлении и оплате заказа (см. схему). На самом деле, диаграмма деятельности используется даже реже чем диаграмма состояний в виду того, что большинство информации о сущностях подаёт именно последняя в то время как первая подаёт именно взаимодействие этих сущностей.



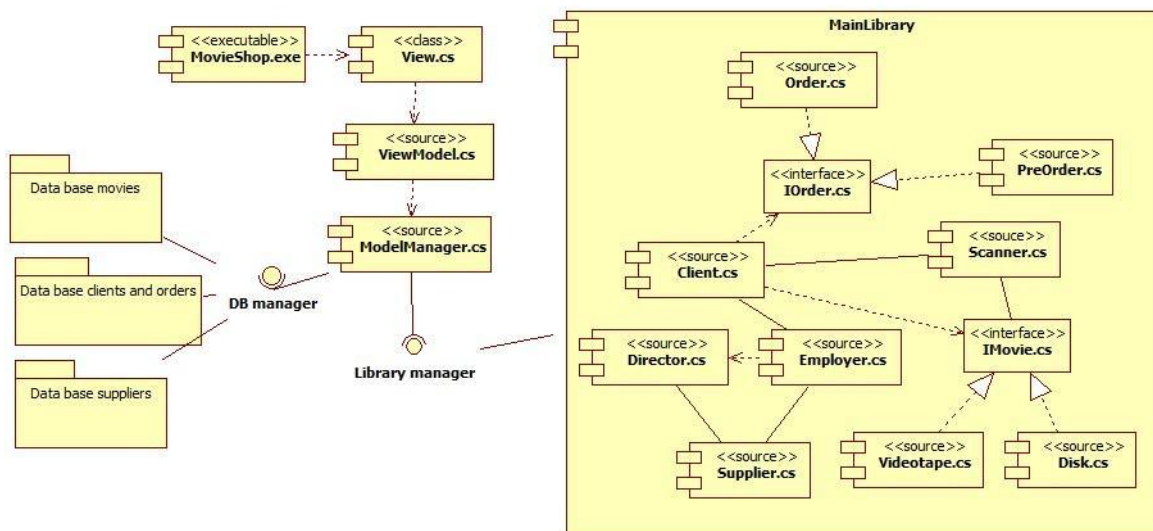
Диаграмму классов я начал создавать еще до составления диаграммы состояний, а потом, уже при составлении диаграммы состояний и деятельности, я обновлял и дополнял диаграмму. Получилась следующая схема:



Как видим, на диаграмме отражены все сущности и взаимодействия между этими сущностями в терминах объектно-ориентированного проектирования на языке UML.

Так как на этой фазе еще и определяется и то на какой платформе будет работать система, на каком оборудовании, я не мог обойти диаграмму компонентов и диаграмму развертывания. Обычно эти диаграммы объединяют в одну, но в целях обучения мы их отдельно строили.

Диаграмма компонентов, в отличие от ранее рассмотренных диаграмм, описывает особенности физического представления системы. Диаграмма компонентов позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами, в роли которых может выступать исходный, бинарный и исполняемый код. Во многих средах разработки модуль или компонент соответствует файлу. Пунктирные стрелки, соединяющие модули, показывают отношения взаимозависимости, аналогичные тем, которые имеют место при компиляции исходных текстов программ. Основными графическими элементами диаграммы компонентов являются компоненты, интерфейсы и зависимости между ними.



Как видим, из диаграммы компоненты представлены в виде файлов, которые поддерживает только операционная система Windows (файл .exe). А файлы .cs говорят о разработке приложения на языке C#.

Диаграмма развертывания предназначена для визуализации элементов и компонентов программы, существующих лишь на этапе ее исполнения (runtime). При этом представляются только компоненты-экземпляры программы, являющиеся исполнимыми файлами или динамическими библиотеками. Те компоненты, которые не используются на этапе исполнения, на диаграмме развертывания не показываются.

Диаграмма развертывания содержит графические изображения процессоров, устройств, процессов и связей между ними. В отличие от диаграмм логического представления, диаграмма развертывания является единой для системы в целом, поскольку должна всецело отражать особенности ее реализации.

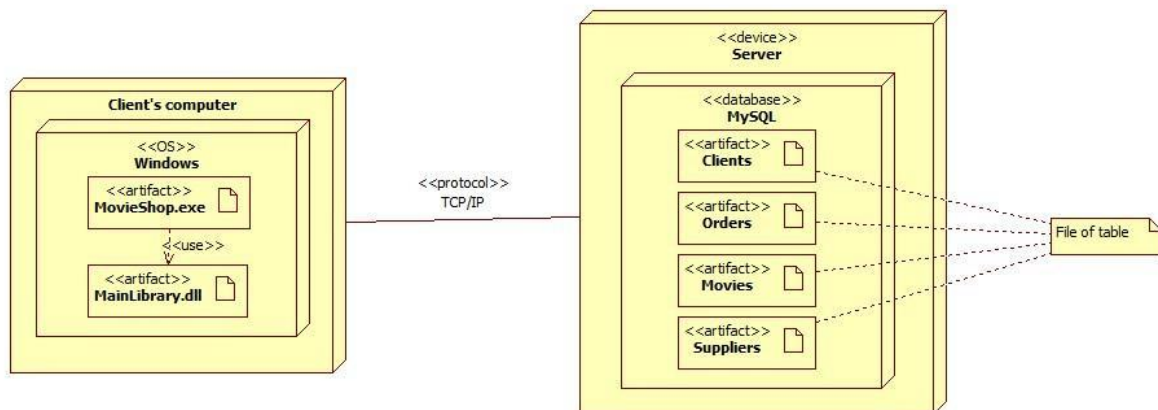


Диаграмма развёртывания показывает, что приложение будет развёрнуто на компьютерах с операционной системой Windows. При этом эти же компьютеры должны постоянно быть подключены к сети, так как приложение будет постоянно связываться с базой данных, в которой будет храниться информация о фильмах, о клиентах, о поставках и поставщиках. Сама база данных поддерживается с помощью СУБД MySQL.

MySQL позволяет работать одновременно нескольким пользователям, кроме того она очень быстро выполняет все операции. Также она является кроссплатформенной - не требует привязки к конкретной платформе, а значит наш сервер может находиться на любой операционной системе.

Фаза построения

На этом этапе, в рамках RAD методологии, выполняется быстрая разработка приложения (coding), построение реальной системы на основе полученных на предыдущем этапе моделей.

Для быстрой разработки приложения можно воспользоваться удобной фичей используемого case-средства (в моём случае StarUML) - **сгенерировать** код проекта на основе построенной диаграммы классов.

Если при написании кода удалось создать какие-либо проектные решения, которые намного лучше чем те, которые описаны

Фаза внедрения

Внедрения системы обычно предусматривает следующие шаги:

- установка системы,
- обучение пользователей,
- эксплуатация.

К любой разработке прилагается полный пакет документации, который включает в себя описание системы, руководства пользователей и алгоритмы работы.

Поддержка функционирования ПО должна осуществляться группой технической поддержки разработчика.

Резюме

В курсе «Проектирование программных систем» я создавал модель информационной системы «Магазин видеопроката» (вариант 1).

Модель проекта создавалась с помощью языка моделирования UML, с помощью case-средства StarUML. В результате получилось создать 6 диаграмм :

- диаграмму прецедентов;
- диаграмму состояний;
- диаграмму деятельности;
- диаграмму классов;
- диаграмму компонентов;
- диаграмму развертывания.

Оценивая диаграмму деятельности и диаграмму состояний, я определил, что для проекта потребуется 3 разработчика программного обеспечения.

Оценивая диаграмму развертывания , нетрудно догадаться, что для развёртывания проекта понадобятся:

- Компьютеры с операционной системой Windows;
- Сервер с MySQL базой данных;
- И сетевое соединение с компьютером и сервером с БД.