МИНИСТЕРСТВО СВЯЗИ ИНФОРМАТИЗАЦИИ И ТЕЛЕКОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ РЕСПУБЛИКИ УЗБЕКИСТАН

ТАШКЕНТСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Кафедра «Информационных технологий»

Курсовая работа

по дисциплине

ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ

на тему «Система администрирования мобильного приложения «Онлайн Кинотеатр»»

Приняла: Каримова В. А.

Выполнил: Белявский Иван

Дата защиты\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ташкент 2016

**ТАШКЕНТСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

Факультет: Компьютерный инжиниринг

Кафедра: Информационные технологии

Направление (специальность): Информатика и информационные технологии

УТВЕРЖДАЮ

Зав. кафедрой В.А. Каримова

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_2016 й.

ЗАДАНИЕ

к курсовой работе

1. По курсовой работе

2. Группа: 223-12

3. Ф.И.О. студента: Белявский Иван Владимирович

4. Ф.И.О руководителя: Каримова Венера Аркиновна

5.Тема курсовой работы: Система администрирования мобильного приложения «Онлайн Кинотеатр»

Тема утверждена на заседании кафедры протоколом №\_\_\_\_ от \_\_\_\_\_\_\_\_\_\_\_\_\_

6. Данные относительно курсовой работы:

График выполнения работы:

|  |  |  |  |
| --- | --- | --- | --- |
| № | Наименование | Начало | Конец |
| 1 | Предпроектный этап |  |  |
| 2 | Проектный этап |  |  |
| 3 | Внедрение системы |  |  |
| 4 | Анализ функционирования |  |  |

Дата:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель: Каримова В. А.

Аннотация

Цель данной курсовой работы заключается в описании процесса проектирования информационной системы администрирования мобильного приложения «Онлайн Кинотеатр». В курсовой работе приведены описания алгоритма работы информационной системы, диаграммы IDEF0, ERD-диаграммы, схемы реляционной модели, UML диаграммы прецедентов, классов и последовательности. Также приведена инструкция с интерфейсом информационной системы.

Объем работы: 34 стр.

Рисунков: 10

Таблиц: 2

Содержание

[Введение 5](#_Toc446419788)

[1. Теоретическая часть 9](#_Toc446419789)

[1.1. Анализ предметной области 9](#_Toc446419790)

[1.2 Функции и задачи администратора ИС 10](#_Toc446419791)

[1.3 Системное администрирование 11](#_Toc446419792)

[1.4 Автоматизация управления сетью 15](#_Toc446419793)

[1.4 Технические требования 20](#_Toc446419794)

[2. Методология проектирования информационной системы 21](#_Toc446419795)

[2.1 IDEF, UML, DFD 21](#_Toc446419796)

[IDEF 0 21](#_Toc446419797)

[IDEF 3 21](#_Toc446419798)

[UML 22](#_Toc446419799)

[DFD 23](#_Toc446419800)

[3. Проектирование системы администрирования для мобильного приложения «Онлайн кинотетр» 25](#_Toc446419801)

[3.1 Диаграмма IDEF0, UML 25](#_Toc446419802)

[Вывод 26](#_Toc446419803)

[3.2 Создание логической модели, ERD-диаграммы 27](#_Toc446419804)

[3.3 Диаграммы UML 29](#_Toc446419805)

[3.3 Диаграмма классов 30](#_Toc446419806)

[Заключение 32](#_Toc446419807)

[Список используемой литературы 33](#_Toc446419808)

# **Введение**

Любая большая современная информационная система (ИС) в любой предметной области, как правило, несёт другую культуру управления. Проблема внедрения таких систем – это проблема восприятия, освоения и внедрения другой культуры управления.

Для большинства предметных областей внедрения информационных технологий характерны высокие требования к качеству их работы и надёжности эксплуатации, от которых во многом зависит эффективность использования и возврата вложенных средств. Эти требования относятся к проблемам, связанным с администрированием ИС, которые представляют собой сетевой программно-аппаратный и информационный ресурс.

***Администрирование*** (административные механизмы) – это процедуры управления, регламентирующие некоторые процессы или их часть.

В нашем случае, оно входит в полномочия администратора ИС, который фиксирует и руководит соответствующими его полномочиям процессами и ситуациями, нуждающимися в целевом управлении и ограничениях.

К таковым процессам относят планирование работ, построение, эксплуатацию и поддержку эффективной ИТ-инфраструктуры, интегрированной в общую архитектуру информационной системы – один из критических факторов успешной реализации стратегических бизнес-целей организации.

***Инфраструктура***включает решения по программному обеспечению, аппаратному комплексу и организационному обеспечению ИС, что соответствует пониманию системы в современных стандартах типа ISO/IEC 15288.

В инфраструктуру ИТ входят:

* вычислительное и телекоммуникационное оборудование;
* каналы связи;
* инженерно-техническое оборудование, обеспечивающее работу вычислительных и телекоммуникационных средств;
* программные приложения, реализующие функциональность ИТ-систем (прикладное программное обеспечение) и обеспечивающие функционирование оборудования (системное программное обеспечение);
* администрация и персонал, осуществляющие эксплуатацию ИС;
* внутренние положения и инструкции, регламентирующие работу персонала с ИТ-системами;
* исходные тексты программных приложений, входящих в ИТ-систему (прикладное и системное ПО);
* системная и пользовательская документация программных приложений и аппаратных комплексов и др.

Современная ИТ-инфраструктура ИС должна обладать высокой производительностью, надёжностью, гибкостью, масштабируемостью и другими качествами.

Совокупность сетевых программно-технических средств ИС образует её архитектуру. Для ИС, ориентированных, в первую очередь, на получение экономического эффекта, стремятся выбрать архитектуру системы с минимальной совокупной стоимостью владения.

*Совокупная стоимость владения ИС* состоит из плановых затрат и стоимости рисков. Совокупная стоимость рисков определяется стоимостью бизнес-рисков, а также вероятностью появления технических, программных и организационно-эксплуатационных рисков.

Так, например, совокупная стоимость владения серверными кластерами определяется как сумма затрат на их приобретение, стоимости серверов, эксплуатационных затрат и стоимости потерь от простоев и нарушения программно-информационного их содержания. Последние в нашем понимании соответствуют стоимости рисков.

Типы ИС, с точки зрения выполняемых ими функций (функционала), решаемых задач и т.п. обычно связывают с делением организаций на представителей крупного, среднего и малого бизнеса. Данное деление достаточно условно. Основными компонентами организаций, по которым с точки зрения ИТ оно обычно осуществляется, являются доход или объём выполняемых работ за год, выраженные в денежных единицах, а также количество персонала и компьютеров.

Так, предлагается считать очень малыми предприятиями те, у которых число сотрудников меньше 20 человек (при этом называют совсем маленькие организации с числом сотрудников от 5 до 10 человек); малыми – с числом сотрудников от 20 до 100, средними – от 100 до 1000 человек. По числу компьютеров к малому бизнесу относят предприятия, у которых не более 50 ПК, от 50 до 100 компьютеров – к среднему, а более 500 – к крупному бизнесу.

Выбор ИС представителями малого и среднего бизнеса зачастую обусловлен их отношением к стоимости системы и срокам её внедрения. Эффективность внедрения и использования ИС во многом зависит от развития и комфортности окружающей среды (инфраструктуры).

Для средних предприятий важным критерием выбора ИС является возможность решать все важные задачи с её помощью, поскольку использование нескольких приложений вызывает потребность их интеграции, что, как правило, влечёт за собой негативные последствия. ИС от нескольких поставщиков с различным ПО сложно эксплуатировать и поддерживать. При этом для малых предприятий из-за небольших масштабов деятельности и меньшего числа бизнес-задач этот критерий не столь важен.

Чем больше организация, тем более формализованные процессы она использует и тем более высоки её требования к производительности и охвату бизнес-процессов. Малые предприятия не имеют собственных ИТ-служб, средние, как правило, имеют небольшую ИТ-службу с ограниченными функциями, например, связанными с эксплуатацией ИС.

Внедряя простые готовые решения представители СМБ могут обходиться стандартными услугами ИТ-организаций поставщиков и обслуживателей программно-технических комплексов и систем по технической поддержке и сопровождению ИС, в том числе аутсорсингом.

Имеется базовый функционал, который подходит большинству организаций. Однако, даже, если предприятия среднего бизнеса используют одни и те же программы, что и предприятия крупного бизнеса, стоимость их внедрения могут значительно отличаться. Как отмечают специалисты, основные отличия между этими типами предприятий зависят от степени их автоматизации, принятой ИТ-стратегии, степени зрелости бизнеса и др.

Кроме того, считается, что для крупных предприятий нет стандартных решений. При этом в 99% случаев используются дополнительные настройки ПО. Многие ИС для СМБ могут представлять собой усечённые версии полномасштабных ИС. Расширение функциональности ИС и включение в неё новых модулей наиболее эффективно решается в масштабируемых системах с модульной архитектурой.

# **1. Теоретическая часть**

# **1.1. Анализ предметной области**

*Администрирование ИС* заключается в предоставлении пользователям соответствующих прав использования возможностей работы с системой (базой, банком данных); обеспечении целостности данных, а также создании многопользовательских приложений.

*Администрирование ИС* – это её инсталляция (установка), управление доступом к ней, обеспечение целостности ИС и др.

*Формирование (инсталляция) ИС* на компьютере заключается в подготовке одного или нескольких файлов данных, которые будут установлены и использоваться в виде единой ИС. ИС создаётся один раз, независимо от того, сколько файлов данных она имеет, и сколько обращений к ней будет производиться.

Процедуру создания ИС можно использовать, если стереть информацию в существующей ИС. При этом будет создана ИС с тем же именем и той же физической структурой.

Создание ИС включает следующие операции:

* создание новых файлов данных, или стирание данных, хранившихся в предыдущих файлах данных;
* создание структур, требующихся для доступа и работы с ИС;
* инициализацию управляющих файлов и журнала для ИС.

Для обеспечения защиты ИС, используют различные ограничения целостности данных, журнализацию, репликацию и резервное копирование.

Создаваемый журнал обычно содержит подробную информацию о загрузке, включая:

* имена входящих в базу данных файлов;
* входные данные и связанные с ними определения таблиц;
* ошибки и результаты работы БД;
* итоговую статистику.

Прежде чем создавать ИС необходимо:

* спланировать таблицы и индексы, а также оценить необходимое им пространство;
* спланировать защиту ИС, включая конфигурацию её журналов с учётом занимаемого ими пространства и стратегию резервного копирования.

*Администрирование информационных систем* включает следующие цели:

1. Установка и настройка сети.
2. Поддержка её дальнейшей работоспособности.
3. Установка базового программного обеспечения.
4. Мониторинг сети.

Выделяют три основные категории пользователей ИС: разработчики, администраторы и собственно пользователи.

**1.2 Функции и задачи администратора ИС**   
Администрирование ИС осуществляется лицом, управляющим этой системой. Такое лицо называется *администратором*.

Если ИС большая, эти обязанности могут выполнять несколько человек (группа администраторов).

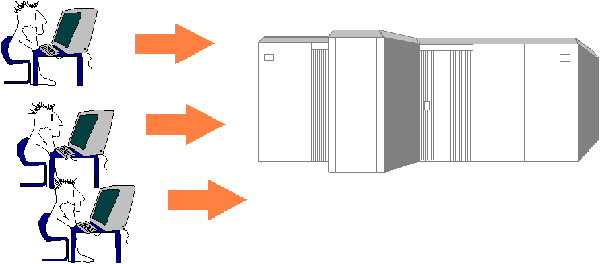
Администратор ИС осуществляет её запуск и останов. Он может использовать табличные пространства для:

* управления распределением памяти для объектов ИС;
* установления квот памяти для пользователей ИС;
* управления доступностью данных, включая режимы (состояния) online или offline;
* копирования и восстановления данных;
* распределения данных по устройствам для повышения производительности.

В процессе своей деятельности администратор ИС взаимодействует с другими категориями её пользователей, а также и с внешними специалистами, не являющимися пользователями ИС (рис.1).

**1.3 Системное администрирование**  *Основной целью системного администрирования* является приведение сети в соответствие с целями и задачами, для которых эта сеть предназначена. Достигается эта цель путём управления сетью, позволяющего минимизировать затраты времени и ресурсов, направляемых на управление системой, и в тоже время максимизировать доступность, производительность и продуктивность системы.

*История системного администрирования* насчитывает несколько десятилетий. Задачи, связанные с управлением вычислительными комплексами (системами), возникли сразу с появлением этих комплексов, то есть первых локальных вычислительных сетей, представлявших некоторый главный (хост) компьютер, к которому подключались терминальные устройства пользователей (Рис. 1).



*Рис.1. Этап 1: модель “хост-компьютер + терминалы”.*

До конца 1980-х годов доминировала централизованная вычислительная модель типа “мэйнфрейм–терминалы”. Она отражала архитектуру средств системного администрирования, то есть администрирования системой. Такое решение означало существование единого образа вычислительной среды. Специалисты отмечают, что по удобству и надёжности эти системы могут служить своеобразным эталоном для оценки всех остальных средств администрирования.

Первые компьютерные информационные системы на базе мэйнфреймов (например, IBM-360/370, или отечественных аналогов – ЕС ЭВМ), либо на базе так называемых мини-ЭВМ (например, PDP-11 и отечественные СМ-4) использовали модель “хост-компьютер + терминалы”. Характерной её особенностью была полная “неинтеллектуальность” терминалов, используемых в качестве рабочих мест – их работой управлял хост-компьютер (Рис.1).

Этот подход обладал некоторыми достоинствами: пользователи могли совместно использовать различные ресурсы хост-компьютера (оперативную память, процессор) и довольно дорогие для того времени периферийные устройства (принтеры, графопостроители, устройства ввода с магнитных лент и гибких дисков, перфосчитыватели, дисковые накопители и иное оборудование, ныне экзотическое). Используемые при этом операционные системы поддерживали *многопользовательский многозадачный режим*, а централизация ресурсов и оборудования облегчала и удешевляла эксплуатацию такой системы.

Недостатки подобной архитектуры заключались, главным образом, в полной зависимости пользователя от администратора и условий работы хост-компьютера. Пользователь (а порой и программист) не имел возможности настроить рабочую среду под свои потребности – используемое программное обеспечение, в том числе и текстовые редакторы, компиляторы, СУБД, целиком и полностью коллективным.

В значительной степени этот недостаток данных систем привёл к появлению и развитию индустрии персональных компьютеров, работающих с ИС. Наряду с дешевизной и простотой эксплуатации, привлекательной особенностью настольных информационных систем стала так называемая “персонализация рабочей среды”, то есть возможность пользователя выбирать для работы наиболее соответствующие его потребностям и необходимые ему программные инструменты (текстовый редактор, СУБД, электронную таблицу и др.). Естественно, что на рынке программных продуктов стало появляться много таких инструментов. Например, электронные таблицы и настольные СУБД (dBase, FoxBase, Сlipper, Paradox и др.), нередко совмещающие в себе собственно СУБД и средства разработки приложений, использующие базы данных.

В используемых подобных инструментариях средства администрирования предназначались для контроля за функционированием отдельных компонентов (сетевым оборудованием, персональными компьютерами и рабочими станциями, запоминающими устройствами, периферией и другими устройствами). Обычно этот контроль заключался в сборе данных о ресурсах вместо подлинного управления их работой. Такой тип управления нельзя отнести к сетевому администрированию в строгом смысле этого слова.

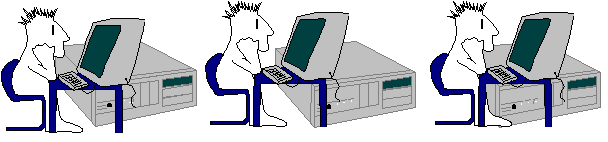
Функциональные области управления, относящиеся к *системному администрированию*, определены в спецификациях ISO, и ориентированы на:

* *решение проблемных ситуаций* (диагностика, локализация и устранение неисправностей, регистрация ошибок, тестирование);
* *управление ресурсами* (учёт, контроль использования ресурсов, выставление счетов за использованные ресурсы и ограничение доступа к ним);
* *управление конфигурацией*, направленное на обеспечение надёжного и эффективного функционирования всех компонентов информационной системы;
* контроль производительности (сбор и анализ информации о работе отдельных ресурсов, прогнозирование степени удовлетворения потребностей пользователей/приложений, меры по увеличению производительности);
* защита данных (управление доступом пользователей к ресурсам, обеспечение целостности данных и управление их шифрованием).

Основной результат длительного развития отрасли системного администрирования выразился в том, что с функциональной точки зрения основные платформы управления довольно похожи друг на друга. Различия между ними кроются в сфере структурного исполнения и, кроме того, связаны с исходными целями, которые ставились на начальных этапах их разработки.

В архитектурной реализации наблюдаются варианты единых интегрированных систем (CA-Unicenter) и продукты, изначально имевшие модульную структуру (HP OpenView). Архитектурные аспекты становятся значимыми, когда возникает потребность в интеграции различных продуктов, например, точечных решений с базовой платформой администрирования.

Фирмы среднего размера заинтересованы не в точечных продуктах, а в законченных решениях, имеющих ограниченную функциональность, интуитивный Web-интерфейс и приемлемую цену.



*Рис.2. Этап 2: Автономная персональная обработка данных*

Следующим этапом развития архитектуры информационных систем было появление сетевых версий СУБД, позволяющих осуществлять *многопользовательскую работу с общими данными в локальной сети*. Этот подход сочетал в себе удобства персонализации пользовательской среды и простоты эксплуатации, а также преимущества, связанные со вновь открывшимися возможностями совместного использования периферии (главным образом сетевых принтеров и сетевых дисков, в том числе хранящих коллективные данные). Большинство реально эксплуатируемых в нашей стране информационных систем имеют такую архитектуру (в том числе некоторые информационные системы, реализованные на Delphi и C++Builder). Этот подход оправдывает себя в случае не очень большого количества пользователей системы, не слишком большого объёма данных и количества таблиц, невысоких требований к защите данных.

# **1.4 Автоматизация управления сетью**

Автоматизированная информационная система (АИС) является частью любого административного механизма – платформой управления и сетевой службой.

***Платформа управления сетью*** (Network management platform) – комплекс программ, предназначенных для управления сетью и входящими в неё системами.

Сетевая служба использует сервис, предоставляемый областью взаимодействия, и обеспечивает связь прикладных процессов, расположенных в различных абонентских системах сети.

**Сетевая** **служба** – это прикладная программа:

● взаимодействующая с клиентами, серверами и данными в сети;

● управляющая процедурами распределённой обработки данных;

● информирующая пользователей о происходящих в сети изменениях.

***Сетевая служба NMS***(Network management service) – служба, выполняющая процессы управления сетью. Она разработана ISO и располагается на прикладном уровне. Сетевая служба NMS определяет:

***●***функции управления;

***●*** виды сервиса, предоставляемые для управления;

***●*** структуру управляющей информации;

***●*** протоколы, определяющие транспортировку управляющей информации.

Система административных регламентов и информационная система являются затратной частью системы управления, но их отсутствие не гарантируют качество, оперативность и эффективность управления.

Следует иметь в виду, что технологии и инструменты являются более стабильной компонентой, чем системы управления. Изменения стратегии, политики, методики, исполнителей обычно приводят к изменению как системы управления, так и информационной системы. При этом инструмент, например существующая АИС, может обеспечить решение новых задач, порой с минимальной дополнительной её настройкой. Смена инструмента обычно приводит к изменению работы системы управления. Поэтому при целостном, целенаправленном формировании и функционировании системы управления, следует осуществлять одновременное развитие автоматизированной информационной системы и административных механизмов управления ею.

Очевидно, что управление сетью, как правило, целесообразно осуществлять с одного рабочего места.

Потребность в контроле за сетью с одной управляющей станции способствовала появлению различных архитектур платформ и приложений администрирования. Наибольшее распространение среди них получила*двухуровневая распределённая архитектура* “менеджер–агенты”. Программа-менеджер функционирует на управляющей консоли, постоянно взаимодействуя с модулями-агентами, которые запускаются в отдельных устройствах сети. На агенты в такой схеме возлагаются функции сбора локальных данных о параметрах работы контролируемого ресурса, внесение изменений в его конфигурацию по запросу от менеджера, предоставление последнему административной информации.

В качестве частичного решения проблемы исчерпания пропускной способности предлагается трёхуровневая схема, в которой часть управляющих функций делегируется важнейшим сетевым узлам. Инсталлированные в этих узлах программы-менеджеры через собственную сеть агентов управляют работой “подотчётных” им устройств и одновременно сами выступают в роли агентов по отношению к основной программе-менеджеру (менеджеру менеджеров), запущенной на управляющей станции. В результате основная часть служебного трафика оказывается локализованной в отдельных сетевых сегментах, поскольку “общение” локальных менеджеров с административной консолью осуществляется только тогда, когда в этом действительно возникает необходимость.

Несмотря на очевидные удобства двухуровневой схемы, её применение в реальной сетевой среде приводит к возрастанию объёмов служебного трафика и, как следствие, снижению эффективной пропускной способности, доступной приложениям. Этот эффект особенно ощущается в сложных сегментированных сетях, содержащих большое количество активных устройств. В качестве частичного решения проблемы исчерпания пропускной способности предложена трёхуровневая схема, в которой часть управляющих функций делегировалась важнейшим сетевым узлам.

Инсталлированные в этих узлах программы-менеджеры через собственную сеть агентов управляют работой “подотчётных” им устройств и в то же время сами выступают в роли агентов по отношению к основной программе-менеджеру (менеджеру менеджеров), запущенной на управляющей станции. В результате основная часть служебного трафика оказывается локализованной в отдельных сетевых сегментах, поскольку “общение” локальных менеджеров с административной консолью осуществляется только тогда, когда в этом действительно возникает необходимость. Трехуровневая архитектура управления впервые была реализована в 1995 году в продукте Solstice компании Sun Microsystems.

Необходимость контролировать работу разнообразного оборудования в гетерогенной среде потребовала унификации основных управляющих процедур. Схема “менеджер–агенты” нашла выражение в протоколе Simple Network Management Protocol (SNMP), быстро ставшим базовым протоколом сетевого администрирования. Для удобства и облегчения работы сетевых администраторов используются системы управления, основанные на SNMP из группы протоколов TCP/IP. Для сетей SNA (IBM's System Network Architecture) другим возможным решением может являться STM – Station Management.

Одна из современных идей совершенствования технологий администрирования сетью заключается в сведении к минимуму роли человека в процессе администрирования ИС. Она подразумевает создание программного обеспечения, удовлетворяющего всем целям нужной системы и совмещающего в себе все функции программного обеспечения необходимого для администрирования ИС, например, совмещение контроля защиты, управления пользователями, маршрутизации, резервного копирования информации в случае сбоев и т.д. Администрирование сети в этом случае осуществляет программа, настройку которой производит администратор сети.

Данный подход позволяет экономить время, расходуемое администраторами сети, в том числе на мониторинг и управление сетью и избавиться от фактора ошибки человека. Наиболее сложно в этом случае создать алгоритм всего процесса администрирования, исключая настройку сети и настройку самого ПО. Такое решение способствует значительному облегчению процесса администрирования, поскольку настройка одной программы намного легче, чем настройка всей сети и всех приложений, связанных с работой в сети.

Другое предложение базируется на использовании беспроводных сетей с высокой скоростью передачи информации, например, на основе информации, передаваемой светом, что позволяет избежать проблем связанных с самим физическим строением сети. Например, Wireless LAN (беспроводные локальные сети) весьма удобны, но имеют недостаточную скорость передачи данных. В беспроводных сетях кроме радиоволн, для передачи информации можно использовать и другие явления, например, свет. Это позволило бы значительно увеличить скорость передачи информации, и избежать многих проблем имеющихся у проводных сетей.

Еще одна идея заключается в создании для администрирования информационных систем интеллектуального компьютера – нейрокомпьютера. Такое решение позволяет свести на нет роль человека в администрировании информационных систем, в то же время добиться максимального быстродействия сетей, и полного их соответствия заданным целям. Существуют первые компьютеры, проводящие вычисления с помощью молекул ДНК. Они предназначены для решения сложной задачи, и способны самообучаться, совершая какие либо ошибки. Аналогичные машины созданные для поддержания производительности сети, способны достигнуть достаточно большого прироста производительности.

# **1.4 Технические требования**

1. Операционная система MS Windows 7/ MS Windows 8,8.1/ MS Windows 10
2. .NET Framework 4.5.
3. Identity Framework.
4. Entity Framework.
5. ASP.NET MVC 5.
6. СУБД MySQL.
7. Среда разработки Microsoft Visual Studio.
8. II Server.
9. Объем ОЗУ 4 ГБ
10. Объем HDD 50 ГБ

# **Методология проектирования информационной системы**

# **2.1 IDEF, UML, DFD**

# **IDEF 0**

IDEF0 – функциональное моделирование (Function Modeling Method)

Реализует методику функционального моделирования сложных систем. Наиболее известной реализацией является методология SADT, предложенная в 1973 г. Д. Россом и впоследствии ставшая основной этого стандарта. Эта методика рекомендуется для начальных стадий проектирования сложных искусственных систем управления, производства, бизнеса, включающих людей, оборудование, программное обеспечение.

Основу методологии IDEF0 составляет графический язык описания бизнес-процессов, отображающих отношение «сущность-связь». Модель IDEF0 представляет собой совокупность иерархически упорядоченных и взаимосвязанных диаграмм. Каждая диаграмма является единицей описания системы и располагается на отдельном листе.

Модель может содержать четыре типа диаграмм:

* контекстную;
* декомпозиции;
* дерева узлов;
* только для экспозиции (FEO).

Контекстная диаграмма является вершиной древовидной структуры диаграмм и представляет собой самое общее описание системы и ее взаимодействия с внешней средой. После описания системы в целом проводится разбиение ее на крупные фрагменты. Этот процесс называется функциональной декомпозицией, а диаграммы, которые описывают каждый фрагмент и взаимодействие фрагментов, называются диаграммами декомпозиции.

# **IDEF 3**

Для описания логики взаимодействия информационных потоков более подходит IDEF3, называемая также workflow diagramming – методологией моделирования, предназначенная для обеспечения структурированного подхода к описанию бизнес-процесса как упорядоченной последовательности событий одновременно с описанием любых участвующих в бизнес-процессе объектов и относящихся к ним правил.

IDEF3 – стандарт моделирования бизнес-процессов, поддерживающий графическое описание непосредственного механизма функционирования системы или организации. IDEF3 содержит правила разработки двух видов сетевых диаграмм:

диаграмм потоков для бизнес-процессов;

диаграмм изменения состояния объекта.

Диаграммы workflow могут быть использованы в моделировании бизнес – процессов для анализа завершенности процедур обработки информации. С их помощью можно описывать сценарии действий сотрудников организации, например последовательность обработки заказа или события, которые необходимо обработать за конечное время. Каждый сценарий сопровождается описанием процесса и может быть использован для документирования каждой функции.

IDEF3 – это метод, предоставляющий аналитикам средство для описания ситуации, в которой процессы выполняются как в определенной последовательности, так и описать объекты, участвующие совместно в одном процессе.

Техника описания набора данных IDEF3 является частью структурного анализа. В отличие от некоторых методик описаний процессов, IDEF3 не ограничивает аналитика чрезмерно жесткими рамками синтаксиса, что может привести к созданию неполных или противоречивых моделей.

IDEF3 может быть также использован как метод создания процессов. IDEF3 дополняет IDEF0 и содержит все необходимое для построения моделей, которые в дальнейшем могут быть использованы для имитационного анализа.

# **UML**

UML – это язык графического описания для объектного моделирования в области разработки программного обеспечения. UML является языком широкого профиля, это – открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML-моделью. UML был создан для определения, визуализации, проектирования и документирования, в основном, программных систем. UML не является языком программирования, но на основании UML-моделей возможна генерация кода.

Использование UML не ограничивается моделированием программного обеспечения. Данный язык также используют для моделирования бизнес- процессов, системного проектирования и отображения организационных структур.

UML позволяет разработчикам программного обеспечения достигнуть соглашения в графических обозначениях для представления общих понятий таких, как класс, компонент, обобщение, агрегация и поведение, а также больше сконцентрироваться на проектировании и архитектуре.

В UML используются следующие виды диаграмм:

1) структурные диаграммы:

- диаграмма классов;

- диаграмма компонентов;

- диаграмма композитной/составной структуры;

- диаграмма кооперации (UML 2.0);

- диаграмма развёртывания;

- диаграмма объектов;

- диаграмма пакетов;

- диаграмма профилей (UML 2.2).

2) диаграммы поведения:

- диаграмма деятельности;

- диаграмма состояний;

- диаграмма вариантов использования.

3) диаграммы взаимодействия:

- диаграмма коммуникации (UML 2.0);

- диаграмма обзора взаимодействия (UML 2.0);

- диаграмма последовательности;

- диаграмма синхронизации (UML 2.0)

Преимущества UML:

1)UML объектно-ориентирован, в результате чего методы описания результатов анализа и проектирования семантически близки к методам программирования на современных объектно-ориентированных языках;

2)UML позволяет описать систему практически со всех возможных точек зрения и разные аспекты поведения системы;

3) диаграммы UML сравнительно просты для чтения после достаточно быстрого ознакомления с его синтаксисом;

4)UML позволяет вводить собственные текстовые и графические стереотипы, а также применяется сфере программной инженерии;

5)UML получил широкое распространение и динамично развивается.

# **DFD**

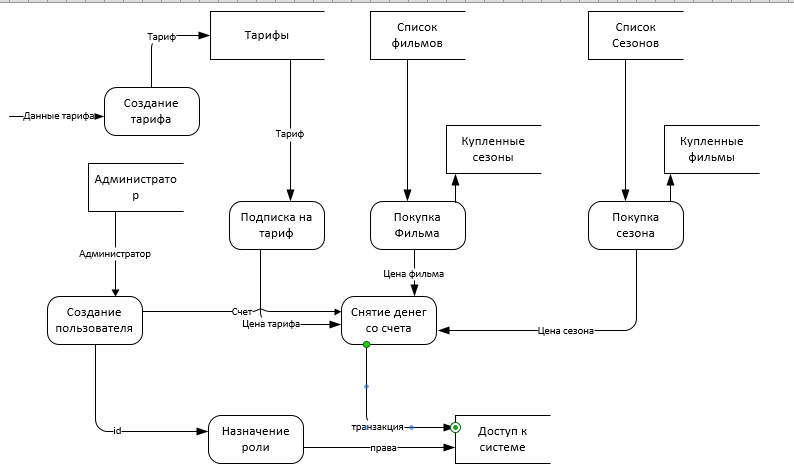
Диаграммы потоков данных (Data flow diagramming, DFD) используются для описания документооборота и обработки информации. Подобно IDEF0, DFD представляет модельную систему как взаимосвязанный набор действий, которые обрабатывают данные в «хранилища данных» как внутри, так и вне границ моделируемой системы. Их можно использовать как дополнение к модели IDEF0 для более наглядного отображения текущих операций документооборота в корпоративных системах обработки информации. DFD описывает:

функции обработки информации (работы);

документы (стрелки), объекты, сотрудников или отделы, которые участвуют в обработке информации;

внешние ссылки, которые обеспечивают интерфейс с внешними объектами, находящимися за границами моделируемой системы;

таблицы для хранения документов (хранилище данных).

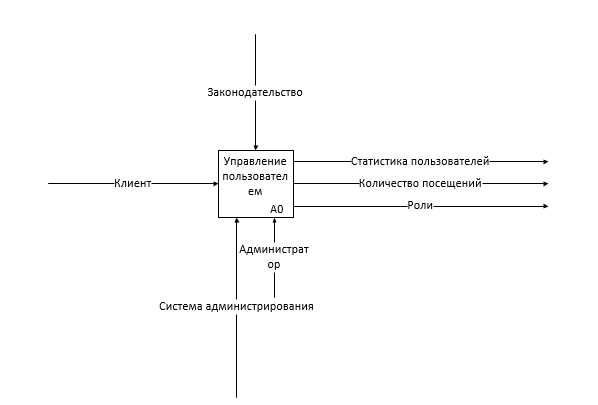


# **Проектирование системы администрирования для мобильного приложения «Онлайн кинотетр»**

# **Диаграмма IDEF0, UML**

IDEF0 — [методология](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D1%8F) функционального моделирования и графическая нотация, предназначенная для формализации и описания [процессов](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%B7%D0%BD%D0%B5%D1%81-%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81).

В данной диаграмме в виде входных параметров представлено задание, его атрибуты такие как имя, текст, приоритет, дата выполнения, фото, ссылки задания а также подчинённый который должен выполнить задание. (рис.2). Выходным параметром является отчет полученный руководителем от подчинённого о выполнение данного ему задания. В управлении используется как процессы получения и выполнения подчиненным данного задания. В механизме участвуют база данных для хранения и передачи задач.

Рис.2. Упрощённая диаграмма IDEF0

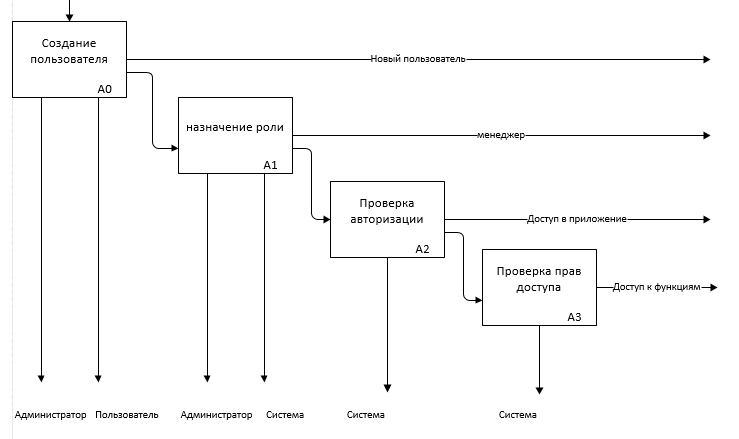


Рис.3. Полная диаграмма IDEF0

Полная диаграмма IDEF0 представляет собой полный жизненный процесс создаваемой задачи. На первом этапе происходит процесс создания задачи, согласно задаче поставленной руководителем. Входными данными для на данном этапе являются атрибуты задачи, а влияющим механизмом база данных устройства руководителя. На седеющем этапе объект заметки, который является входным значением, передается на сервер. Далее сервер посылает данный объект заметки подчиненному. На этих двух этапах участвует механизм протокола передачи данных и в управление используется параметры безопасной передачи данных. (рис.3)

Вывод

В данной части курсовой работы был произведён анализ предметной области который включает в себя область применения и технические требования. Также были реализованы диаграммы IDEF0 .

3.2 Создание логической модели, ERD-диаграммы

1. Описание предметной области.

Данное WEB приложение содержит данные о ролях и пользователях. Таблица задач содержит заметки и атрибуты данных заметок. Таблица клиентов содержит список пользователей их роли и связи.

2. **-** Список сущностей:

Администратор (Таблица 1).

|  |  |
| --- | --- |
| Сущность | Атрибуты |
| Пользователь | Фамилия  Имя  Логин  Пароль  Электронная почта  Пол  Дата рождения  Баланс |
| Роль | Название |

Таблица 1. Список сущностей

-Основные связи представлены в таблице 2.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Сущность | Связь | Сущность | Тип | Модальность |
| 1 | Заметка | Назначается | Пользователь | Многие -к-многим | Должен-может |
| 1 | Заметка | Получает | Роль | Многие-к-многим | Может-должен |

Таблица 2. Основные связи

Далее строится диаграмма сущность-связь по нотации Питера Чена.

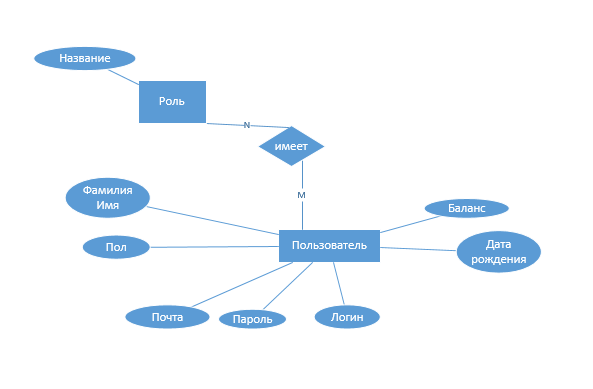


Рис.5. Диаграмма сущность-связь по нотации Питера Чена

Далее строится схема реляционной модели информационной системы. Сущность работы с базой данных сводится к созданию идентичных таблиц хранения данных.

Схема реляционной модели предметной области: "Система администрирования мобильного приложения Онлайн Кинотеатр" представлена на рисунке 6.

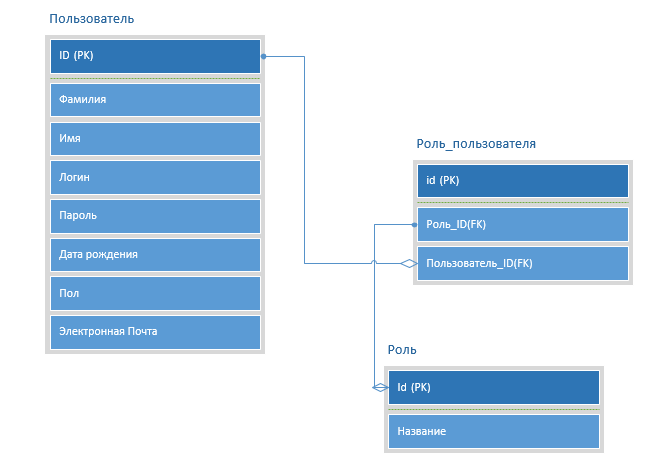


Рис.6. Схема реляционной модели

# **3.3 Диаграммы UML**

Одним из наиболее эффективных способов взаимодействия между заказчиком и исполнителем является создание Диаграмм UML. В частности, диаграммы прецедентов.

Имеется один участник: Администратор. Администратор создает нового пользователя, назначает ему роль и доступ к определенным ресурсам, также посредством API-контроллера, система позволяет регистрироваться через мобильное приложение. В обязанности администратора также входит развертывание сервера , расчет статистики, buckup и восстановление базы данных

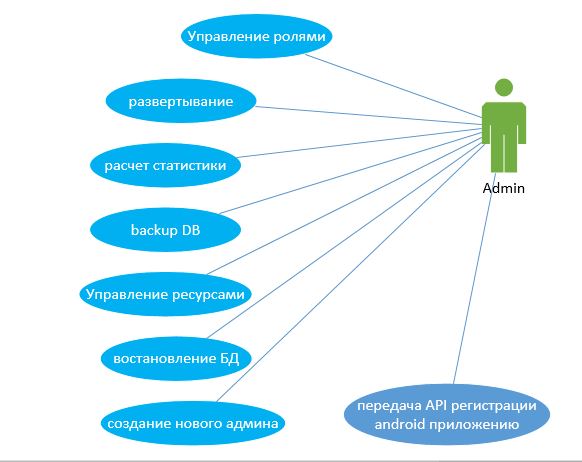
****

Рис.7. Диаграмма прецедентов (Use case diagram)

* 1. Диаграмма классов

Диаграмма классов представлена на рисунке 7.

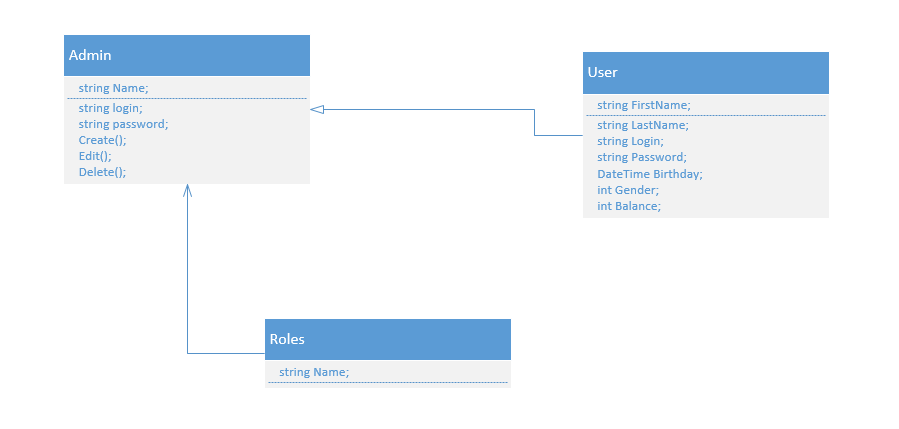


Рис.7. Диаграмма классов

Также создаётся диаграмма последовательности. Диаграмма последовательности — это [диаграмма](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0), на которой показано взаимодействие объектов (обмен между ними сигналами и сообщениями), упорядоченное по времени, с отражением продолжительности обработки и последовательности их проявления. Данная диаграмма представлена на рисунке 8.

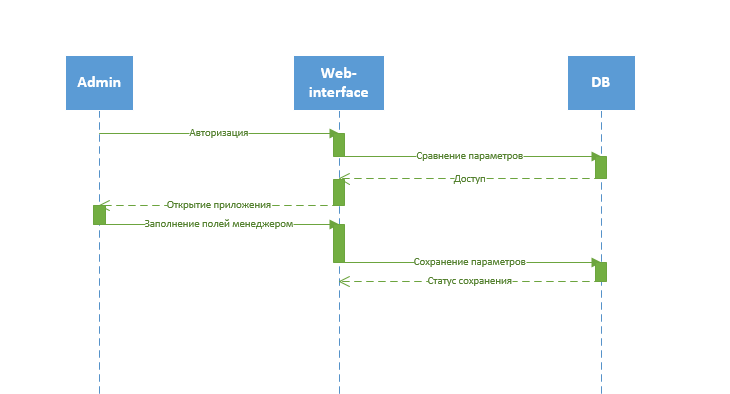


Рис.8. Диаграмма последовательности

В данном случае рассмотрена ситуация создания пользователя.

Заключение

Проделав курсовой проект, были закреплены теоретические и практические знания, полученные на занятиях по дисциплине «Проектирование информационных систем».

Процесс проектирования организационных систем основан на совместном применении взаимодополняющих методов. Одной из важнейших задач управления на современном этапе является исследование и совершенствование методологии проектирования организационных систем в соответствии с изменяющимися условиями.

Проектирование – это деятельность человека или организации по созданию проекта, то есть прототипа, прообраза предполагаемого или возможного объекта, состояния; комплекта документации, предназначенной для создания определённого объекта, его эксплуатации, ремонта и ликвидации, а также для проверки или воспроизведения промежуточных и конечных решений, на основе которых был разработан данный объект.

Список используемой литературы

1. Головничер М.Н. Проектирование информационных систем // Структура информационных систем. 2013. С. 46-78.
2. Котлер Ф., Келлер К.Л. Маркетинг менеджмент.12-е изд – СПб.: Питер, 2006 – 816 с.
3. Мхитарян С.В. Маркетинговая информационная система. – М.: Эксмо, 2006 – 336 с. – (Профессиональные издания для бизнеса)
4. Беленькая М.Н , Малиновский С.Т. , Яковенко Н.В. “АДМИНИСТРИРОВАНИЕ В ИНФОРМАЦИОННЫХ СИСТЕМАХ”
5. Материалы сайта: <http://www.bigspb.ru>