

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

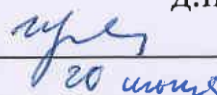
ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ НАУК
Кафедра программного обеспечения

ДОПУЩЕНО К ЗАЩИТЕ В ГЭК
И ПРОВЕРЕНО НА ОБЪЕМ
ЗАИМСТВОВАНИЯ

Заведующий кафедрой

д.п.н., профессор

И.Г. Захарова


20 июня

2016 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Разработка и оптимизация Web-приложений, работающих в режиме реального времени, в условиях ограниченных ресурсов.

02.04.03 Математическое обеспечение и администрирование информационных систем

Магистерская программа «Математическое обеспечение и администрирование информационных систем

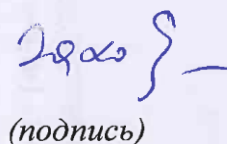
(Высокопроизводительные вычислительные системы)»

Выполнил работу
Студент 2 курса
очной формы обучения


(подпись)

Поляков
Димитрий
Андреевич

Научный руководитель
к.ф.-м.н., доцент


(подпись)

Захаров
Сергей
Дмитриевич

Рецензент
ведущий инженер-программист
Уральского ГУ
Центрального Банка РФ


(подпись)

Ялдыгин
Валерий
Борисович

Тюмень 2016

Оглавление

Введение.....	3
Глава 1. Проектирование веб-приложения.....	6
Анализ бизнес требований.....	6
Подбор технологической базы.....	9
Структура базы данных.....	15
Глава 2. Математическая модель модуля планирования карьеры.....	19
Персональные и карьерные показатели сотрудника.....	19
Компетенции.....	19
Ключевые показатели эффективности.....	21
Требуемые языки.....	23
Сфера деятельности.....	24
Построение графа карьерных перемещений.....	24
Глава 3. Оптимизация механизма планирования карьеры.....	30
Влияние начальных условий задачи.....	30
Определение магистрального направления.....	31
Эвристическая оценка охвата вершины.....	32
Алгоритм поиска кратчайшего пути.....	35
Заключение.....	40
Список литературы.....	42

Введение

2014 год отрицательно сказался на экономическом положении большинства стран мира. Определенный набор событий привел многие страны в положение, находящемуся на пороге инфляции и девальвации валюты, что сказалось на финансировании внутренних проектов многих организаций, и ПАО Сбербанк — не исключение.

Структура подразделений информационно-технологического обеспечения Банка и принцип организации работы устроены таким образом, что вся разработка продуктов сосредоточена в основном в дочерней компании Банка — ОАО «Сбербанк. Технологии», который, в свою очередь, не может быстро исполнять проекты, компенсирующие потребности локальных банков в инструментах работы как с данными клиентов, так и с внутренней информацией. Поэтому, разработка части инструментов ложится на плечи сотрудников офисов IT территориальных банков, которые в свою очередь ограничены в ресурсах для ведения разработки.

Ресурсы разработки IT-проектов это совокупность условий, которые дают возможность реализовать цели проекта. Далее, рассмотрим влияние экономического кризиса на каждом из условий:

- организационная структура и кадровый потенциал — как правило, любой финансовый кризис сказывается на кадрах, что в свою очередь ограничивает возможности в выделении сотрудников на тот или иной проект;
- бюджет — аналогичная предыдущему пункту ситуация, результат — исходный недостаток средств и возможностей;
- информационное обслуживание — это условие, на деле, выполняется практически всегда, вне зависимости от условий финансирования,

благодаря современному уровню развития интернет технологий и наполненности технических интернет-ресурсов целевой информацией;

- материально-техническая база проекта — наиболее важное условие. Зачастую, в результате ограничения средств, у компании отсутствует возможность выделить дополнительные мощности на работу проекта.

Целью моей работы является поиск методов разработки высоконагруженных веб-приложений, работающих в режиме реального времени, в условиях ограниченных ресурсов. Разработку подобного рода приложений можно представить в виде следующих этапов (задач):

- Изучение бизнес-требований к Web-приложению, поставленных организацией;
- Проектирование архитектуры веб-приложения с учетом имеющихся вычислительных ресурсов, определение наиболее загруженных модулей будущего приложения;
- Проектирование структуры базы данных, предназначенной для хранения больших объемов информации. В данной работе рассматривается база данных, в задачи которой входит хранение кадровой и персональной информации сотрудников отдельно взятого предприятия (ПАО Сбербанк), используемой для расчета трудовых показателей каждого сотрудника;
- Решение задачи оптимизации найденных «узких» мест приложения с дальнейшей реализацией.

Как можно видеть из списка задач, объектом моего исследования является разработка высоконагруженного веб-приложения для корпоративного использования с ограниченными вычислительными ресурсами. Предметом исследования являются методы реализации, без привлечения материальных

затрат, посредством формирования предложений карьерных перемещений сотрудников.

Для достижения результатов, были предприняты следующие шаги:

- Разработка технического задания согласно бизнес-требованиям к будущему приложению;
- Проектирование модульного веб-приложения;
- Анализ модулей приложения на предмет нагрузки, создаваемой модулем на приложение;
- Построение математической модели механизмов проблемных модулей;
- Решение задачи оптимизации указанных механизмов.

В результате, перед нами встает достаточно сложная задача — спроектировать и разработать приложение согласно бизнес-требованиям, практически не имея финансирования, мощностей и, что логично, в сжатые сроки.

Глава I. Проектирование веб-приложения

Анализ бизнес-требований

Для любого работника Сбербанка в зависимости от его должности определены факторы, отражающие эффективность работы этого сотрудника. Все эти показатели являются важной и неотъемлемой частью современной банковской системы, ведь чем лучше коэффициенты, тем лучше показатели банка в целом. В том числе каждого работника поощряют за выполнение показателей в денежном эквиваленте.

К сожалению, данная система не является идеальной сразу по нескольким причинам:

- В своей повседневной деятельности сотрудники занимаются, как правило, рутинной, скучной работой, что отрицательно сказывается на их производительности. Это влечет за собой ухудшение работы всей организации;
- Еще один существенный минус – это отсутствие систематичности. В нынешнем варианте вся история различных показателей хранится по разрозненным файлам на локальных компьютерах, что затрудняет подсчет показателей как по сотруднику, так и интегральных по отделам, управлениям и т.д. Это влечет за собой ряд неприятных последствий: ошибки, различные махинации, малая осведомленность сотрудников о своих же успехах, а руководителей об успехах подчиненных им ведомств.

Также, согласно концепции «Счастливый сотрудник — счастливый клиент», разработанной руководством ПАО «Сбербанк», главная задача топ-менеджеров банка — обеспечить своих сотрудников всем необходимым для работы, причем это касается не только материального обеспечения, но также и личной заинтересованности в выполняемой работе. Поэтому, когда встал вопрос

о повышении личной отдачи сотрудников в рамках Западно-Сибирского банка ПАО «Сбербанк» (далее просто ТБ — территориальный банк), руководством Банка в лице заместителей председателя – кураторов Розничного Блока и Блока по работе с персоналом было принято решение о разработке и внедрении корпоративной социальной сети, объединяющей в себе ряд инструментов для обеспечения нематериальной мотивации сотрудников.

В результате был инициирован проект по разработке и внедрению корпоративной социальной сети для сотрудников. Её целью является создание для сотрудников комфортных условий работы, предоставление ряда инструментов для общения между собой, а также инструмента для планирования собственной карьеры.

Согласно бизнес-требованиям, портал должен включать в себя следующие модули:

1. Социальная сеть, выполняющая следующие задачи:
 - а. Представление персональной и кадровой информации сотрудника;
 - б. Обеспечение сотрудникам возможности общения в реальном времени;
 - в. Представление подразделения ТБ как совокупность группы сотрудников, их показателей, графика обучения сотрудников;
 - г. Группы пользователей по интересам;
2. Модуль обработки показателей эффективности сотрудников и кадровой информации, включающих в себя:
 - а. Компетенции сотрудников (необходимый набор требований к профессиональным и личным навыкам сотрудников, стоящих на отдельно взятой должности);

- б. Ключевые показатели эффективности, являющиеся относительными коэффициентами выполнения служебных обязанностей;
 - в. Квартальные и годовые оценки сотрудников (далее просто Оценка 5+), состоящие из набора самооценок сотрудника и набора оценки сотрудника его линейным руководителем;
 - г. Перечень текущих вакансий ТБ;
 - д. Знание языков сотрудниками;
 - е. Личные навыки сотрудников;
 - ж. Инструмент реального времени, позволяющий формировать по запросу сотрудника наиболее оптимальный путь для достижения выбранной им карьерной позиции.
3. Модуль формирования тестов и опросов, а также сбор результатов и статистики;
 4. Модуль автоматической организации обучения сотрудников, стоящих на определенных должностях.

Из всех модулей проекта, наиболее сложной и одновременно наиболее необходимой частью проекта является реализация инструментов нематериальной мотивации сотрудников, включающих в себя инструменты поиска наиболее привлекательной должности для сотрудника с точки зрения объема времени, необходимого для достижения поставленной цели (карьерной позиции).

Данный модуль, согласно бизнес-требованию проекта должен представлять собой инструмент, предоставляющий сотруднику возможность поставить перед собой цель достижения определенной карьерной позиции. На

основе запроса сотрудника должен формироваться наиболее приемлемый для него путь достижения поставленной цели, исходя из требований к должностям, персональных качеств и карьерных показателей сотрудников. В виду того, что данный набор инструментов связан непосредственно с модулем автоматического обучения сотрудников, сотруднику должен в реальном времени предоставляться свод информации, демонстрирующий текущее состояние процесса его продвижения. Так как информация о карьерных перемещениях может меняться в реальном времени (это связано как с изменением базовых параметров должностей, так и с текущим положением дел в подразделении, через должность в котором хочет пройти сотрудник, а так же с отказами, которые может сформировать любой из руководителей), планируемый путь передвижения по должности необходимо периодически пересчитывать и предоставлять сотруднику обновленную информацию о процессе. Выглядит данный механизм довольно странно, однако, как показывает практика банковской среды, а так же опыт, накопленный крупными организациями, подобный подход помогает стимулировать к развитию как самих сотрудников, так и процесс управления персоналом в целом, и как следствие, оказывать влияние на все процессы, происходящие внутри организации.

В связи с описанными выше причинами, в ходе анализа спроектированных модулей было принято единогласное решение о том, что необходимо упростить и оптимизировать механизм формирования путей карьерных перемещений, о чем я и расскажу в своей работе.

Подбор технологической базы

Поскольку бизнес-требование к порталу подразумевает наличие большого числа модулей, работающих в асинхронном режиме, было предпринято решение о базировании архитектуры проекта на паттерне разработки веб-приложений под названием ASP .NET. Это технология, позволяющая реализовать веб-приложение

согласно концепции Model-View-Controller,

Ввиду масштабности проекта было принято решение об использовании вместо представлений и хранимых процедур использовать Entity Framework 6.0, который занимается автоматическим формированием запросов к базе данных, используя расширение синтаксиса языка запросов к коллекциям на основе лямбда-выражений.

Процесс функционирования портала был организован следующим образом:

- При публикации веб-приложения происходит инициализация объекта контекста, что необходимо для проверки целостности структуры используемой базы данных. По окончании проверки, сервер загружает необходимые настройки приложения;
- От клиента поступает запрос на предоставление информации. Данный запрос в контексте объекта класса `System.Web.Request` поступает в программный роутер приложения, где обрабатывается строка запроса приложения и происходит маппинг на нужный контроллер;
- Далее информация, хранящаяся в запросе попадает в контроллер. Данный контроллер может давать ответ одного из следующих типов – общее представление, частичное представление, контроллер формирования JSON-пакетов. Контроллер в соответствии с заложенными в него методами, получает по запросу необходимую информацию из базы данных и, в зависимости от типа возвращаемой информации, либо передает данные в представление, либо формирует из них JSON-пакет и передает его клиенту;
- При передаче информации на представление, в зависимости от типа представления, сформированная в нем информация по необходимости оборачивается в мастер страницу, являющейся DOM-шаблоном возвращаемого HTML-документа. После информация уходит конечному

пользователю.

Особняком стоит группа контроллеров, которые автоматически формируются методами классов, реализованных при помощи библиотеки SignalR, и обеспечивающих непрерывное поддержание дуплексного, либо полудуплексного канала связи с браузером клиента. Эти контроллеры общаются не непосредственно с браузером клиента, а с программной прослойкой, реализованной на языке JavaScript. Именно на группе таких контроллеров и реализованы модули портала, работающие в режиме реального времени, в частности модуль планирования карьеры, обеспечивающий сотруднику возможность формировать путь своего карьерного перемещения.

Эта группа контроллеров, как реализация целевых и, как следствие, наиболее нагруженных модулей приложения и является узким местом, оптимизацию работы которого нам необходимо провести. Рассмотрение механизма будет рассмотрено в главе 2.

В виду маломощности целевого сервера, были использованы следующие технологии и приняты следующие решения:

- ASP .NET SignalR – библиотека классов для обеспечения «режима реального времени»;
- Т.к. объем оперативной памяти сервера мал для хранения информации в сессии на серверной части приложения, было принято решение хранить публичную информацию, отображаемую на странице пользователя в сессии браузера клиента;
- Проведена оптимизация большинства LINQ-запросов, а также написано несколько расширений для LINQ to Entities;
- Все модули проекта переведены в асинхронный режим работы;

- Написана автоматическая конфигурация IIS сервера с целью корректно распределять нагрузку между доступными процессорными мощностями, а также модуль отложенной загрузки файловых потоков.

Реализованный портал представляет собой классическое веб-приложение, работающее в асинхронном режиме по схеме взаимодействия Model-View-Controller.

“Основная цель применения этой концепции состоит в отделении бизнес-логики (модели) от её визуализации (представления, вида). За счёт такого разделения повышается возможность повторного использования. Наиболее полезно применение данной концепции в тех случаях, когда пользователь должен видеть те же самые данные одновременно в различных контекстах и/или с различных точек зрения. В частности, выполняются следующие задачи:

- 1. К одной модели можно присоединить несколько видов, при этом не затрагивая реализацию модели. Например, некоторые данные могут быть одновременно представлены в виде электронной таблицы, гистограммы и круговой диаграммы;*
- 2. Не затрагивая реализацию видов, можно изменить реакции на действия пользователя (нажатие мышью на кнопке, ввод данных) — для этого достаточно использовать другой контроллер;*
- 3. Ряд разработчиков специализируется только в одной из областей: либо разрабатывают графический интерфейс, либо разрабатывают бизнес-логику. Поэтому возможно добиться того, что программисты, занимающиеся разработкой бизнес-логики (модели), вообще не будут осведомлены о том, какое представление будет использоваться. ”*

- Обобщенный MVC в исполнении C#. [6]

Выбор данного паттерна обусловлен тем, что данная архитектура, как разработка компании Microsoft, имеет полную поддержку всех доступных на данный момент средств обеспечения режима реального времени.

Так как целевые модули приложения требуют реализации в формате real-time web, одну их ведущих ролей занимает группа классов, наследованных от классов библиотеки SignalR, позволяющих автоматически сгенерировать интерфейс взаимодействия с клиентом посредством реализации группы контроллеров, тип взаимодействия которого определяется автоматически в

зависимости от доступных к использованию возможностей браузера. При помощи данной библиотеки реализованы следующие модули проекта:

- средство общения в реальном времени (чат, групповые сообщения);
- система блогов и управления подписками;
- уведомления о системных событиях;
- обеспечение непрерывного обновления информации и состоянии сформированного сотрудником пути перемещения до желаемой карьерной позиции.

После анализа всех планируемых модулей, исходных кадровых и персональных данных сотрудников, поиска и реализации решений, освобожденные от нагрузки ресурсы позволили реализовать real-time систему общения сотрудников. Для того, чтобы решить вопрос с передаче данных между клиентом и сервером в режиме реального времени, было предпринято решение об использовании библиотеки ASP .NET SignalR (архитектура взаимодействия на рис. 1).

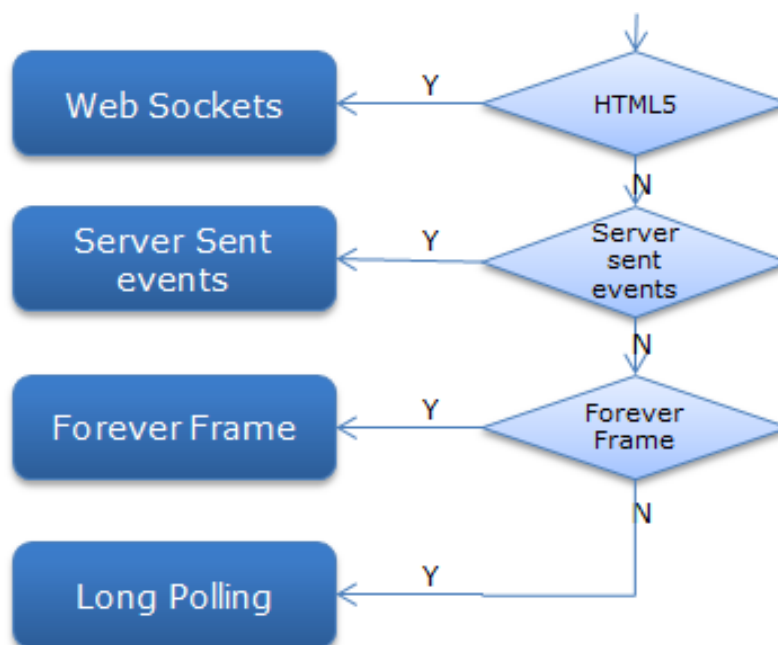


Рис. 1 – Схема выбора метода реализации real-time контроллеров приложения в зависимости от доступных браузеру технологий

Данная библиотека позволяет автоматизировать процесс общения между браузером клиента, серверной частью приложения и базой данных, за счет автоматической реализации процесса передачи данных в зависимости от того, какие из технологий реализации real-time поддерживает браузер клиента.

Классы, реализованные на основе классов, описанных в данной библиотеке, позволили создать внутри группы контроллеров веб-приложения группу методов, автоматически формирующих дуплексные и полудуплексные каналы передачи информации браузеру клиента. На основе этих методов в результате и были реализованы описанные ранее модули приложения, работающие в режиме реального времени.

Структура базы данных

Ввиду большого количества обрабатываемой информации, приведем описание только наиболее нагруженной части базы данных – карьера сотрудников:



Рис 2. Частичная схема построенной базы данных

Как мы видим, на данной схеме представлена часть структуры базы данных, отвечающей за представление информации о должностях ПАО Сбербанк. Данная структура связывает между собой такие сущности, как:

- Перечень сотрудников с указанием персональной информации;
- Должности сотрудников;
- Подразделения;
- Роли должностей;
- Уровни менеджмента должностей;
- Профили должностей, несущих в себе информацию о требованиях к кандидату на данную должность;
- Список формируемых системой возможных перемещений между карьерными позициями.

В данной работе наибольший интерес из перечисленной информации представляет информация о подразделениях, должностях, уровнях менеджмента и показателей сотрудников. Поскольку в наиболее загруженном модуле приложения – модуле планирования карьеры – используется вся перечисленная информация, рассмотрим реализацию этой части структуры базы данных более подробно:

Подразделения представляют собой сущности, внутри которых хранится информация о сфере деятельности, которой занимаются сотрудники данного подразделения и информация о материнском подразделении. Следовательно, структура таблицы представляет собой дерево, где каждая запись в таблице базы данных представляет собой узел дерева организационной структуры предприятия. На этом же уровне, но в другой таблице, хранится информация о конечных листьях дерева организации – должностях. Должность, помимо перечисленной в подразделении информации, несет в себе связь с таблицей, содержащей информацию об уровне менеджмента должности. Данная характеристика определяет уровень ответственности, перечень и уровни

профессиональных навыков, которым должен соответствовать сотрудник, находящийся на данной должности. Профессиональные навыки (компетенции) являются базовой сущностью для сущностей «текущий уровень навыка сотрудника» и «планируемое для должности значение».

В отношении рассматриваемого в работе модуля карьерного планирования, вся информация, необходимая для функционирования механизма построения путей перемещений хранится в указанных на рис.2.

Рассмотрим каждую из таблиц, представленных на рис. 2 подробнее:

`hf#employees` – хранит персональную информацию о сотрудниках организации, такую как ФИО, дата рождения, идентификатор должности, табельный номер, логин в доменной зоне организации, идентификатор файла с фотографией, время последнего входа в систему, рейтинг сотрудника и дату последнего обновления рейтинга и статус (Доступен, Занят, В отпуске);

`hf#departments` – хранит информацию о всех подразделениях организации. В нее входит название подразделения, идентификатор родительского подразделения, идентификатор блока организации, в которую входит данное подразделение, идентификаторы начальника и заместителя начальника подразделения, идентификатор города, в котором дислоцируется подразделение;

`hf#posts` – хранит информацию о группах должностей в разрезе подразделений. Это означает, что в рамках каждого отдельного подразделения, набор должностей с одинаковым названием, являясь набором карьерных позиций с одинаковым набором требований к должности, формирует так называемую «группу» должностей, каждая из которых является записью в таблице. В информацию о должности входит: Название, идентификатор родительского подразделения, разряд должности, уровень менеджмента, сфера деятельности;

`hf#competences` – компетенции, таблица с общей информацией о

существующих внутри организации профессиональных компетенциях. Хранит информацию о названии, дате создания, создателе, типе компетенции, а также её описание;

hf#competence_templates – шаблоны профессиональных компетенций групп должностей, формирующихся в зависимости от уровня менеджмента и сферы деятельности данной должности. Хранят информацию о компетенции и требуемых уровнях каждой из компетенций в виде экспертной оценки в диапазоне от 0 до 4;

hf#post_profile – содержит дополнительные требования к должности, такие как перечень и уровни знания требуемых языков, описание должностных обязанностей и необходимый опыт;

hf#promotions – хранит возможные карьерные перемещения между должностями. Включает в себя начальную и конечную должности, а также число, определяющее сложность перехода.

Остальные таблицы являются либо словарями, либо таблицами связей.

Взаимодействие с базой данных реализовано с использованием библиотеки Entity Framework 6.0, позволяющей взаимодействовать с хранилищем данных посредством Linq to Entities, без использования представлений и хранимых процедур. Это позволило как уменьшить время разработки, так и оптимизировать процесс получения данных из хранилища.

Для реализации портала была сформирована база данных, состоящая из 87 таблиц, 67 из которых представляют собой сущности, 20 из которых являются таблицами связи. Таблицы содержат в сумме 319 полей, 64 первичных ключа, 114 связей по дополнительным ключам.

Глава 2. Математическая модель модуля планирования карьеры

Персональные и карьерные показатели сотрудника

В рамках ПАО Сбербанк у каждого сотрудника существует следующий набор показателей:

- **Профессиональные компетенции.** Этот показатель демонстрирует профессиональные навыки сотрудника в той или иной области, в связи с чем его можно сравнить с понятиями «знания и умения»;
- **Ключевые показатели эффективности.** Этот показатель демонстрирует, насколько эффективно сотрудник выполняет свои должностные обязанности, что позволяет сравнить КПЭ с понятием «навыки»
- **В качестве персональных навыков сотрудника, влияющих на алгоритм подбора должности, стоит указать перечень языков, которыми владеет сотрудник и сфера деятельности, в которой он имеет образование.**

Каждый из указанных показателей принимается в расчет при поиске наиболее подходящей сотруднику должности.

Помимо показателей сотрудника, в алгоритме обязательным образом участвует шаблон требований, присущий каждой должности индивидуально. Данный шаблон в организации называется «профиль должности». В этот профиль входят следующие требования:

- **Шаблон компетенций с необходимыми уровнями соответствия в зависимости от опыта работы на данной должности (что говорит нам о том, что даже после вступления в должность перед сотрудником стоит задача расти и развиваться)**
- **Перечень ключевых показателей эффективности, которые сотруднику**

придется выполнять, находясь в данной должности. Здесь стоит отметить, что данное требование не обязательно при расчете соответствия сотрудника, однако при долгосрочном планировании карьеры стоит учесть следующий фактор: в случае, если должность является «транзитной» на пути к желаемой, сотруднику придется развивать текущие компетенции, т. к. у вышестоящих должностей необходимый опыт работы сотрудника растет, тогда как период обязательного повышения компетенции постоянен. С учетом того, что несоответствие компетенциям сотрудника влияет на оценку за период, это может оказаться критичным фактором;

- Сфера деятельности, в которой сотрудник должен иметь высшее образование;
- Дополнительные языки, которыми должен владеть сотрудник, с указанием уровня знания этих языков.

Теперь сравним перечень показателей сотрудника с обязательными требованиями к должности:

Компетенции

Как было сказано ранее, компетенция представляет собой профессиональный навык. Каждый сотрудник, имея компетенцию, имеет указанный уровень владения данной компетенцией. В системе Сбербанка существует 5 основных и 4 промежуточных уровня владения данной компетенцией:

0 – полное отсутствие навыка;

1 – базовый уровень навыка;

2 – владение на уровне специалиста;

3 – повышенный уровень владения навыком;

4 – уровень топ-руководителей.

Промежуточные уровни представляют собой дробные значения с шагом в 0,5.

Данные значения не являются порядковыми данными, как может показаться при первом взгляде. Дело в том, что данные оценки являются приведением интервальных данных, основанных на экспертной оценке, в дискретную форму по следующему принципу:

$$c_k^d = \begin{cases} 0, & \text{если } 0 \leq c_k^i \leq 0.25; \\ 0.5, & \text{если } 0.25 < c_k^i \leq 0.75; \\ 1, & \text{если } 0.75 < c_k^i \leq 1.25; \\ 1.5, & \text{если } 1.25 < c_k^i \leq 1.75; \\ 2, & \text{если } 1.75 < c_k^i \leq 2.25; \\ 2.5, & \text{если } 2.25 < c_k^i \leq 2.75; \\ 3, & \text{если } 2.75 < c_k^i \leq 3.25; \\ 3.5, & \text{если } 3.25 < c_k^i \leq 3.75; \\ 4, & \text{если } 3.75 < c_k^i \leq 4; \end{cases}$$

где c_k^d – дискретное значение k – той компетенции в наборе,

c_k^i – интервальное значение k – той компетенции в наборе

Подобное приведение от экспертной оценки к дискретным данным, позволяет нам осуществлять базовые операции сложения и умножения на элементами данного множества.

Тогда для каждой должности, её шаблон компетенций определим, как набор:

$$C = \{c_1, \dots, c_m\},$$

где c_i для $i \in [1, \dots, M]$ – уровень i – той компетенции в требованиях

Ключевые показатели эффективности

На этапе трудоустройства эти показатели являются необязательными для расчета соответствия сотрудника должности, однако, т.к. данные показатели, как обязательные к выполнению, могут принести неудобства с точки зрения карьерного роста сотрудника, использование их в качестве дополнительных данных желательно. Поэтому, перечень КПЭ должности будем использовать как параметр сортировки в алгоритме поиска кратчайшего пути, как показатель сравнения двух узлов с одинаковыми весами следующим образом:

Определим сущность ключевого показателя эффективности. Ключевой показатель эффективности представляет собой плановое значение, которое является результатом экспертной оценки. Его плановое значение влияет на результат работы сотрудника за период следующим образом:

1. Если КПЭ является прямым показателем, то считаем, что сотрудник выполнил норму, если его фактическое значение по данному показателю выше планового;
2. Если КПЭ является реверсивным показателем, то считаем, что сотрудник выполнил норму, если его фактическое значение по данному показателю приближается к плановому, но не выше его.

На основе вышеизложенного, определим сложность КПЭ, как статистическую величину:

Пусть

$$P_e^k = \frac{\sum_{i=1}^n F_i^k}{n}, i \in [1, \dots, n] -$$

где P_e^k – процентное выполнение КПЭ k за n периодов сотрудником e

Тогда

$$a_k = \frac{\sum_{e=1}^n p_e^k}{n},$$

где a_k – сложность КПЭ k .

Таким образом, положим, что сложность отдельного КПЭ представляет собой величину, определяющую средний процент выполнения данного КПЭ сотрудниками. В учет берутся только сотрудники, которые имеют данный КПЭ в своих должностных обязанностях.

В результате получим перечень показателей с поставленным каждому показателю в соответствие уровень сложности, определяющийся по правилу, описанному выше. Проиндексируем показатели в соответствии с сортировкой и определим полученный индекс, как уровень сложности (A_k).

Тогда для всей должности, сложность её будет определяться как вектор:

$$A = \{a_1, \dots, a_n\},$$

где a_i для $i \in [1..N]$ – сложность i – того КПЭ

Также, полученное значение можно использовать, чтобы осуществлять поиск наиболее подходящих сотруднику должностей, сортируя их по сумме уровней сложности выполняемых КПЭ.

Требуемые языки

Влияют на вероятность благоприятного исхода при принятии на должность. Однако, в случае, если сотрудник вообще не знает требуемого языка, вероятность успешного принятия на работу стремится к нулю. Следовательно, наиболее полезными из всех рекомендаций, сформированных программой, будут

те, которые не содержат должностей с требованиями, на выполнение которых сотруднику понадобится минимальное количество времени. Поэтому в графе всех существующих перемещений будем выделять взвешенный ориентированный подграф из должностей, которые не содержат языков, которые сотрудник не знает вообще.

Для ясности, обозначим данный показатель следующим образом в виде набора:

$$L = \{l_1, \dots, l_n\},$$

где l_i для $i \in [1, \dots, N]$ – уровень знания i – того языка в требованиях

Сфера деятельности

ПАО Сбербанк – крупное предприятие, и вполне ожидаемо, что его сотрудники работают в самых разных направлениях. Это касается и требования к образованию сотрудников. Сотруднику будет гораздо сложнее вести свою карьеру в подразделении (на должности) со сферой деятельности, отличной от его образования. Поэтому данный показатель мы также будем использовать в алгоритме. Обозначим его следующим образом:

S

Таким образом, наша математическая модель теперь включает в себя описание должности, как следующий набор параметров:

$$V = \{S; L = \{l_1, \dots, l_n\}; A = \{a_1, \dots, a_n\}, C = \{c_1, \dots, c_m\}\}$$

Построение графа карьерных перемещений

В ходе составления бизнес требований к проекту было сформулировано общее правило построения схемы карьерных перемещений внутри организаций, которое строится на таком понятии должности, как грейд. Грейд – это разряд

должности, целое число, которое определяет множество карьерных показателей сотрудника, такие, как его зарплата, уровень менеджмента и опыта.

Правило гласит, что для большинства должностей (преимущественное большинство) карьерное перемещение возможно в обе стороны, с условием максимальной разницы между грейдами должности меньше, либо равно двум. Так как сотрудника в большинстве случаев не интересует понижение в должности, для оптимизации процесса будем использовать это правило при построении графа только в положительную сторону.

Сам граф карьерных перемещений, ввиду отсутствия четкого математического определения понятия «карьерное перемещение», будем строить, исходя из существующей истории карьерных перемещений сотрудников между должностями за последние 30 лет (данная информация оцифрована и хранится в базе данных).

Определим, что:

1. Вершинами графа являются группы должностей, сгруппированных по подразделениям и названию;
2. Ребром между двумя вершинами называется карьерное перемещение между двумя должностями.

В результате подсчета статистики карьерных перемещений, получим следующие значения:

Сущность	Общее число
Должности	6040
Перемещения	3430

Используем понятие уровень менеджмента, заложенное в сущность

должности. Согласно ему для каждой должности формируется шаблон профессиональных компетенций по определенному правилу. Благодаря этой информации, для немассовых специальностей за вес ребра мы можем использовать разницу между шаблонами компетенций сравниваемых должностей. В представлении нашей математической модели это будет модуль разницы двух векторов. Переобозначим компетенции левой и правой должностей в рамках одного карьерного перемещения следующим образом:

Для исходной должности:

$$C_l = \{f_1, \dots, f_n\}$$

Для конечной должности:

$$C_r = \{t_1, \dots, t_m\}$$

$$\text{причем } m \geq n$$

Тогда ребро между двумя вершинами определим, как набор:

$$C_e = C_l - C_r = \{t_1 - f_1, \dots, t_n - f_n, t_{m-n}, \dots, t_m\} = \{r_1, \dots, r_m\},$$

где C_e – набор разниц повышаемых при перемещении компетенций,
 r_i для $i \in [1, \dots, M]$ – разница между одинаковыми компетенциями
 исходной и конечной должностей

Логично, что профессиональные навыки различаются между собой по степени сложности, однако данный параметр в системе не определен, поэтому вес ребра определим, как сумму всех элементов набора:

$$W_{f,t} = \sum_{i=1}^m r_i$$

Однако, распределение должностей в организации устроено таким образом,

что должностей с меньшими разрядами гораздо больше, чем должностей с больше. Это видно на следующем рисунке:



Рис. 3. Распределение должностей по разрядам

Так как обязательно будут существовать ребра между вершинами с одинаковыми наборами компетенций (из-за одинакового уровня менеджмента), определим для них ребро как разницу векторов сложности ключевых показателей эффективности:

$$W_{f,t} = A_t - A_f = \sum_{i=1}^n a_i^t - a_i^f + \sum_{i=m-n}^m a_i^t,$$

где вторая сумма – это сумма сложностей КПЭ,
отсутствующих в наборе первой должности

Теперь построим граф, используя в качестве узлов должности в подразделениях, а за ребра – статистику карьерных перемещений:

	Общее число	Задействованных на графе
Должности (узлы)	6040	2706
Перемещения (ребра)	3430	3430

Как видим, не все узлы нашего графа объединены ребрами. Для каждого из них, согласно правилу о грейдах, сформируем недостающие пути:

	Общее число	Задействованных на графе	Со сформированными по правилу
Должности (узлы)	6040	2706	6040
Перемещения (ребра)	3430	3430	25837

Теперь в нашем распоряжении находится взвешенный ориентированный граф карьерных перемещений, который мы можем начинать использовать для того, чтобы выполнять поиск кратчайшего пути до желаемой сотрудником должности.

Математическая модель графа в таком случае будет выглядеть следующим образом:

$$G = (V, E),$$

где G – упорядоченная пара,

V – непустое множество вершин (групп должностей),

E – множество упорядоченных пар вершин,

каждой из которых поставлен в соответствие вес (сложность перемещения)

Причем:

$$V = \{ S; L = \{l_1, \dots, l_n\}; A = \{a_1, \dots, a_n\}, C = \{c_1, \dots, c_m\}, V_u \},$$

$$E_{v_1, v_2} = \{V_1, V_2, W_{v_1, v_2}\},$$

$$\text{где } W_{v_1, v_2} = \begin{cases} \sum_{i=1}^n a_i^{v_1} - a_i^{v_2} + \sum_{i=m-n}^m a_i^{v_2}, & \text{если } C_{v_1} = C_{v_2} \\ \sum_{i=1}^m c_i^{v_1} - c_i^{v_2} & , \text{ если } C_{v_1} \neq C_{v_2} \end{cases},$$

V_u – вершина – родитель, должность, являющаяся линейным руководителем рассматриваемой

Определения всех элементов, представленных в конечной модели, были даны ранее в этой работе.

Теперь перейдем к решению задачи поиска кратчайшего пути на данном графе.

Глава 3. Оптимизация механизма планирования карьеры

Влияние начальных условий задачи

Еще раз рассмотрим граф, который мы получили в предыдущем пункте. Это взвешенный ориентированный граф, вершинами которого являются должности, ребрами – карьерные перемещения между должностями.

Главным этапом механизма планирования карьеры является поиск наиболее оптимального для сотрудника пути достижения выбранной им карьерной позиции. В интерпретации этот этап представляет собой задачу поиска кратчайшего пути от текущей должности сотрудника до конечной, выбранной им же.

На механизм отбора подходящих сотруднику должностей влияют описанные в начале второй главы показатели должности и сотрудника. Рассмотрим формулировку должности в построенной нами математической модели:

$$V = \{S; L = \{l_1, \dots, l_k\}; A = \{a_1, \dots, a_n\}, C = \{c_1, \dots, c_m\}, V_u\}$$

Сотрудника же, в свою очередь, представим в виде набора следующих параметров:

$$X = \{S_x, L_x = \{l_1, \dots, l_n\}\},$$

где X – сотрудник,

S_x – образование сотрудников,

L – набор известных сотруднику языков

Основываясь на полученных данных, процесс поиска оптимального пути теперь представляет собой **определение магистрального направления** и **вычисление эвристической оценки охвата** как предварительные этапы и **поиск кратчайшего пути** как основной этап.

Определение магистрального направления

Данный этап представляет собой предварительные манипуляции с исходным деревом. Дело в том, что сотруднику, имеющему образование в конкретной сфере деятельности, будет сложно продвигаться по карьерной лестнице в другой сфере. Например, сотруднику с высшим экономическим образованием в банковской сфере будет сложнее получить должность начальника офиса IT-активов, нежели сотруднику с высшим техническим образованием. За этот фактор отвечают параметры S и S_x для должности и сотрудника соответственно.

Помимо сферы деятельности, у должности, как оговаривалось ранее, хранится набор требуемых языков. В случае, если конечная должность, выбранная сотрудником, требует знания большего числа языков (неизвестных ему), включим данные языки в набор сотрудника, полагая, что он (сотрудник) сделал выбор, исходя из предоставленной информации о конечной должности:

$$X = \{S_x, L_x = \{L\} \cdot \{L_t\}\}$$

Учитывая вышесказанное, уменьшим исходный граф, обрезав все должности, отличные от сферы деятельности, в которой сотрудник имеет образование и требующие знание языков, неизвестных сотруднику:

Для сотрудника

$$X = \{S_x, L_x\},$$

Сформируем граф G_1 , такой, что:

$$G_1 = (V = \{S; L; A; C\}, E = \{V_i, V_j, W_{v_i, v_j}\}),$$

где $S = S_x$ и $L \in L_x$.

Отдельно отметим, что внутри графа существуют группы вершин, связанных между собой **вертикалью подчинения**. Это группы вершины, связанных между собой ребрами, таким образом, что каждая последующая должность, формирующая вершину, является руководящей для предыдущей, а также имеет разряд строго больше.

В результате, получим подграф, который назовем «магистралью сферы деятельности». В рамках этого графа нам и придется решать задачу поиска оптимального пути.

Эвристическая оценка охвата вершины

Для решения поставленной задачи необязательно решать полную задачу поиска кратчайшего пути. В данной работе я использую эвристический метод оценки охвата вершин (алгоритм Reach). В рамках текущей задачи, напомним, граф представляет собой хранилище данных о возможных карьерных перемещениях между должностями, с использованием разности требований к должностям в качестве веса ребра.

В исходной модификации (алгоритм Reach [1]) вводится понятие **охвата** вершины, представляющего собой две величины: относительно вершины, которая лежит на кратчайшем пути, это число представляет собой:

$$r(v, P) = \min\{dist(s, v), dist(v, t)\},$$

где v – вершина,

P – кратчайший маршрут,

s – исходная вершина,

t – конечная вершина,

Относительно всего графа существует понятие охвата графа, являющееся максимумом по всем кратчайшим путям, содержащим вершину v :

$$r(v) = \max_p r(v, P)$$

Помимо этого также существует понятие охвата ребра между вершинами v, w , которое представляет собой минимальное среди расстояний:

$$r(v, w) = \min\{dist(s, v), dist(v, t)\},$$

Аналогично охват ребра относительно графа:

$$r(v, w) = \max_P r((v, w), P) -$$

где $r(v, w)$ – максимум по всем кратчайшим путям, содержащим ребро vw .

Используя указанные понятия, в ходе алгоритма, на шаге n , вершина удаляется, в случае, если её охват относительно графа меньше, чем минимум среди суммы дистанций через транзитную вершину:

$$dist(s, v) + dist(v, w),$$

где w – целевая вершина,

и оценкой расстояния ко конечной вершины. Причем оценка расстояния до конечной вершины должна быть подобрана таким образом, чтобы если текущий маршрут – кратчайший, то условие удаления вершины не выполняется.

В виду особенностей задачи, модифицируем алгоритм следующим образом. Охват вершины будем вычислять только относительно всего графа, за оценку приняв коэффициент разницы между суммами компетенций начальной и конечной должностей.

Интерпретация: Эмпирическое наблюдение за карьерными перемещениями сотрудников позволяет полагать, что сотруднику необходимо предварительно освоиться в текущей сфере деятельности и набраться соответствующего опыта, для того, чтобы в дальнейшем не испытывать трудностей при переходе на более высокие должности (т.е. руководящие). Также, анализ требований к существующим должностям показал, что с определенного разряда, разница между должностями более не включает в себя сугубо

профессиональные навыки, то как банковское дело, или системное администрирование. Разница между руководящими должностями состоит в основном в уровне руководящих компетенций, включающих в себя менеджмент, управление процессами и командную работу. Это позволяет нам изначально поставить алгоритму задачу сформировать сотруднику начальный путь в рамках некой оценки таким образом, чтобы первоначальные изменения в уровне профессиональных компетенций происходили как можно «мягче» для сотрудника.

На примере нашего графа, рассмотрим ход алгоритма:

1. На основе параметров начальной и конечной вершины формируем оценку;
2. В области начальной вершины вычисляем её охват согласно оценке и формируем подграф;
3. Строим кратчайший путь в полученном подграфе до должности с максимальным разрядом;
4. Остальной путь для сотрудника формируем из вершин, стоящих на вертикали подчинения.

В результате осталось лишь подобрать оптимальный для данной задачи алгоритм решения классической задачи поиска кратчайшего пути.

В качестве результата параграфа рассмотрим общее описание рассмотренной здесь эвристики:

1. Если текущая позиция сотрудника находится вне целевой сферы образования, решим задачу поиска кратчайшего пути на подграфе, сформированном на вершинах в окрестности позиции сотрудника, с приоритетом на должности, разряд которых выше текущего разряда сотрудника. Т.к. на нашем графе существуют ребра, сформированные согласно правилу о разнице разрядов, такой путь существует всегда;

2. Далее, сократим число узлов оставшегося графа за счет отсеивания «неблагоприятных» вершин – тех позиций, которые находятся вне интересующей сотрудника сферы деятельности и тех, для работы на которых требуется знание новых языков. Должности, где есть язык, известный сотруднику, но с меньшим уровнем знания, не рассматривается, как отсутствующий;
3. За оценку охвата вершины примем следующее правило: т.к. шаблон профессиональных компетенций сотрудника подчиняется правилу формирования согласно уровню линейного менеджмента, примем за оценку охвата сумму разниц между бланком низшего и высшего уровней менеджмента;
4. Решим задачу поиска кратчайшего пути на построенном подграфе.

Алгоритм поиска кратчайшего пути

В моей работе применен алгоритм Дейкстры для нахождения кратчайшего пути в графе. Также, для проведения сравнительного анализа скорости поиска, рассмотрены алгоритмы Флойда и Левита.

Рассмотрим сложность вычислений алгоритма Дейкстры с использованием двоичных и фибоначчиевых куч:

В простейшем случае, когда для поиска вершины с минимальным суммарным весом пути просматривается всё множество вершин, а для хранения весов используется массив, время работы алгоритма Дейкстры есть:

$$O(V^2)$$

Основной цикл выполняется порядка V раз, в каждом из них на нахождение минимума тратится порядка V операций. На циклы по соседям каждой посещаемой вершины тратится количество операций, пропорциональное количеству рёбер E (поскольку каждое ребро встречается в этих циклах ровно

дважды и требует константное число операций). Таким образом, общее время работы алгоритма:

$$O(V^2 + E),$$

но, так как

$$E \leq V(V - 1),$$

оно составляет

$$O(V^2)$$

Использование такой структуры данных, как двоичные кучи позволяет снизить число вершин, которые необходимо рассмотреть, что снижает сложность алгоритма на порядок, за счет снижения количества операций сравнения, что снижает верхнюю оценку сложности:

$$O(E \log V)$$

Использование же Фибоначчиевых куч позволяет получить наилучшую оценку:

$$O(V \log V + E)$$

Помимо описанных выше модификаций алгоритма Дейкстры существует реализация с использованием многоуровневых корзин данных, позволяющих определять приоритетные вершины не только для текущего шага, но и для каждой ветви процесса построения кратчайшего пути в целом.

Модификация с использованием многоуровневых корзин работает за счет использования в алгоритме Дейкстры куч данных, которые представляют собой очереди вершин, сортируемых по суммарному весу ребер. В качестве структуры хранения данных принимается приоритетная очередь вершин, в которую добавляются вершины при прохождении. На этапе релаксации производится оценка суммарного веса ребер, и, если, по условию из алгоритма Дейкстры,

вершина не проходит для прохождения на данном этапе, она удаляется из текущей очереди и добавляется в новую корзину. Тем самым, корзины формируют собой приоритетные очереди, поиск по которым напоминает поиск в глубину.

При такой реализации, сложность работы алгоритма можно оценить как:

$$O(E + V + V\sqrt{C}),$$

где C – максимальный вес ребра в графе

Три рассмотренных модификации относятся к различным реализациям алгоритма Дейкстры. Помимо него, существуют два других алгоритма, решающих такую же задачу: это алгоритм Флойда и алгоритм Левита.

Первый алгоритм вызывает сложности с практической реализацией, т.к. необходим большой объем памяти, чтобы хранить информацию о матрицах путей и матрицах расстояний: его сложность оценивается следующим образом:

Три вложенных цикла содержат операцию, исполняемую за константное время:

$$\sum_{V,V,V} O(1) = O(V^3)$$

Помимо этого, существует решение, где хранятся значения не всех вершин, а только значения для предыдущей вершины, так как следующее получается рекурсивно. Его сложность описывается, как:

$$\sum_{V,V,V} O(1) = O(V^2)$$

Второй алгоритм – алгоритм Левита заключается в хранении вершин в трех массивах – пустые (необработанные вершины), приоритетная очередь и обработанные вершины. Минус алгоритма заключается в том, что, в отличие от Дейкстры, данный метод использует несколько более сложное правило сравнения весов, а также существует вероятность того, что одну и ту же вершину придется обрабатывать дважды. В худшем случае сложность возрастает экспоненциально, но на практике сложность алгоритма описывается функцией:

$$O(E \log V)$$

Итак, на основе проведенных оценок, принимаем решение об использовании в качестве наиболее подходящего алгоритма поиска кратчайшего пути алгоритм Дейкстры с многоуровневыми корзинами, так как эта модификация имеет наилучшую временную оценку и среди всех используемых структур данных, для этой необходим минимальный объем памяти. В приложении к работе имеется таблица сравнения временных оценок в разрезе теоретической сложности алгоритма, с учетом времени, необходимого на запрос к базе данных и объема памяти, необходимого для хранения информации об обработанных вершинах и весах.

Стоит заметить, что в системе существуют случаи, когда у сотрудника нет подходящего образования для вступления на целевую позицию и случаи, когда сфера образования сотрудника отличается от сферы деятельности на целевой позиции. В этом случае поступаем следующим образом: формируем сотруднику путь от начальной должности, до ближайшей в подходящей сфере деятельности. В случае, если образование отсутствует вовсе, предлагаем сотруднику пройти по текущей сфере. На основе полученных данных, решим задачу поиска кратчайшего пути при помощи полученных коэффициентов на урезанном графе, при помощи модификации алгоритма Дейкстры с использованием «умной» приоритетной очереди – многоуровневых корзин.

Результатом проведенной работы стал следующий алгоритм:

1. Предварительно удаляем из графа все вершины, для которых:

$$S_x \neq S,$$

где S_x – образование сотрудника,

S – сфера деятельности должности

2. Для начальной вершины рассчитываем охват в рамках всего графа.

Пусть

$V_s = \{C_s, \dots\}$ – начальная вершина,

$V_t = \{C_t, \dots\}$ – конечная вершина

Тогда

$$K_c = (C_s - C_t)/K_z,$$

где K_c – общая разность компетенций (максимальный путь),

K_z – экспертный коэффициент, определяющий

процент от общей разницы, который необходимо получить,

чтобы сотрудник набрался опыта в сфере

3. При помощи информации об уровнях менеджмента определяем разряд, соответствующий вершине со следующими параметрами:

$$\bar{V}_x = \{\overline{C_x, \dots}\}, \text{ такое что}$$

$$C_x = C_s + K_c$$

4. Из должностей в наборе \bar{V}_x выбираем ту, у которой:

$$\forall V_w \in \bar{V}_x, A_w = \min\{A_x\},$$

где A – сложность выполняемых КПЭ

5. В окрестности начальной вершины V_s строим путь до вершины V_w ;

6. Достаиваем путь до конечной должности рекурсивно через V_u , где V_u – должность линейного руководителя.

В результате работы алгоритма получим приблизительное решение задачи поиска кратчайшего пути, определяющее наиболее оптимальный для сотрудника путь достижения должности с набором опыта в соответствующей сфере. Т.к. руководящих должностей меньше, чем должностей специалистов, сотрудники, воспользовавшиеся данным маршрутом перемещения, не попавшие на должность руководителя, однако, попадут в резерв подразделения на эту должность, что является побочным положительным эффектом.

Заключение

В ходе работы были выполнены следующие задачи:

- спроектирована и реализована архитектура Web-приложения, представляющая собой платформу для реализации неограниченного количества модулей, обеспечивающую должный уровень комфорта как самого приложения, так и процесса разработки;

- в ходе проектирования архитектуры выявлен наиболее загруженный модуль приложения;

- спроектирована и реализована структура базы данных, позволяющая обращаться к данным посредством Linq-запросов, с минимальными потребностями в вычислительных ресурсах;

- по результатам проведенных исследований и испытаний, проведенного анализа характеристик сотрудника как объекта, был разработан эвристический метод, предоставляющий приближенное решение задачи модуля карьерного планирования, который является наиболее сложной вычислительной задачей проекта.

Результатом работы стало корпоративное Web-приложение, представляющее собой социальную сеть, нацеленную на мотивацию сотрудников к карьерному росту в непринудительной форме. Приложение, благодаря проделанной оптимизации механизмов сбора, оценки и обработки информации, успешно функционирует на базе маломощного оборудования, технические характеристики которого представлены в работе. Благодаря отсутствию нагрузки со стороны наиболее загруженного модуля приложения, также была реализована система общения сотрудников, работающая в режиме

реального времени, которая повышает уровень комфорта рабочих процессов и личную заинтересованность сотрудников в карьерном росте.

Использованные в работе алгоритмы, методы оценки и подходы к анализу данных можно использовать для применения к построению подобных систем на базе кадровой информации других организаций. Помимо этого, представленный алгоритм по условиям использования подходит и для других прикладных задач в области управления процессами.

Данное исследование является первым шагом к реализации системы автоматического управления персоналом. В дальнейшем планируется автоматизация процесса обучения и стимулирования профессионального развития сотрудников посредством эвристической оценки успешности сотрудника в определенных сферах деятельности и построение на полученной математической модели для построения полуавтоматической системы управления персоналом.

Список литературы

1. Goldberg, Andrew V. «Reach for A*: an Efficient Point-to-Point Shortest Path Algorithm» – Massachusetts Institute of Technology – 2009.
2. В.И. Левин. Алгоритм проверки устойчивости решения оптимизационных задач в условиях неопределенности//Эвристические алгоритмы и распределенные вычисления; Самара: СамГУ – 2014.
3. В.В. Быкова. FTP-алгоритмы на графах ограниченной древовидной ширины//Прикладная дискретная математика; Т.: – 2012.
4. Алан Купер. Психбольница в руках пациентов; СПб: «Символ-Плюс» – 2009.
5. М.С. Назарцев, В.А. Козлов, С.В. Самсонов, В.А. Корнилов. Технология разработки многостраничных веб-приложений с межстраничной передачей параметров // Технические науки – от теории к практике; Новосибирск: 2015.
6. С. В. Рогачев. Обобщенный MVC в исполнении С#; Пермь: «Лукойл-Информ» – 2007.