

РЕФЕРАТ

Пояснительная записка 128 с., 49 рис., 9 табл., 21 источник
Программное средство автоматизации приемной кампании БГУИР

Объектом исследования является программное средство автоматизации приемной кампании БГУИР.

Ключевые слова:

ASP.NET MVC, JavaScript, Microsoft SQL Server, приемная комиссия БГУИР, зачисление.

Цель работы — решение проблемы автоматизации и организации работы приемной комиссии университета по зачислению и управлению поданными абитуриентами документами, что позволит снизить временные затраты и в целом упростить организацию процесса управления приемной кампанией в БГУИР.

Предлагаемое программное средство позволяет исключить из процесса зачисления в университет человеческое вмешательство, упрощает работу по информированию абитуриентов о ходе приемной кампании, а также значительно упрощает поиск и обработку документов абитуриентов.

Проведен анализ достоинств и недостатков существующих программных продуктов. С их помощью разработаны и спроектированы функциональные требования к приложению.

На основе функциональных требований разработана архитектура программного средства и модель базы данных.

Разработаны тесты для проверки соответствия функциональным требованиям и корректности работы приложения.

Приведено технико-экономическое обоснование эффективности разработки и использования программного средства.

Отображены результаты исследования по обеспечению безопасных условий труда инженеров-программистов в БГУИР.

Разрабатываемое программное средство должно позволить упростить работу с поданными абитуриентами в ходе приемной кампании документами и автоматизировать процесс зачисления в университет, полностью исключив человеческое вмешательство.

ОПРЕДЕЛЕНИЯ И СОКРАЩЕНИЯ

В настоящей пояснительной записке применяются следующие определения и сокращения.

Абитуриент – лицо, изъявившее желание поступить в УВО для получения высшего образования.

Аутентификация – проверка подлинности предъявленного пользователем идентификатора.

Вертикальное масштабирование – увеличение производительности каждого компонента системы с целью повышения общей производительности.

Высшее образование – уровень основного образования, направленный на развитие личности студента, курсанта, слушателя, их интеллектуальных и творческих способностей, получение ими специальной теоретической и практической подготовки, завершающийся присвоением квалификации специалиста с высшим образованием, степени магистра.

Горизонтальное масштабирование – разбиение системы на более мелкие структурные компоненты и разнесение их по отдельным физическим машинам (или их группам), и (или) увеличение количества серверов, параллельно выполняющих одну и ту же функцию.

Инициализация – приведение областей памяти в состояние, исходное для последующей обработки или размещения данных.

Масштабируемость – способность системы, сети или процесса справляться с увеличением рабочей нагрузки (увеличивать свою производительность) при добавлении ресурсов.

Многопоточность – свойство платформы или приложения, состоящее в том, что процесс, порождённый в операционной системе, может состоять из множества потоков, выполняющихся независимо друг от друга, то есть без предписанного порядка во времени.

Правила – Правила приема в высшие учебные заведения.

Программа – данные, предназначенные для управления конкретными компонентами системы обработки информации в целях реализации определенного алгоритма.

Программное обеспечение – совокупность программ системы обработки информации и программных документов, необходимых для эксплуатации этих программ.

Программный модуль – программа или функционально завершённый фрагмент программы, предназначенный для хранения, трансляции, объединения с другими программными модулями и загрузки в оперативную память.

Подпрограмма – программа, являющаяся частью другой программы и удовлетворяющая требованиям языка программирования к структуре программы.

Спецификация программы – формализованное представление требований, предъявляемых к программе, которые должны быть удовлетворены при ее разработке, а также описание задачи, условия и эффекта действия без указания способа его достижения.

Фреймворк – программное обеспечение, облегчающее разработку и объединение различных компонентов большого программного проекта.

АСПЗиЗ – автоматизированная система подачи заявлений и зачисления

БГУИР – Белорусский государственный университет информатики и радиоэлектроники

БД – база данных

ВУЗ – высшее учебное заведение

ООП – объектно-ориентированное программирование
ОС – операционная система
ПК – приемная комиссия
ПС – программное средство
РИКЗ – Республиканский институт контроля знаний
СУБД – система управления базой данных
УВО – учреждение высшего образования
УО – учреждение образования
ЦТ – централизованное тестирование
ЭВМ – электронная вычислительная машина
AJAX - Asynchronous Javascript and XML (асинхронный JavaScript и XML)
JSON – JavaScript Object Notation
SQL – Structured Query Language (язык структурированных запросов)
URL – Uniform Resource Locator (единообразный локатор ресурса)

ВВЕДЕНИЕ

Работа приемной комиссии является такой сферой деятельности, где очень важна автоматизация работы с документами. В ходе приема документов у абитуриентов рабочей группе приемной комиссии приходится обрабатывать информацию из заявлений, документов, удостоверяющих личность, документов об образовании, сертификатов о прохождении централизованного тестирования, справках о льготах, документов, подтверждающих право абитуриента на зачисление вне конкурса и других. Подобная работа требует большой сосредоточенности, внимательности и усидчивости. Однако использование компьютерных технологий позволяет значительно упростить процесс обработки информации, заменить бумажные архивы на базу данных и ускорить поиск необходимых сведений об абитуриенте в разы.

В БГУИР с 2011 года введена электронная система зачисления абитуриентов. Сам же программный комплекс «Электронный абитуриент» используется для автоматизации приема документов у абитуриентов с 2010 года [1]. Использование комплекса доказало, что время подачи документов при соответствующем разделении обязанностей рабочей группы приемной комиссии сокращается в разы, а процесс зачисления минимизирует влияние человеческого фактора на его результаты. В то же время при изменении правил приема в высшие учебные заведения достаточно переписать лишь часть программного кода, чтобы комплекс продолжил свою эффективную работу.

При всех своих достоинствах «Электронный абитуриент» пока позволяет производить зачисление только в рамках одного вуза. Использование сети Интернет же может позволить абитуриенту подавать документы одновременно в несколько вузов страны, указав в порядке приоритета несколько специальностей, как это уже сделано в БГУИР [2].

Целью данного проекта является разработка программного средства автоматизации деятельности приемной комиссии по приему документов и хранению информации об абитуриентах, позволяющего учитывать особенности работы приемных комиссий разных ВУЗов посредством гибкой настройки параметров работы системы через пользовательский интерфейс. Разработанное ПС должно быть доступно из любого места, подключенного к сети Интернет, а также иметь возможность расширения за счет модульной архитектуры.

Подобная разработка должна обладать повышенной надежностью, защищенностью и гибкостью. Личные данные абитуриентов должны быть максимально защищены от хищения и распространения. Работа в сети Интернет сопряжена с риском перехвата данных, особенно если работа производится в публичной локальной сети. Чтобы этого избежать, можно использовать шифрование передаваемых данных, как клиентское, так и серверное. Поддерживаемый всеми современными браузерами протокол передачи данных SSL также обеспечивает дополнительную защиту от утечки информации.

ВУЗы Республики Беларусь принимают документы в ходе приемной кампании по внутренним правилам приема документов. Так, БГУИР ежегодно публикует особенности приема документов в БГУИР, которые отличаются от предыдущих версий. Разрабатываемое программное средство должно быть гибким настолько, чтобы его можно было развернуть и быстро адаптировать к правилам приема любого ВУЗа Беларуси. Этого можно достигнуть за счет использования шаблонов и расширений. Основным документом для разработки технического задания необходимо считать «Кодекс Республики Беларусь об образовании» [3].

Для удобства использования интерфейс разрабатываемого программно-

го средства должен быть понятным и легко осваиваемым. Поэтому для решения задачи можно использовать современные возможности HTML5, CSS3 и популярного JavaScript фреймворка jQuery. В совокупности они позволяют реализовать адаптивные интерфейсы, которые одинаково хорошо отображаются на устройствах с различным разрешением (в том числе и на смартфонах).

Хранимые объемы данных даже после приема всех документов в ходе приемной кампании одного ВУЗа все равно остаются небольшими (подразумевается физическая память). Поэтому разрабатываемое программное средство должно иметь модуль для автоматического резервного копирования данных. Данную задачу может решить настройка серверного планировщика либо написание shell-скрипта, привязанного к определенному времени исполнения.

Таким образом, необходимо разработать программное средство, которое должно сочетать в себе удобство использования, надежность хранения данных и отказоустойчивость, позволяющее автоматизировать деятельность приемной комиссии, связанную с приемом, хранением и обработкой документов, как полученных от абитуриентов, так и внутренних, а также зачислением абитуриентов в ВУЗ.

АНАЛИЗ ПРОТОТИПОВ, ЛИТЕРАТУРНЫХ ИСТОЧНИКОВ ФОРМИРОВАНИЕ ТРЕБОВАНИЙ К ПРОЕКТИРУЕМОМУ ПРОГРАММНОМУ СРЕДСТВУ

Аналитический обзор литературы

Кодекс Республики Беларусь об образовании

Основным документом, которым руководствуются ВУЗы Республики Беларусь во время приемной кампании, является Кодекс Республики Беларусь об образовании (№ 243-З от 13 января 2011 г.) [3]. Этот документ регулирует образовательные отношения в стране и определяет основные положения приемных кампаний университетов в Республике Беларусь. Согласно Кодексу в учреждение высшего образования для получения высшего образования I ступени на все формы получения образования принимаются лица, имеющие общее среднее образование, профессионально-техническое образование с общим средним образованием или среднее специальное образование, а для получения высшего образования II ступени — лица, получившие высшее образование I ступени.

Прием лиц для получения высшего образования I ступени по специальностям (направлениям специальностей, специализациям), которые указаны в Правилах приема лиц для получения высшего образования I ступени, осуществляется после прохождения ими профессионального отбора, по итогам которого эти лица допускаются или не допускаются к участию в конкурсе по соответствующей специальности (направлению специальности, специализации).

Правила приема в высшие учебные заведения

Правила приема в высшие учебные заведения (Указ Президента Республики Беларусь от 20 марта 2014 г. № 130) — документ, непосредственно регулирующий порядок приема лиц для получения высшего образования I ступени в очной (дневной, вечерней) и заочной, в том числе дистанционной, формах получения образования в учреждения высшего образования Республики Беларусь независимо от подчиненности и формы собственности, за исключением УВО, находящихся в подчинении органов государственной безопасности Республики Беларусь, порядок приема в которые определяется Комитетом государственной безопасности [4].

Прием лиц, изъявивших желание поступить в УВО для получения высшего образования, в государственные УВО за счет средств бюджета осуществляется в соответствии с контрольными цифрами приема, которые утверждаются учредителями УВО по специальностям и формам получения высшего образования (в том числе для получения высшего образования в сокращенный срок по специальностям и формам получения высшего образования по образовательным программам, интегрированным с образовательными программами среднего специального образования по согласованию с Министерством образования в пределах средств, определяемых бюджетом).

Для организации приема лиц для получения высшего образования в учреждении высшего образования создается приемная комиссия, возглавляемая его руководителем. Приемная комиссия осуществляет свою деятельность в соответствии с Положением о приемной комиссии учреждения высшего образования, утверждаемым Министерством образования Республики Беларусь. В БГУИР действует Порядок приема в учреждение образования «Белорусский государственный университет информатики и радиоэлектроники» [5]. Данный порядок ежегодно утверждается ректором университета.

Абитуриенты, за исключением абитуриентов, указанных в пункте 12 Правил приема в высшие учебные заведения, подают в приемную комиссию УВО следующие документы:

заявление на имя руководителя УВО по установленной Министерством образования форме;

оригиналы документа об образовании и приложения к нему (за исключением лиц, поступающих для получения второго высшего образования);

оригиналы сертификатов централизованного тестирования, проведенного в Республике Беларусь в год приема;

медицинскую справку о состоянии здоровья по форме, установленной Министерством здравоохранения;

документы, подтверждающие право абитуриента на льготы при зачислении для получения высшего образования;

6 фотографий размером 3х4 см.

Кроме документов, перечисленных выше, в приемную комиссию при необходимости дополнительно представляются:

выписка (копия) из трудовой книжки, и (или) копия гражданско-правового договора, и (или) копия свидетельства о государственной регистрации индивидуального предпринимателя (для абитуриентов, поступающих для получения высшего образования в заочной или очной (вечерней) (за счет средств бюджета) форме получения высшего образования);

договор о целевой подготовке специалиста (рабочего, служащего) — для лиц, участвующих в конкурсе на условиях целевой подготовки специалиста;

копия диплома о высшем образовании с приложением, а для студентов УВО — письменное согласие руководителя УВО с места основной учебы

и справка о том, что гражданин является обучающимся (с указанием результатов освоения содержания образовательных программ высшего образования на момент выдачи справки) (для лиц, поступающих для получения второго и последующего высшего образования), а также справка о том, что высшее образование получено на платной основе (для лиц, поступающих для получения второго высшего образования за счет средств бюджета впервые);

рекомендация должностного лица, осуществляющего общее руководство сводной ротой почетного караула при подготовке и проведении государственных торжественных мероприятий (для лиц, указанных в абзаце восьмом части первой пункта 26 Правил);

ходатайство соответствующей федерации (союза, ассоциации) по виду (видам) спорта (для лиц, указанных в пункте 33 Правил);

копия удостоверения мастера спорта Республики Беларусь, мастера спорта Республики Беларусь международного класса (для лиц, указанных в части четвертой пункта 25 Правил).

Приемная комиссия имеет право дополнительно запросить у абитуриента документы, необходимые для принятия соответствующего решения.

Зачисление абитуриентов в УВО для получения высшего образования за счет средств бюджета и на платной основе на места по очной и заочной формам получения образования (за исключением специальностей, перечисленных в части второй настоящего пункта) проводится по конкурсу на основе общей суммы баллов, подсчитанной по результатам сдачи вступительных испытаний и среднего балла документа об образовании.

По решению приемной комиссии УВО конкурс может проводиться по факультету, группе факультетов, специальности (направлению специальности), группе специальностей (направлений специальностей), специализации, группе специализаций.

По решению Министерства образования при реализации автоматизированного зачисления конкурс может проводиться по группе УВО.

Порядок приема в УО БГУИР

Согласно особенностям приема в БГУИР [5], абитуриент в заявлении может указать специальности только одной формы получения образования.

Конкурс на: - дневную полную форму получения образования за счет средств бюджета (группы 1, 2); - заочную полную форму получения образования за счет средств бюджета (группы 4, 5); - дневную сокращенную форму получения образования за счет средств бюджета и на платной основе (группа 8); - вечернюю и заочную сокращенные формы получения образования на платной основе (группы 9-11)

проводится отдельно по каждой из групп специальностей. Зачисление осуществляется на специальности в рамках каждой группы. При этом абитуриент имеет возможность участвовать в конкурсе на любое число специальностей в рамках одной группы в порядке приоритета, определенного им самим в заявлении при подаче документов.

Абитуриенты ранжируются в рамках выбранной ими группы специальностей на основе общей суммы набранных баллов и зачисляются на специальность в соответствии с указанными в заявлении приоритетами. Абитуриенты, не прошедшие по конкурсу на первую указанную ими специальность из группы, участвуют в конкурсе на следующие специальности из указанного ими приоритетного перечня специальностей этой группы.

С 2015 года абитуриенты, подавшие документы для получения образования за счет средств бюджета и не прошедшие по конкурсу ни на одну из перечисленных в их заявлении специальностей, не подлежат зачислению в БГУИР по выбранной форме получения образования.

Для участия в конкурсе по другой форме получения образования абитуриент может подать новое заявление в приемную комиссию БГУИР в сроки, определенные Министерством образования Республики Беларусь для приема документов на эту форму обучения.

Конкурс для получения образования на платной основе на:
дневную полную форму получения образования (группы 1, 2);
вечернюю полную форму получения образования (группа 3);
на заочную полную форму получения образования (группы 4, 5);
дистанционную полную форму получения образования (группы 6, 7)

проводится в два этапа отдельно для специальностей технико-технологического и экономического профилей для каждой формы получения образования.

На первом этапе осуществляется отбор по каждой форме получения образования и каждой группе специальностей на основе общей суммы баллов, подсчитанной по результатам сдачи вступительных испытаний и среднего балла документа об образовании.

На втором этапе зачисление осуществляется на специальности в рамках каждой группы соответствующей формы получения образования.

При этом абитуриенты имеют возможность участвовать в конкурсе на любое число специальностей в рамках группы. Абитуриенты ранжируются на основе общей суммы набранных баллов и зачисляются на специальности в соответствии с указанными в заявлении приоритетами. Абитуриенты, не прошедшие по конкурсу на первую указанную ими специальность, участвуют в конкурсе на следующую (вторую, и т.д. в пределах списка специальностей группы) специальность из указанного ими в заявлении приоритетного перечня специальностей. Те, кто не прошел по конкурсу ни на одну из перечисленных

в заявлении специальностей, зачисляются на образовавшиеся вакантные места других (родственных) специальностей в рамках профиля по решению приемной комиссии.

В разрабатываемом программном средстве автоматизации деятельности приемной комиссии необходимо учесть все законодательные особенности, связанные с приемными кампаниями в стране.

Обзор аналогов программных средств

Для создания принципиально нового решения в виде программного продукта для решения вопросов автоматизации деятельности приемной комиссии необходимо ознакомиться с существующими аналогами в данной сфере. Анализ достоинств и недостатков этих аналогов позволит сформировать требования к проектируемому программному средству, учитывающие опыт существующих разработок и внести в них улучшения или изменения.

В качестве исследуемых аналогов были выбраны программные продукты, связанные с деятельностью приемных комиссий либо сферой образования, как наиболее близкие по области применения к разрабатываемому программному средству. Помимо этого были проанализированы продукты, связанные с электронной подачей заявлений на оказание различных видов услуг. Источником информации послужили электронные базы в сети Интернет.

В результате поиска были обнаружены ресурсы, представленные в таблице ниже. В ней сведены данные по найденным аналогам и их существенным признакам.

Результаты поиска аналогов программных средств

Наименование ресурса и источник	Признаки выявленных аналогов
1. Система программ «1С:Образование 4.1. Школа 2.0» (http://edu.1c.ru/) [6]	Система программ «1С:Образование 4.1. Школа 2.0» предназначена для организации и поддержки образовательного процесса. Функции системы: а) формирование локальной коллекции ЦОР и организация содержательной работы с ней; б) контроль и самоконтроль учебной деятельности пользователей; в) осуществление импорта/экспорта ЦОР; г) управление списком пользователей (учителей и учащихся), составом и перечнем учебных классов; д) отслеживание состояния работы учащихся в реальном времени.
2. Интернет ресурс «Система электронной подачи заявления» (http://beldor.cent.r.by) [7]	Система обеспечивает выполнение следующих функций: а) электронную подачу заявления на получение специального разрешения посредством электронной формы заявления, размещенной на сайте РУП «Белдорцентр» – услуга предоставляется только зарегистрированным пользователям; б) автоматизированный процесс контроля поступающей информации; в) организацию на сайте для зарегистрированных пользователей электронных «рабочих кабинетов» для обмена данными между пользователем (заявителем) и службами транспортного контроля, в том числе для информирования об этапах рассмотрения заявления, а также для доступа к личным данным о поданных заявлениях, выданных специальных разрешениях и другой информации.
3. Интернет ресурс «Портал государственных и муниципальных услуг Санкт-Петербурга» (http://gu.spb.ru) [8]	Подача электронного заявления с возможностью бронировать время регистрации заключения брака в выбранном органе ЗАГС Санкт-Петербурга (не посещая орган ЗАГС для личной подачи заявления), с возможностью уплаты госпошлины в онлайн-режиме. Для получения государственной услуги необходимо получить "Логин" и "Пароль" в Многофункциональных центрах предоставления государственных услуг.
4. Интернет ресурс приемной комиссии Российского Государственного Геологоразведочного университета (http://mgri-rggru.ru/abitur/) [9]	Электронная подача документов является одним из вариантов подачи документов, реализованная для экономии времени при сдаче в Приемную комиссию комплекта документов поступающим. До подачи документов поступающий имеет возможность после регистрации на сайте Приемной комиссии заполнить электронные формы и отправить информацию вместе с приложенными копиями (сканами) документов на проверку.

В результате поиска были выявлены схожие технические решения, используемые при разработке исследуемых программных продуктов.

Проанализируем отличия разрабатываемого программного средства от выявленных устройств.

1) Система программ «1С:Образование 4.1. Школа 2.0» (пункт 1 таблица 1.1). Данный продукт написан на платформе 1С и широко используется на территории стран СНГ. Является платным. Схожесть продуктов обусловлена сферой применения (система образования), наличием электронного ведения документов обучающихся, возможностью онлайн-отслеживания состояния дел обучающихся.

Программные продукты этой серии обладают следующими достоинствами:

- гибкая настройка системы под конкретные нужды учреждения образования, установившего систему;
- наличие круглосуточной технической поддержки для пользователей, приобретших полную версию продукта;
- полная автоматизация электронного документооборота предприятия;
- выпуск регулярных обновлений для устранения дефектов;
- наличие обновляемой базы данных актуального состояния государственных документов, регулирующих сферу образования и финансовую составляющую работы образовательных учреждений Российской Федерации;
- соответствие современным стандартам защищенности данных пользователя;
- возможность экспорта документов в популярные форматы работы с данными (Word, Excel, PDF, CSV);
- наличие разграничения действий пользователей по ролям, ограничения доступа рядовых пользователей к служебной информации;
- наличие настольного приложения, веб-версии и мобильного приложения;
- отслеживание состояния работы учащихся в режиме реального времени;
- наличие информативного и подробного руководства пользователя.

Выявленные недостатки данного продукта:

- сложен в освоении и доработке, чем отпугивает рядового пользователя;
- высокая стоимость полной версии продукта;
- привязка к законодательству Российской Федерации, что требует дополнительной сложной настройки продукта при использовании его в Республике Беларусь;
- сложность обновления продукта на новую версию из-за особенностей построения БД.

2) Интернет ресурс «Система электронной подачи заявления» (пункт 2 таблица 1.1). Схожими параметрами является упрощение ведения документооборота. Однако механизм электронной подачи заявления притянут за уши, так как пользователю системы все равно необходимо продублировать введенную информацию на бумажном носителе, что в АСПЗиЗ исключается наличием оператора ЭВМ, вносящим информацию в систему.

Достоинства сервиса:

- возможность подачи документов заранее через сеть Интернет;
- наличие справочника с информацией по запрашиваемой услуге.

Недостатки сервиса:

- необходимость дублирования информации на материальных носителях;
- неудобная форма заполнения заявления;
- ручная обработка заявлений оператором.

3) Интернет ресурс «Портал государственных и муниципальных услуг Санкт-Петербурга» (пункт 3 таблица 1.1). Схожесть – автоматизированная обработка документов. Однако данная система используется в другой сфере деятельности и не исключает двойного введения информации в систему, как и ресурс, рассмотренный в п.2.

Достоинства сервиса:

удобная форма заполнения заявления;
возможность отслеживать статус заявки в режиме реального времени;
наличие информационной базы по запрашиваемым услугам;
интуитивно-понятный интерфейс;
разнообразие услуг;
наличие мобильной версии сервиса.

Недостатки сервиса:

узкая привязанность к конкретным организациям;
необходимость дублирования заявления на бумажном носителе.

4) Интернет ресурс приемной комиссии Российского Государственного Геологоразведочного Университета (пункт 4 таблица 1.1). Используется в той же сфере, только на территории РФ. Принципиальное различие в том, что у абитуриентов есть возможность самостоятельно оформить документы, не выходя из дома. Однако данный этап не отменяет личного присутствия абитуриента при подаче заявления.

Достоинства сервиса:

возможность подавать заявление по сети Интернет;
заявление формируется автоматически на основе введенных абитуриентом данных.

Недостатки сервиса:

заявление можно подать только на одну специальность;
подача заявления работает только во время приемной кампании;
привязка сервиса к определенному университету.

В результате поиска аналогов выявлены программные средства, схожие по назначению, однако обладающие существенными функциональными отличиями и представляемыми возможностями. Каждый ресурс используется для целей, отличных от целей разрабатываемого продукта. Аналогов, содержащих автоматизацию зачисления абитуриентов в ВУЗ, не было найдено. Как следствие, патентная чистота разрабатываемого программного продукта является очевидной. Следовательно, разработка программного средства автоматизации приемной кампании обоснована и целесообразна.

Требования к проектируемому программному средству

Назначение разработки

Функциональное назначение программного средства

Функциональным назначением программного средства является предоставление возможности автоматизированной подачи заявления в приемную комиссию ВУЗа, а также управления электронными версиями поданных абитуриентом документов.

Эксплуатационное назначение программного средства

Программное средство может использоваться как для подачи абитуриентом документов в приемную комиссию и отслеживания статуса документов в режиме реального времени, так и для упрощения сотрудникам приемной комиссии процесса приема, обработки и хранения поданных абитуриентами документов.

Конечными пользователями программного средства могут являться как лица, являющиеся абитуриентами ВУЗа (имеющие доступ к ограниченному предоставлению информации), так и лица, являющиеся сотрудниками приемной комиссии университета (имеющие доступ к внутренним сервисам ПС).

Состав выполняемых функций

Программное средство должно обеспечивать возможность выполнения

перечисленных ниже функций:

функции настройки системы под нужды определенного университета;

функциональность автоматизация работы с документами;

функциональность для защищенности данных пользователей;

функции экспорта документов в популярные форматы работы с данными (doc, docx, xls, csv, pdf);

функциональность отслеживания состояния системы в режиме реального времени;

функция подачи заявления в университет через сеть Интернет;

мобильная версия сервиса;

функциональность формирования и печати всей необходимой документации (титульные листы личных дел, расписки, договора, статистика и т.д.);

функция интеграции с базой данных РИКЗ для получения достоверных сведений о сертификатах ЦТ;

функциональность формирования статистики поданных заявлений для публикации на сайте университета;

функциональность генерации списков зачисленных и приказа о зачислении в университет;

функция создания таблицы проходных баллов на специальности;

функциональность разграничения прав доступа к различным сервисам системы посредством ролей пользователей;

функциональность информационной рассылки абитуриентам на электронные ящики, указанные при создании персонального электронного кабинета абитуриента;

сервис сообщений между абитуриентом и приемной комиссией.

Требования к организации входных данных

Входные данные для программного средства должны быть представлены в виде вводимого пользователем с клавиатуры текста: идентификатор; пароль; электронная почта; паспортные данные; данные, необходимые для формирования заявления. После процесса аутентификации клиенту предоставляются возможности работы с системой.

Данные, вводимые пользователем, должны проверяться на корректность, как в процессе аутентификации, так и перед осуществлением каких-либо действий в системе.

Требования к организации выходных данных

В качестве выходных данных будут выступать веб-страницы, отображающие пользовательские данные и результаты их обработки. Выходные данные должны быть представлены в виде сформированных таблиц или документов в популярных форматах работы с данными (doc, docx, xls, csv, pdf).

Требования к временным характеристикам

Требования к временным характеристикам должны зависеть от количества работающих в данный момент времени человек, т.к. программное средство предоставляет совместный доступ к системе.

Требования к надежности

Требования к обеспечению надежного (устойчивого) функционирования программы

Надежное (устойчивое) функционирование программы должно быть обеспечено выполнением совокупности организационно-технических мероприятий, перечень которых приведен ниже:

организацией бесперебойного питания технических средств;

выполнением требований «ГОСТ 31078-2002. Защита информации. Испытания программных средств на наличие компьютерных вирусов»;

необходимым уровнем квалификации сотрудников профильных подразделений.

Время восстановления после отказа

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать времени, необходимого на перезагрузку операционной системы и запуск программы, при условии соблюдения условий эксплуатации технических и программных средств.

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

Отказы из-за некорректных действий оператора

Отказы программы возможны вследствие некорректных действий пользователя при взаимодействии с операционной системой. Во избежание возникновения отказов программы по указанной выше причине следует обеспечить работу конечного пользователя без предоставления ему ограниченного доступа.

Обеспечить максимально безотказную работу программы при любых входных данных, а также при любых действиях пользователя. Программный продукт должен соответствовать требованиям безопасности, установленным ГОСТ 27451-87, ГОСТ 26104-89.

Требования к составу и параметрам технических и программных средств

Требования к техническим средствам

Серверная часть программного средства должна функционировать на ЭВМ со следующими минимальными характеристиками:

процессор Xeon с тактовой частотой 2 ГГц и более;

1 жесткий диска объемом в 100 гб;

оперативная память 4 Гб и более;

сетевая карта Ethernet 1 Гбит.

Клиентская часть программного средства должна функционировать на ЭВМ со следующими минимальными характеристиками:

процессор Intel Pentium 4 с тактовой частотой 2 ГГц и более;

оперативная память 2 Гб и более;

сетевая карта Ethernet 10/100 Мбит.

операционная система: OS Windows, Linux, Mac OS X.

Требования к информационной и программной совместимости

Требования к информационным структурам и методам решения

Требования к информационным структурам на входе и выходе, а также к методам решения не предъявляются.

Требования к исходным кодам и языкам программирования

Исходные коды программы должны быть реализованы на языке C#. В качестве интегрированной среды разработки программы должна быть использована среда Visual Studio 2012.

Требования к программным средствам, используемым программой

Системные программные средства, используемые программой, должны быть представлены локализованной версией операционной системы Windows, Linux или Mac OS X.

Требования к защите информации и программ

В системе должен быть обеспечен надлежащий уровень защиты информации в соответствии с законом о защите персональной информации и программного комплекса в целом от несанкционированного доступа – “Инфор-

мационные технологии. Средства защиты информации от несанкционированного доступа в автоматизированных системах. Общие требования.” по СТБ 34.101.9-2004 от 01.09.2004.

Требование к пользовательскому интерфейсу

Разработать удобный и интуитивно понятный пользовательский интерфейс. Программа будет состоять из серверной части и частей абитуриента, сотрудника приемной комиссии и ответственного секретаря приемной комиссии. Цвет интерфейса: матовый, спокойный цвет.

Требования к патентной чистоте

Проектируемое решение программного продукта должно обладать патентной чистотой.

Обоснование выбора языка

Язык программирования JavaScript

JavaScript – прототипно-ориентированный сценарный язык программирования. Является диалектом языка ECMAScript [10].

JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам.

Основные архитектурные черты: динамическая типизация, слабая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса.

На JavaScript оказали влияние многие языки, при разработке была цель сделать язык похожим на Java, но при этом лёгким для использования непрограммистами. Языком JavaScript не владеет какая-либо компания или организация, что отличает его от ряда языков программирования, используемых в веб-разработке.

Название «JavaScript» является зарегистрированным товарным знаком компании Oracle Corporation.

JavaScript является объектно-ориентированным языком, но используемое в языке прототипирование обуславливает отличия в работе с объектами по сравнению с традиционными класс-ориентированными языками. Кроме того, JavaScript имеет ряд свойств, присущих функциональным языкам – функции как объекты первого класса, объекты как списки, карринг, анонимные функции, замыкания – что придаёт языку дополнительную гибкость.

Несмотря на схожий с Си синтаксис, JavaScript по сравнению с языком Си имеет коренные отличия:

- объекты, с возможностью интроспекции;
- функции как объекты первого класса;
- автоматическое приведение типов;
- автоматическая сборка мусора;
- анонимные функции.

В языке отсутствуют такие полезные вещи, как:

- модульная система: JavaScript не предоставляет возможности управлять зависимостями и изоляцией областей видимости;
- стандартная библиотека: в частности, отсутствует интерфейс программирования приложений по работе с файловой системой, управлению потоками ввода-вывода, базовых типов для бинарных данных;
- стандартные интерфейсы к веб-серверам и базам данных;
- система управления пакетами, которая бы отслеживала зависимости и автоматически устанавливала их.

Данный язык является основополагающим в реализации данного про-

граммного продукта, реализованного в рамках этого дипломного проекта.

В качестве формата передачи данных могут использоваться фрагменты простого текста, HTML-кода, JSON или XML.

Формат данных JSON

JSON (англ. JavaScript Object Notation) – текстовый формат обмена данными, основанный на JavaScript и обычно используемый именно с этим языком. Как и многие другие текстовые форматы, JSON легко читается людьми [11].

Несмотря на происхождение от JavaScript (точнее, от подмножества языка стандарта ECMA-262 1999 года), формат считается языконезависимым и может использоваться практически с любым языком программирования. Для многих языков существует готовый код для создания и обработки данных в формате JSON.

За счёт своей лаконичности по сравнению с XML, формат JSON может быть более подходящим для сериализации сложных структур. Если говорить о веб-приложениях, в таком ключе он уместен в задачах обмена данными как между браузером и сервером (AJAX), так и между самими серверами (программные HTTP-интерфейсы).

Поскольку формат JSON является подмножеством синтаксиса языка JavaScript, то он может быть быстро десериализован встроенной функцией `eval()`. Кроме того, возможна вставка вполне работоспособных JavaScript-функций. В языке PHP, начиная с версии 5.2.0, поддержка JSON включена в ядро в виде функций `json_decode()` и `json_encode()`, которые сами преобразуют типы данных JSON в соответствующие типы PHP и наоборот.

JSON-текст представляет собой (в закодированном виде) одну из двух структур:

Набор пар ключ:значение. В различных языках это реализовано как объект, запись, структура, словарь, хэш-таблица, список с ключом или ассоциативный массив. Ключом может быть только строка, значением – любая форма. Упорядоченный набор значений. Во многих языках это реализовано как массив, вектор, список или последовательность.

Это универсальные структуры данных: как правило, любой современный язык программирования поддерживает их в той или иной форме. Они легли в основу JSON, так как он используется для обмена данными между различными языками программирования.

В качестве значений в JSON используются структуры:

объект – это неупорядоченное множество пар ключ:значение, заключённое в фигурные скобки «{ }». Ключ описывается строкой, между ним и значением стоит символ «:». Пары ключ:значение отделяются друг от друга запятыми; массив (одномерный) – это упорядоченное множество значений. Массив заключается в квадратные скобки «[]». Значения разделяются запятыми. Значение может быть строкой в двойных кавычках, числом, объектом, массивом, одним из литералов: `true`, `false` или `null`. Таким образом, структуры могут быть вложены друг в друга.

строка – это упорядоченное множество из нуля или более символов юникода, заключённое в двойные кавычки. Символы могут быть указаны с использованием escape-последовательностей, начинающихся с обратной косой черты «\». Строка очень похожа на одноимённый тип данных в языках C и Java.

Число тоже очень похоже на C- или Java-число, за исключением того, что используется только десятичный формат. Пробелы могут быть вставлены между любыми двумя синтаксическими элементами.

Данный способ используется полностью во всех механизмах передачи

данных между клиентским приложением и сервером.

Язык программирования C#

C# – объектно-ориентированный язык программирования, который относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML [12].

Переняв многое от своих предшественников – языков C++, Pascal, Модула, Smalltalk и, в особенности, Java – C#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем, например, C# в отличие от C++ не поддерживает множественное наследование классов (между тем допускается множественное наследование интерфейсов).

C# разрабатывался как язык программирования прикладного уровня для CLR и, как таковой, зависит, прежде всего, от возможностей самой CLR. Это касается, прежде всего, системы типов C#, которая отражает BCL. Присутствие или отсутствие тех или иных выразительных особенностей языка диктуется тем, может ли конкретная языковая особенность быть транслирована в соответствующие конструкции CLR. CLR предоставляет C#, как и всем другим .NET-ориентированным языкам, многие возможности, которых лишены «классические» языки программирования.

На данном языке полностью написан сервер разработанного программного средства.

Для создания данного программного продукта понадобились знания в таких базовых технологиях как HTML (Hyper Text Markup Language), а так же CSS (Каскадные таблицы стилей). Данные технологии являют собой основу web-разработки.

Кэширование используется для ускорения вызовов и обращения к страницам сайта при помощи сохранения результатов их работы в файл.

Для обновления данных в реальном времени будет использоваться технология Ajax.

МОДЕЛИРОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ И РАЗРАБОТКА ФУНКЦИОНАЛЬНЫХ ТРЕБОВАНИЙ

Описание функциональности ПС

Для представления функциональной модели была выбрана диаграмма вариантов использования UML [13], которая отражает отношения между актерами и прецедентами и позволяет описать систему на концептуальном уровне. Прецедент соответствует отдельному сервису системы, определяет один из вариантов её использования и описывает типичный способ взаимодействия пользователя с системой. UML предназначен для определения, визуализации, проектирования и документирования программных систем.

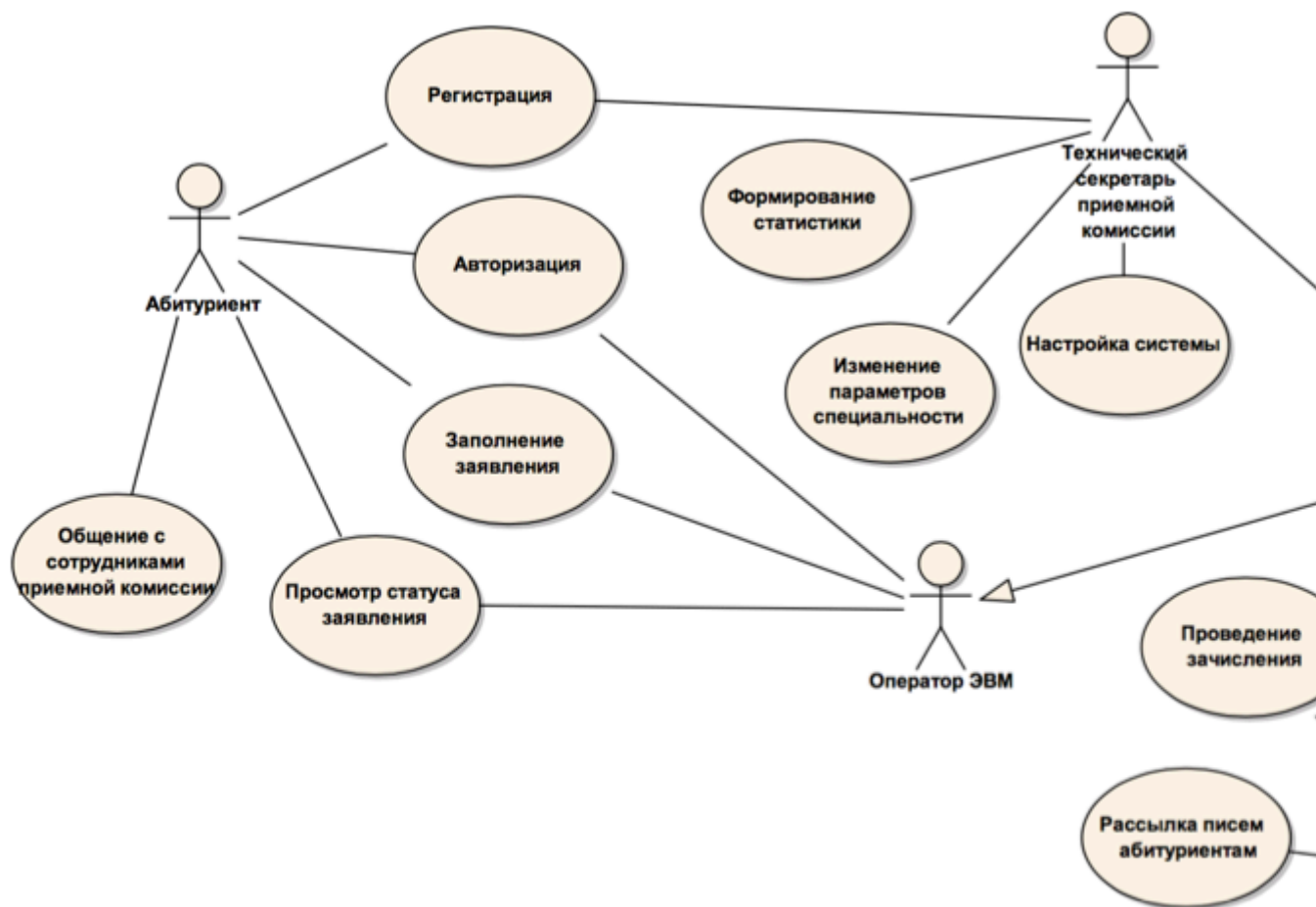


Диаграмма вариантов использования

Диаграмма вариантов использования разрабатываемого программного средства представлена на рисунке 2.1. На диаграмме можно выделить два основных составляющих элемента – актер и прецедент. Актер – стилизованный человек, обозначающий набор ролей пользователя, взаимодействующего с некоторой сущностью. Прецедент – эллипс с надписью, обозначающий выполняемые системой действия, приводящие к наблюдаемым актером результатам.

На основании представленной диаграммы вариантов использования можно сделать вывод, что в системе будет существовать пять основных актеров: абитуриент, оператор ЭВМ, технический секретарь приемной комиссии,

сотрудник приемной комиссии, ответственный секретарь приемной комиссии. Рассмотрим каждый из прецедентов более подробно для каждого актера.

Абитуриент

Абитуриенту предоставляются следующие возможности:

регистрация: новый пользователь, желающий пользоваться программным средством, вводит свои данные в соответствующую регистрационную форму (уникальный идентификатор; пароль; электронную почту), после чего заполненную форму отправляет на сервер;

авторизация: пользователь, являющийся полноценным пользователем программного средства, вводит собственный уникальный идентификатор и пароль, указанный при регистрации, в форму, после чего отправляет эти данные на удаленный сервер, который проверяет их и предоставляет доступ к системе;

заполнение заявления: у авторизованного абитуриента есть возможность заполнить заявление необходимыми данными с проверкой системой введенных данных и отправки запроса на сервер;

просмотр статуса заявления: во время работы с системой абитуриент может просмотреть статус своего заявления, который может принимать значения «Заполнено полностью», «Заполнено не полностью», «Подано в университет», «Зачислен в университет», «Не зачислен», «Забрано»;

общение с сотрудниками приемной комиссии: абитуриент имеет возможность задать вопрос через сервис «Сообщения» и отслеживать статус темы с сообщениями в режиме реального времени.

Оператор ЭВМ

Он является промежуточным звеном для таких ролей, как Технический секретарь приемной комиссии, Сотрудник приемной комиссии, Ответственный секретарь приемной комиссии. У всех данных ролей общим является авторизация, заполнение заявления, просмотр статуса заявления.

Сотрудник приемной комиссии

Сотрудник приемной комиссии – унаследованная роль от Оператора ЭВМ, обладает всеми его функциями и дополнительно предоставляются возможности:

ответы на вопросы абитуриентов: в сервисе «Сообщения» сотрудник приемной комиссии может дать ответ на вопрос абитуриента либо перенаправить вопрос ответственному секретарю приемной комиссии, соответствующе пометив сообщение;

печать документов: на основании введенной в систему информации сотрудник может сформировать и распечатать ряд документов (расписку, титульный лист личного дела и т.д.);

изменение данных абитуриента: при проверке документов и нахождении несоответствия сотрудник может изменять данные об абитуриенте в системе.

Ответственный секретарь приемной комиссии

Ответственный секретарь приемной комиссии – унаследованная роль от Сотрудника приемной комиссии, обладает всеми его функциями, а также: рассылка писем абитуриентам: при необходимости ответственный секретарь может создавать рассылку на электронные адреса абитуриентов определенной выборки с информационными сообщениями о ходе приемной кампании; проведение зачисления: ответственный секретарь может запустить процесс зачисления внутри системы, что автоматически изменит статусы заявлений всех абитуриентов, а также сформирует списки зачисленных и таблицы проходных баллов.

Технический секретарь приемной комиссии

Технический секретарь приемной комиссии – унаследованная роль от Сотрудника приемной комиссии, обладает всеми его функциями, а также:

- формирование статистики: технический секретарь может генерировать и публиковать статистику на сайте университета, на бумажном носителе, а также в сервисе «Статистика» системы;
- настройка системы: технический секретарь обладает возможностью настраивать параметры системы, управлять правами пользователей;
- регистрация: только технический секретарь может назначать пользователям права сотрудников приемной комиссии и выше;
- изменение параметров специальности: технический секретарь имеет доступ к настройке данных о специальностях, вроде плана набора, названия специальности, формы и вида обучения, факультета и т.д.

Спецификация функциональных требований

На основании анализа исходных данных для проектируемого программного средства можем выделить, что основной целью является создание качественного программного продукта, позволяющего решить существующие проблемы таких как:

- автоматизация процесса подачи заявления и зачисления в университет для абитуриента;
- автоматизация процесса принятия и обработки документов для сотрудников приемной комиссии;
- предоставление абитуриентам сервисов для общения с приемной комиссией в режиме реального времени, а также для дистанционного отслеживания статуса своего заявления.

В ходе разработки будут реализованы следующие возможности:

- функции настройки системы под нужды определенного университета;
- функциональность автоматизация работы с документами;
- функциональность для защищенности данных пользователей;
- функции экспорта документов в популярные форматы работы с данными (Word, Excel, PDF);
- функциональность отслеживания состояния системы в режиме реального времени;
- функция подачи заявления в университет через сеть Интернет;
- мобильная версия сервиса;
- функциональность формирования и печати всей необходимой документации (титульные листы личных дел, расписки, договора, статистика и т.д.);
- функция интеграции с базой данных РИКЗ для получения достоверных сведений о сертификатах ЦТ;
- функциональность формирования статистики поданных заявлений для публикации на сайте университета;
- функциональность генерации списков зачисленных и приказа о зачислении в университет;
- функция создания таблицы проходных баллов на специальности;
- функциональность разграничения прав доступа к различным сервисам системы посредством ролей пользователей;
- функциональность информационной рассылки абитуриентам на электронные ящики, указанные при создании личного электронного кабинета абитуриента;
- сервис сообщений между абитуриентом и приемной комиссией.

Проект представляет собой онлайн систему формирования заявления и

проведения зачисления.

Доступ к системе будет предоставляться пяти категориям пользователей.

Абитуриент. Предоставляется возможность создания персонального электронного кабинета, заполнения заявления, редактирования его в любое время, общения с представителями приемной комиссии через сервис «Сообщения».

Оператор ЭВМ. Может создавать электронный кабинет для абитуриента, а также вносить дополнительные необходимые для приема документов данные.

Технический секретарь приемной комиссии. Имеет доступ к настройке системы, управляет правами пользователей системы, генерирует и публикует статистику поданных заявлений на сайте университета и в сервисе «Статистика».

Сотрудник приемной комиссии. Может отвечать на вопросы абитуриентов в сервисе «Сообщения», печатать необходимые документы на основе введенных в базу данных об абитуриенте, изменять данные абитуриентов по мере необходимости.

Ответственный секретарь приемной комиссии. Имеет право проводить зачисление в университет и рассылать письма на электронные ящики абитуриентов с информацией о ходе приемной кампании.

Для каждого вида пользователя предоставляется отдельный вход в свой кабинет. Для каждой роли предусмотрено наличие своего набора функций, который могут дублироваться для разных видов ролей.

Основные функции данного программного средства:

функции регистрации и аутентификации пользователя:

авторизация и аутентификация. Пользователь имеет доступ для входа в систему посредством ввода логина и пароля;

выход из системы. Для окончания работы пользователь выходит из системы.

После успешного выхода система возвращает пользователя на страницу авторизации;

настройка аккаунта. Интерфейс, позволяющий пользователям изменить допустимые настройки своего аккаунта. Пользователь получает доступ для изменения данных своего аккаунта при нажатии иконки «Настройки» в правом верхнем углу страницы;

функции настройки системы под нужды определенного университета:

настройка информации об университете;

настройка плана набора в университет;

настройка набора факультетов и специальностей университета;

изменение модуля проведения зачисления в университет в соответствии с внутренним порядком приема документов;

функциональность автоматизация работы с документами:

формирование и печать расписок;

формирование и печать заявлений абитуриентов;

формирование и печать договоров между студентами и университетом;

формирование и печать титульных листов личных дел абитуриентов;

формирование и печать справок в военно-учетный стол лицам мужского пола, зачисленным в вуз;

формирование и печать конвертов для извещений о зачислении абитуриентов в вуз;

функции экспорта документов в популярные форматы работы с данными (Word, Excel, PDF):

экспорт статистики поданных заявлений;

экспорт заявлений;

экспорт списка зачисленных (по специальностям, группам специальностей, вузу);
экспорт проходных баллов;
функциональность отслеживания состояния системы в режиме реального времени;
функция подачи заявления в университет через сеть Интернет;
оформление заявления в сети Интернет;
мобильная версия сервиса:
приложения для основных мобильных операционных систем (iOS, Android, Windows Phone);
мобильная веб-версия сайта системы для популярных браузеров (Safari, Chrome, Firefox, Internet Explorer, Opera);
функция интеграции с базой данных РИКЗ для получения достоверных сведений о сертификатах ЦТ;
получение данных абитуриента на основании введенных паспортных данных;
проверка введенных баллов для сертификатов;
функциональность формирования статистики поданных заявлений для публикации на сайте университета:
генерация статистики;
публикация на сайте университета;
печать бумажной версии статистики;
функциональность генерации списков зачисленных и приказа о зачислении в университет;
функциональность информационной рассылки абитуриентам на электронные ящики, указанные при создании личного электронного кабинета абитуриента:
выбор целевой аудитории рассылки;
создание и редактирования текста сообщения;
отправка писем;
отслеживания статуса рассылки сообщений;
сервис сообщений между абитуриентом и приемной комиссией;
просмотр тем сообщений;
создание ответов на сообщения абитуриентов;
закрытие или удаление темы.

ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА

Разработка программной архитектуры

Прежде чем приступить к непосредственной реализации программного средства, необходимо определиться с архитектурой коллективной обработки расписания, а также компонентов, на основе которых будет построено конечное приложение.

В первую очередь, необходимо провести анализ необходимой аппаратной конфигурации, на которой будут работать части конечного программного средства, и описать их взаимодействие между собой. Для описания узлов и их связей будем использовать диаграмму развертывания (рисунок 3.1).

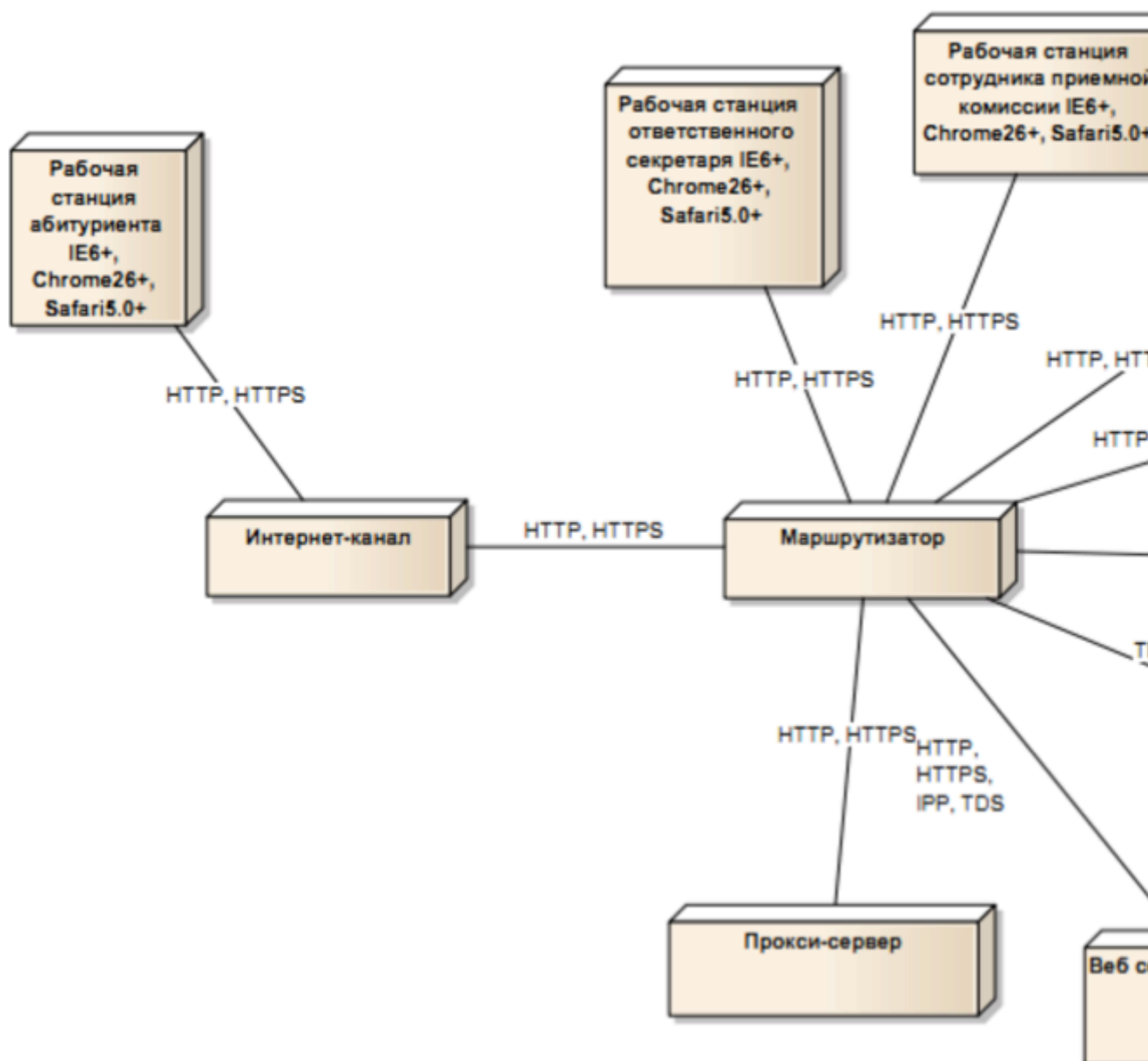


Диаграмма развертывания

На основе вышеизображенной диаграммы можно сделать следующие выводы:

узлы могут располагаться в различных частях мира и взаимодействовать между собой через сеть Интернет;

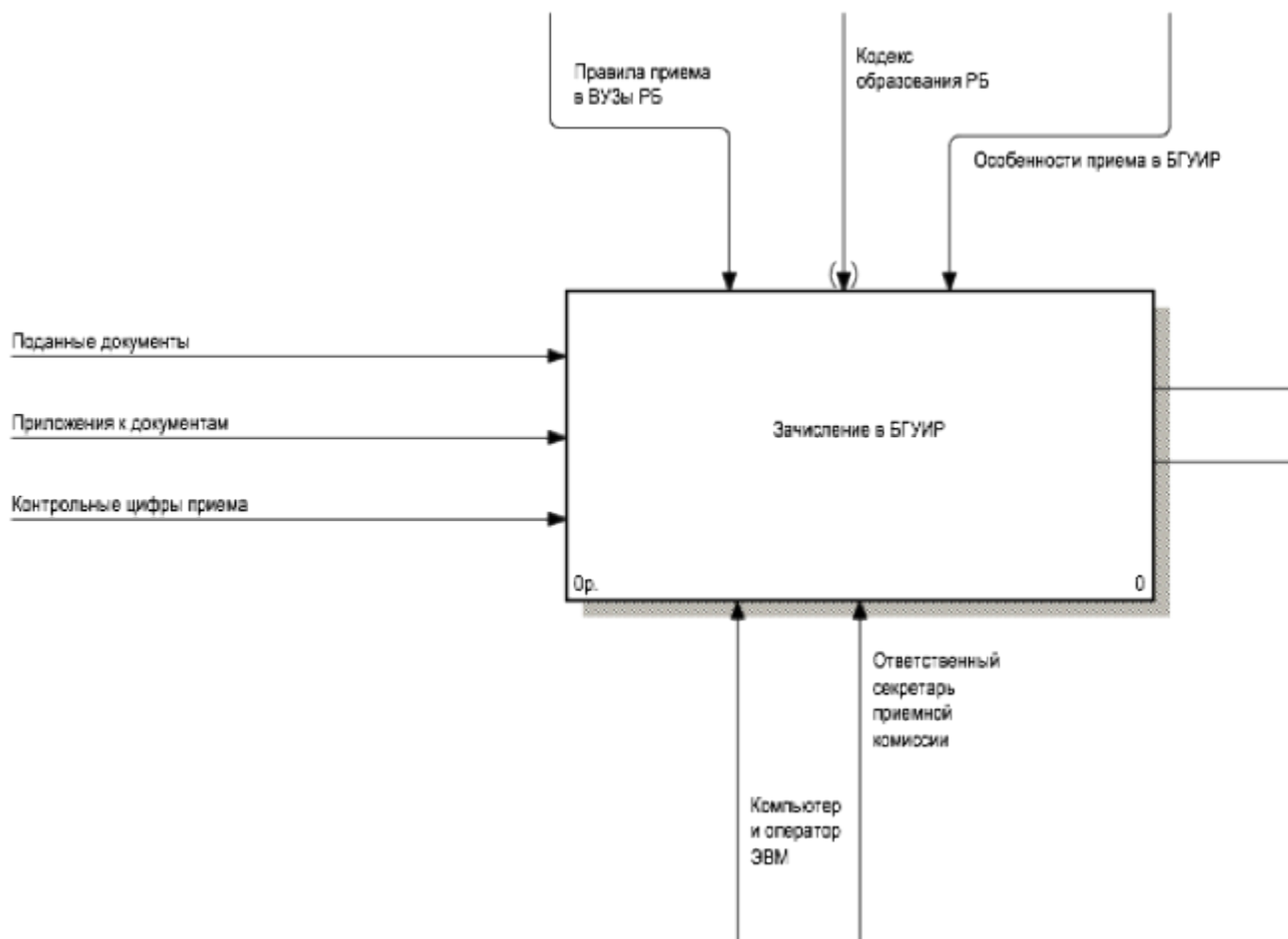
сервер базы данных поддерживаются в рабочем состоянии отдельно от основного сервера;

клиент, осуществляющий работу с системой с помощью HTTP, HTTPS;

HTTPS протокол применяется как клиентами, так и всевозможными серверами, с целью осуществления обмена запросами и ответами на них.

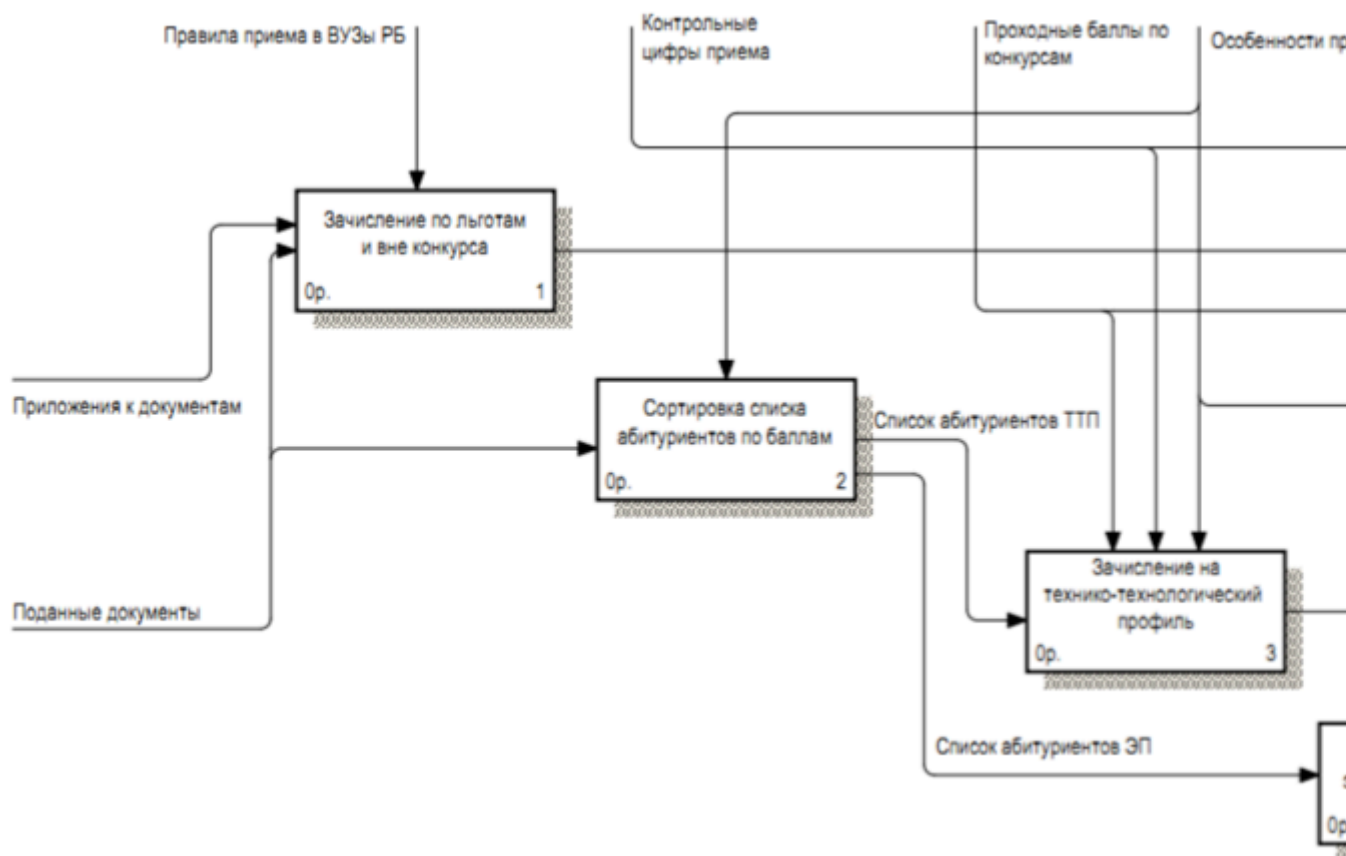
Проектирование взаимодействия модулей системы

Данная модель выполнена по методологии IDEF0 (рисунок 3.2). Она описывает взаимодействия входных, выходных данных, а также механизм и управление отдельными модулями и взаимодействия между ними. Рассматривается с точки зрения ответственного секретаря приемной комиссии.



Модель системы

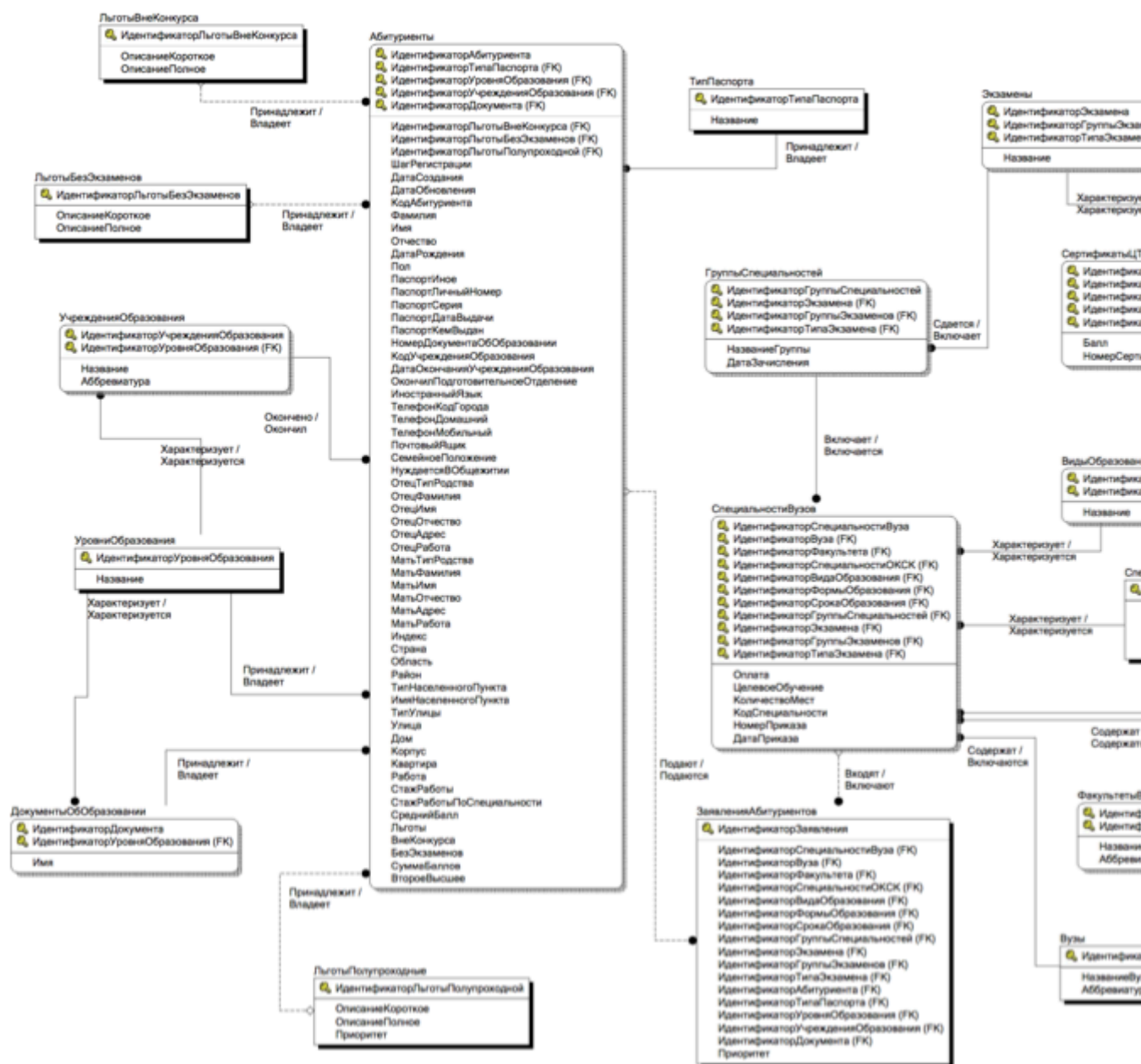
На рисунке 3.3 изображена декомпозиция контекстной диаграммы и процесс проведения зачисления на группы специальностей в соответствии с Порядком приема в БГУИР в 2015 году.



Модель проведения зачисления на группы специальностей

Разработка модели базы данных

Неотъемлемой частью конечного программного средства является база данных, используемая системой в процессе работы. Информационную модель предметной области можно представить на языке IDEF1X. Модель базы данных представлена на рисунке 3.4, а также на листе формата А1 (см. Графическое приложение).



Информационная модель предметной области

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

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

Сущность «Абитуриенты»

Представляет данные об абитуриенте следующими полями:

ИдентификаторАбитуриента – идентификатор абитуриента (первичный ключ);
ИдентификаторТипаПаспорта – внешний ключ на сущность «ТипПаспорта»;
ИдентификаторУровняОбразования – внешний ключ на сущность «Уровни-Образования»;
ИдентификаторУчрежденияОбразования – внешний ключ на сущность «УчрежденияОбразования»;
ИдентификаторДокумента – внешний ключ на сущность «ДокументыОбОбразовании»;
ИдентификаторЛьготыВнеКонкурса – внешний ключ на сущность «ЛьготыВнеКонкурса»;
ИдентификаторЛьготыБезЭкзаменов – внешний ключ на сущность «ЛьготыБезЭкзаменов»;
ИдентификаторЛьготыПолупроходной – внешний ключ на сущность «ЛьготыПолупроходной»;
ШагРегистрации – этап принятия документов абитуриента;
ДатаСоздания – дата создания записи об абитуриенте;
ДатаОбновления – дата обновления записи об абитуриенте;
КодАбитуриента – номер личного дела абитуриента;
Фамилия – фамилия абитуриента;
Имя – имя абитуриента;
Отчество – отчество абитуриента;
ДатаРождения – дата рождения абитуриента;
Пол – пол абитуриента;
ПаспортИное – нестандартный тип документа абитуриента;
ПаспортЛичныйНомер – личный номер документа абитуриента;
ПаспортСерия – серия документа абитуриента;
ПаспортНомер – номер документа;
ПаспортДатаВыдачи – дата выдачи документа;
ПаспортКемВыдан – орган, выдавший документ;
НомерДокументаОбОбразовании – идентификатор документа об образовании абитуриента;
КодУчрежденияОбразования – код учреждения образования, которое абитуриент закончил в год подачи документов;
ДатаОкончанияУчрежденияОбразования – дата окончания учреждения образования;
ОкончилПодготовительноеОтделение – признак прохождения абитуриентом подготовительного отделения вуза;
ИностранныйЯзык – иностранный язык, который абитуриент изучал в общеобразовательной школе;
ТелефонКодГорода – код города домашнего телефона абитуриента;
ТелефонДомашний – номер домашнего телефона абитуриента;
ТелефонМобильный – номер мобильного телефона абитуриента;
ПочтовыйЯщик – электронный адрес абитуриента;
СемейноеПоложение – семейное положение абитуриента;
НуждаетсяВОбщежитии – признак нужды абитуриента в общежитии при поступлении в вуз;
ОтецТипРодства – тип родства отца (опекуна) абитуриента с абитуриентом;
ОтецФамилия – фамилия отца (опекуна);
ОтецИмя – имя отца (опекуна);
ОтецОтчество – отчество отца (опекуна);
ОтецАдрес – адрес отца (опекуна);

ОтецРабота – место работы отца (опекуна);
МатьТипРодства – тип родства матери (опекунши) абитуриента с абитуриентом;
МатьФамилия – фамилия матери (опекунши);
МатьИмя – имя матери (опекунши);
МатьОтчество – отчество матери (опекунши);
МатьАдрес – адрес матери (опекунши);
МатьРабота – место работы матери (опекунши);
Индекс – домашний индекс;
Страна – страна проживания абитуриента;
Область – области проживания;
Район – район проживания;
ТипНаселенногоПункта – тип населенного пункта;
ИмяНаселенногоПункта – название населенного пункта;
ТипУлицы – тип улицы;
Улица – название улицы;
Дом – номер дома;
Корпус – номер корпуса;
Квартира – номер квартиры;
Работа – место работы абитуриента;
СтажРаботы – общий стаж работы абитуриента;
СтажРаботыПоСпециальности – стаж работы абитуриента по специальности, на которую подаются документы;
СреднийБалл – средний балл документа об образовании;
Льготы – льготы, указанные абитуриентом в Электронном кабинете;
ВнеКонкурса – признак зачисления абитуриента вне конкурса;
БезЭкзаменов – признак зачисления абитуриента без экзаменов;
СуммаБаллов – общая сумма баллов абитуриента;
ВтороеВысшее – признак получения абитуриентом второго высшего образования.

Сущность имеет идентифицирующие связи «многие к одному» с сущностями «ТипПаспорта», «Уровни образования», «Учреждения образования», «ДокументыОбОбразовании», а также не идентифицирующие связи «многие к одному или нулю» с сущностями «ЛьготыВнеКонкурса», «ЛьготыБезЭкзаменов» и «ЛьготыПолупроходные». Также сущность имеет не идентифицирующую связь «ноль или один ко многим» с сущностью «ЗаявленияАбитуриентов».

Сущность «ВидыОбразования»

Представляет виды образования со следующими полями:

ИдентификаторВидаОбразования – идентификатор вида образования (первичный ключ);
ИдентификаторФормыОбразования – внешний ключ на сущность «ФормыОбразования»;
Название – название вида образования.

Сущность имеет связь «многие к одному» с сущностью «ФормыОбразования» и связь «один ко многим» с сущностью «СпециальностиВузов».

Сущность «Вузы»

Представляет описания вузов со следующими полями:

ИдентификаторВуза – идентификатор вуза (первичный ключ);
НазваниеВуза – наименование вуза;
АббревиатураВуза – аббревиатура вуза.

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

Вузов».

Сущность «ГруппыСпециальностей»

Представляет группы специальностей со следующими полями:

ИдентификаторГруппыСпециальностей – идентификатор группы специальностей (первичный ключ);

ИдентификаторГруппыЭкзаменов – внешний ключ на сущность «ГруппыЭкзаменов»;

НазваниеГруппы – название группы специальностей;

ДатаЗачисления – дата проведения зачисления на группу специальностей.

Сущность имеет связь «один ко многим» с сущностью «СпециальностиВузов» и связь «многие ко многим» с сущностью «Экзамены».

Сущность «ДокументыОбОбразовании»

Представляет документы о получении образования со следующими полями:

ИдентификаторДокумента – идентификатор документа об образовании (первичный ключ);

ИдентификаторУровняОбразования – внешний ключ на сущность «УровниОбразования»;

Имя – название документа об образовании.

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

Сущность «ЗаявленияАбитуриентов»

Представляет заявления абитуриентов со следующими полями:

ИдентификаторЗаявления – идентификатор заявления абитуриента (первичный ключ);

ИдентификаторСпециальностиВуза – внешний ключ на сущность «СпециальностиВузов»;

ИдентификаторАбитуриента – внешний ключ на сущность «Абитуриенты»;

Приоритет – порядковый номер указания выбранной специальности абитуриентом в заявлении.

Сущность имеет связь «многие к одному» с сущностями «Абитуриенты» и «СпециальностиВузов».

Сущность «ЛьготыБезЭкзаменов»

Представляет описания видов льгот без экзаменов со следующими полями:

ИдентификаторЛьготыБезЭкзаменов – идентификатор льготы (первичный ключ);

ОписаниеКороткое – короткое описание льготы;

ОписаниеПолное – полное описание льготы.

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

Сущность «ЛьготыВнеКонкурса»

Представляет описания видов льгот вне конкурса со следующими полями:

ИдентификаторЛьготыВнеКонкурса – идентификатор льготы (первичный ключ);

ОписаниеКороткое – короткое описание льготы;

ОписаниеПолное – полное описание льготы.

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

Сущность «ЛьготыПолупроходные»

Представляет описания видов полупроходных льгот со следующими полями:

ИдентификаторЛьготыПолупроходной – идентификатор льготы (первичный ключ);

ОписаниеКороткое – короткое описание льготы;

ОписаниеПолное – полное описание льготы;

Приоритет – приоритет льготы при полупроходном балле.

Сущность имеет связь «многие ко многим» с сущностью «Абитуриенты».

Сущность «СертификатыЦТ»

Представляет сертификаты централизованного тестирования со следующими полями:

ИдентификаторСертификатаЦТ – идентификатор сертификата ЦТ (первичный ключ);

ИдентификаторУчастникаЦТ – внешний ключ на сущность «УчастникиЦТ»;

ИдентификаторЭкзамена – внешний ключ на сущность «Экзамены»;

Балл – балл сертификата;

НомерСертификата – номер сертификата.

Сущность имеет связь «многие к одному» с сущностью «УчастникиЦТ».

Сущность «СпециальностиВузов»

Представляет специальности вузов со следующими полями:

ИдентификаторСпециальностиВуза – идентификатор специальности (первичный ключ);

ИдентификаторВуза – внешний ключ на сущность «Вузы»;

ИдентификаторФакультета – внешний ключ на сущность «ФакультетыВузов»;

ИдентификаторСпециальностиОКСК – внешний ключ на сущность «СпециальностиОКСК»;

ИдентификаторВидаОбразования – внешний ключ на сущность «ВидыОбразования»;

ИдентификаторСрокаОбразования – внешний ключ на сущность «СрокиОбразования»;

ИдентификаторГруппыСпециальностей – внешний ключ на сущность «ГруппыСпециальностей»;

Оплата – признак платной формы обучения;

ЦелевоеОбучение – признак целевого получения образования;

КоличествоМест – количество мест на специальность;

КодСпециальности – код специальности;

НомерПриказа – номер приказа о зачислении на специальность;

ДатаПриказа – дата приказа о зачислении на специальность.

Сущность имеет связь «один ко многим» с сущностями «ГруппыСпециальностей», «ФакультетыВузов», «Вузы», «ВидыОбразования», «СрокиОбразования», а также связь «многие к одному» с сущностями «ЗаявленияАбитуриентов».

Сущность «СпециальностиОКСК»

Представляет специальности в общем классификаторе специальностей и квалификаций со следующими полями:

ИдентификаторСпециальностиОКСК – идентификатор специальности (первичный ключ);

ИмяСпециальности – наименование специальности;

КодСпециальности – код специальности;

КвалификацияСпециальности – квалификация специальности;

АббревиатураСпециальности – аббревиатура специальности.

Сущность имеет связь «многие к одному» с сущностью «СпециальностиВузов».

Сущность «СрокиОбразования»

Представляет сроки получения образования со следующими полями:
ИдентификаторСрокаОбразования – идентификатор срока получения образования (первичный ключ);

Название – название срока получения образования.

Сущность имеет связь «один ко многим» с сущностью «Специальности-Вузов».

Сущность «ТипПаспорта»

Представляет типы документов, удостоверяющих личность, со следующими полями:

ИдентификаторТипаПаспорта – идентификатор типа документа (первичный ключ);

Название – название типа документа.

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

Сущность «ТипыЭкзаменов»

Представляет типы вступительных испытаний со следующими полями:

ИдентификаторТипаЭкзамена – идентификатор типа экзамена (первичный ключ);

НазваниеТипа – название типа;

КороткоеНазваниеТипа – короткое название типа экзамена.

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

Сущность «УровниОбразования»

Представляет уровни получения образования со следующими полями:

ИдентификаторУровняОбразования – идентификатор уровня (первичный ключ);

Название – наименование уровня получения образования.

Сущность имеет связь «один ко многим» с сущностями «УчрежденияОбразования», «ДокументыОбОбразовании», «Абитуриенты».

Сущность «УчастникиЦТ»

Представляет участников централизованного тестирования со следующими полями:

ИдентификаторУчастникаЦТ – идентификатор участника ЦТ (первичный ключ);

Фамилия – фамилия участника;

Имя – имя участника;

Отчество – отчество участника;

СерияПаспорта – серия паспорта участника ЦТ;

НомерПаспорта – номер паспорта участника ЦТ.

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

Сущность «УчрежденияОбразования»

Представляет виды учреждений образования со следующими полями:

ИдентификаторУчрежденияОбразования – идентификатор вида учреждения образования (первичный ключ);

ИдентификаторУровняОбразования – внешний ключ на сущность «УровниОбразования»;

Название – наименование учреждения образования;

Аббревиатура – аббревиатура учреждения образования.

Сущность имеет связь «один ко многим» с сущностью «Абитуриенты» и связь «многие к одному» с сущностью «УровниОбразования».

Сущность «ФакультетыВузов»

Представляет факультеты вузов со следующими полями:

ИдентификаторФакультета – идентификатор факультета (первичный ключ);
ИдентификаторВуза – внешний ключ на сущность «Вузы»;
Название – наименование факультета;
Аббревиатура – аббревиатура факультета.

Сущность имеет связь «один ко многим» с сущностью «Специальности-Вузов» и связь «многие к одному» с сущностью «Вузы».

Сущность «ФормыОбразования»

Представляет формы получения образования со следующими полями:
ИдентификаторФормыОбразования – идентификатор формы получения образования (первичный ключ);
Название – наименование формы получения образования.

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

Сущность «Экзамены»

Представляет вступительные испытания со следующими полями:
ИдентификаторЭкзамена – идентификатор экзамена (первичный ключ);
ИдентификаторГруппыЭкзаменов – идентификатор группы вступительных испытаний (вторичный ключ);
ИдентификаторТипаЭкзамена – внешний ключ на сущность «ТипыЭкзаменов»;
Название – наименование вступительного испытания.

Сущность имеет связь «один ко многим» с сущностями «СертификатыЦТ» и «ГруппыСпециальностей» и связь «многие к одному» с сущностью «ТипыЭкзаменов».

Разработка схемы алгоритма работы с программой

Схема алгоритма выполнена на листе формата А1 (см. Графическое приложение), а также на рисунке 3.5. Данная схема отображает алгоритм работы с сайтом. Браузер пытается подключиться и загрузить страницу. Далее пользователь имеет ряд путей работы: переходы по ссылкам внутри сайта и использование его разделов. Если пользователь закрывает вкладку или переходит по внешней ссылке, то работа с сайтом закончена.

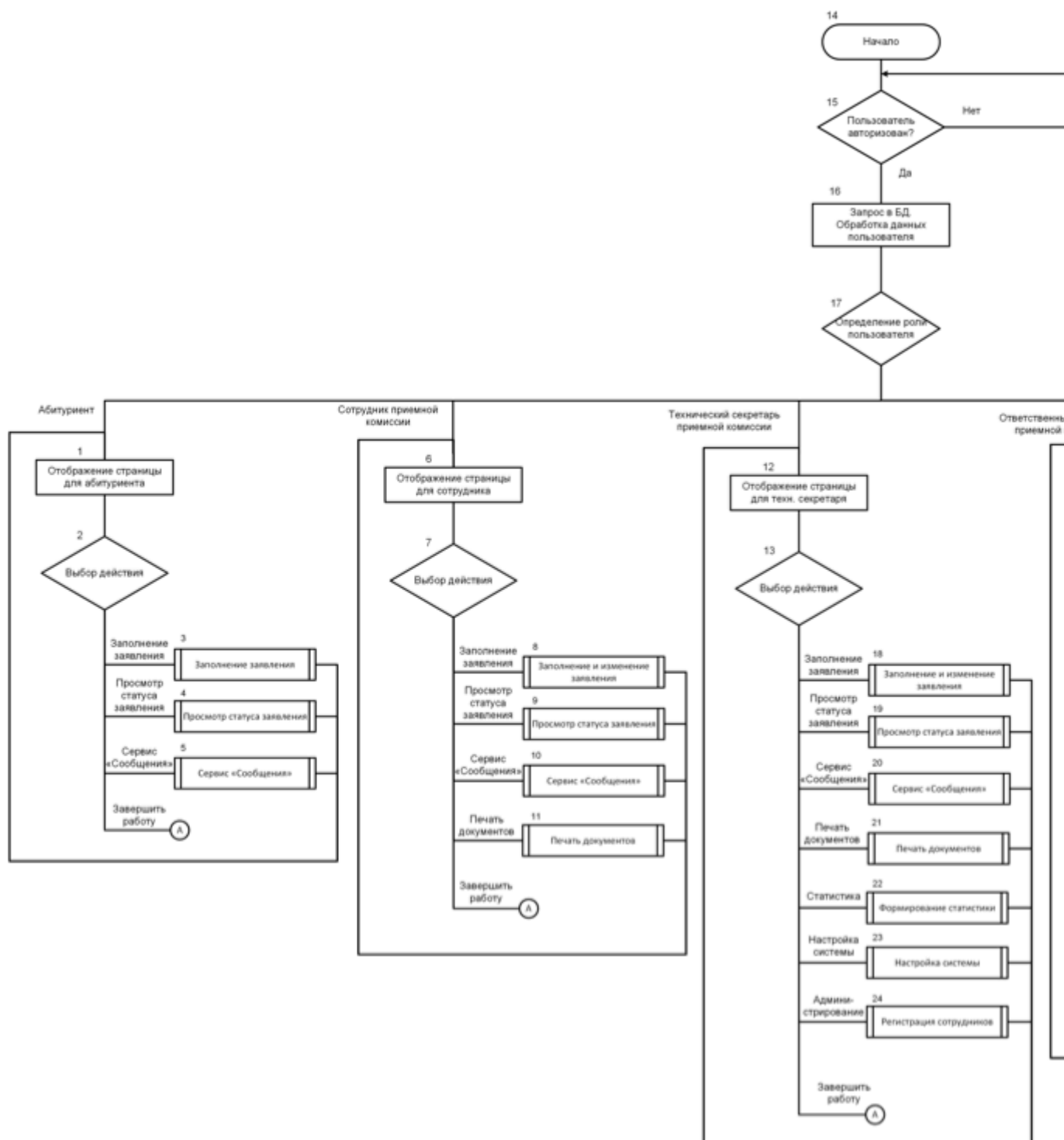


Схема алгоритма работы с программой

Разработка алгоритма входа пользователя в систему

Вход пользователя в систему состоит из аутентификации и авторизации. Аутентификация представляет собой поиск соответствия пользователя названному им идентификатору. Авторизация – предоставление этому пользователю возможностей в соответствии с положенными ему правами. Алгоритм аутентификации представлен на рисунке 3.6.

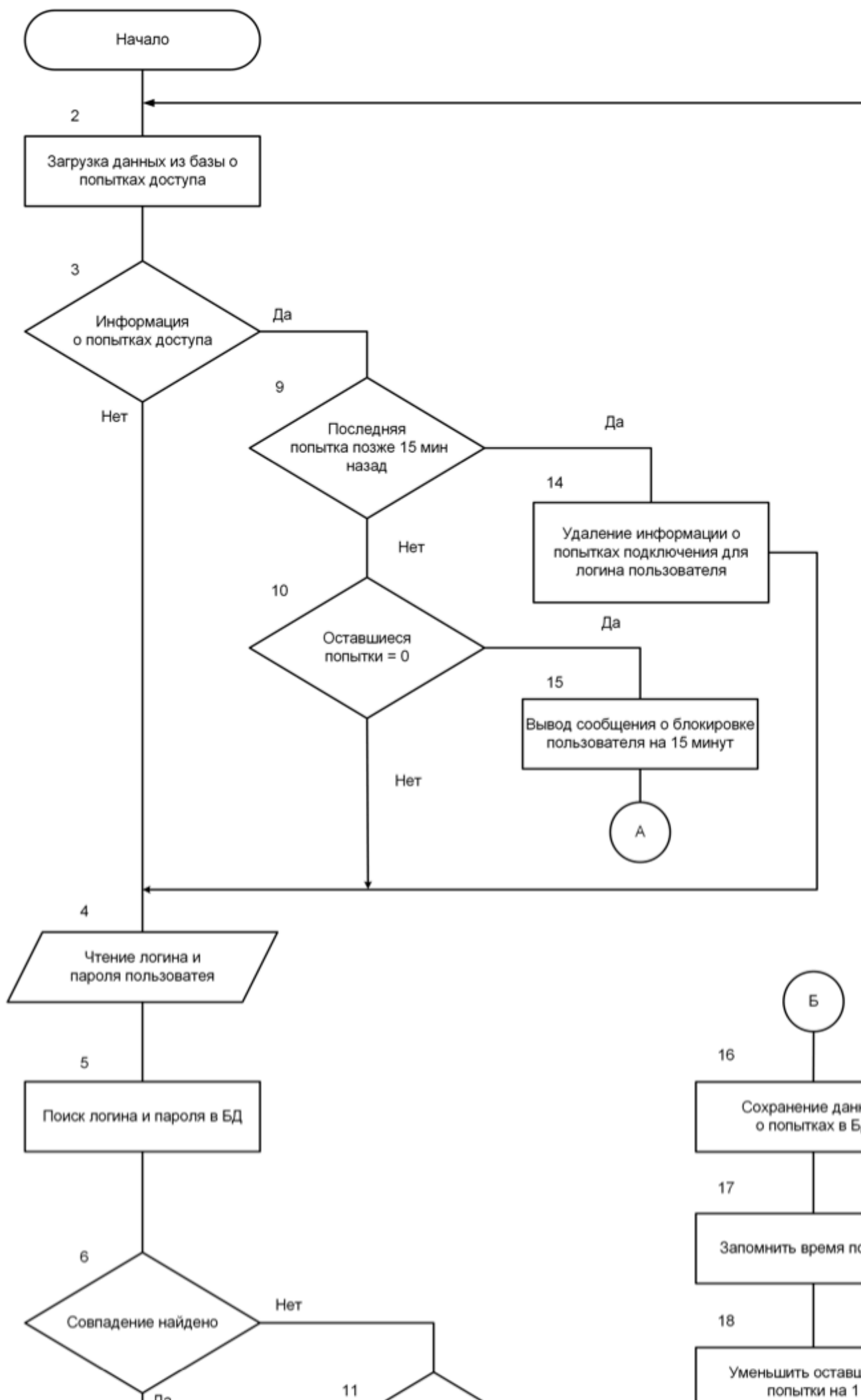


Схема алгоритма входа с блокировкой

Данный алгоритм реализует функцию аутентификации в системе с блокировкой на 15 минут по троекратному вводу неверных данных пользователя.

Сразу после входа на страницу аутентификации система пробует извлечь из контекста приложения данные об уже произведенных пользователем неудачных попытках входа в систему.

Если данные доступны, то система проверяет, как давно была произведена последняя неудачная попытка входа в систему. Если более 15 минут назад, то информация о попытках доступа стирается и пользователю предоставляется доступ к полям ввода идентификационных данных. Если менее 15 минут, то проверяется количество оставшихся попыток, и если попыток не осталось, то приложение перенаправляет пользователя на страницу с сообщением, что его адрес заблокирован на 15 минут, после чего алгоритм прекращает свою работу.

Если у пользователя еще остались попытки или данные о попытках отсутствуют, программа ждет ввода имени пользователя и пароля, после чего ищет пользователя с таким именем и паролем в БД. Если поиск завершился успешно, то пользователя аутентифицирован и может начинать работу с программным средством. В противном случае, происходит сохранение текущего времени, как времени последней неудачной попытки, и уменьшение количества оставшихся попыток на 1, а если это первая попытка, то предварительно инициализируется и сохраняется в контекст приложения информация о количестве оставшихся попыток равным трем.

Разработка алгоритма обработки заявлений абитуриентов

Одной из важных функциональных возможностей системы является обработка заявлений абитуриентов. Данная функциональность доступна только для роли «Оператор ЭВМ» и наследуемых от нее.

В начале работы система составляет запрос к БД на получение данных о существующих в системе заявлениях (блок 11). После чего формируется код страницы и отображается таблица (блок 12). Оператору ЭВМ предоставляется выбор, какое действие он совершит далее:

- обработка отдельного заявления;
- поиск необходимого заявления;
- фильтр (по специальности или дате подаче заявления);
- завершение работы по обработке заявлений.

После настройки фильтра или поиска необходимого заявления (блок 18 и 21) оператор возвращается на страницу отображения заявлений, но уже с измененными параметрами для отображения.

С отдельным заявлением оператор может выполнить следующие действия:

- удалить заявление (блоки 2, 3);
- отредактировать заявление (блоки 8, 9);
- изменить статус заявления (блоки 14-17).

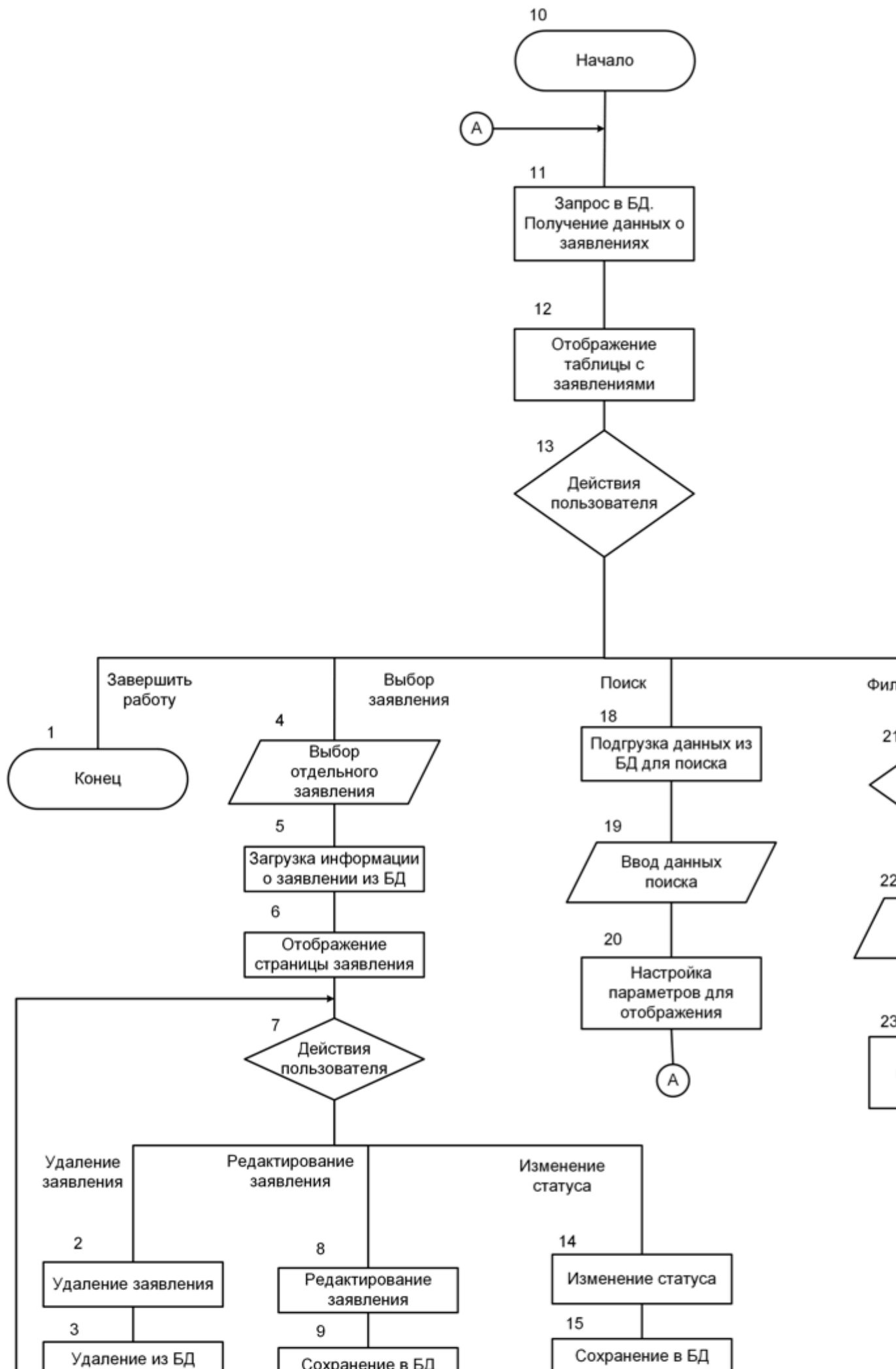


Схема алгоритма обработки заявок пользователя