

УДК 004

**АВТОМАТИЗАЦИЯ РАБОТЫ ПРИЕМНОЙ КОМИССИИ****Чембаев Виталий Валерьевич**

магистрант

Мордовский государственный университет им. Н.П. Огарёва, Саранск

*author@apriori-journal.ru*

**Аннотация.** В статье рассматривается автоматизация работы приемной комиссии. Обзор инструментов для программной реализации.

**Ключевые слова:** приемная комиссия; автоматизация; UML; объектно-ориентированное программирование; C#; ASP.NET; база данных.

---

**AUTOMATION OF THE SELECTION COMMITTEE****Chembaev Vitaliy Valeryevich**

undergraduate

Mordovian state university of N. P. Ogaryov, Saransk

**Abstract.** In article automation of work of a selection committee is considered. Reviewing instruments of programming realization.

**Key words:** selection committee; automation; UML; object-oriented programming; C #; ASP.NET; database.

Приемная комиссия создаётся для организации приема документов от поступающих, проведения вступительных испытаний, конкурса и зачисления в Университет. Деятельность Приемной комиссии направлена на обеспечение соблюдения прав граждан на получение образования, установленных Конституцией Российской Федерации, законности и гласности проведения всех процедур приёма [1].

В настоящее время прием заявок от абитуриентов осуществляется на основе обычных документов. Одной из проблем приемной комиссии является большое количество заявлений от абитуриентов, которая затрагивает операторов приемной комиссии. Это влияет на скорость обработки заявлений, а так же, из-за большого их количества, может привести к их путанице или потере.

В сравнении с имеющимся, новое решение позволит обеспечить более удобный режим доступа к информации, повысить быстродействие, обеспечить надёжное хранение данных и более полный охват функций, подлежащих автоматизации.

Систему можно смоделировать с помощью нескольких разновидностей UML (Unified Modeling Language) диаграмм. Любые (в том числе и программные) системы проектируются с учетом того, что в процессе своей работы они будут использоваться людьми и/или взаимодействовать с другими системами. Сущности, с которыми взаимодействует система в процессе своей работы, называются экторами [2]. Были выявлены основные экторы и варианты использования (см. рис. 1).

Для реализации системы был выбран ASP(ActiveServerPages) .NETMVC (Model – View – Controller) – схема использования нескольких шаблонов проектирования, с помощью которых модель приложения, пользовательский интерфейс и взаимодействие с пользователем разделены на три отдельных компонента таким образом, чтобы модификация одного из компонентов оказывала минимальное воздействие на остальные [3].

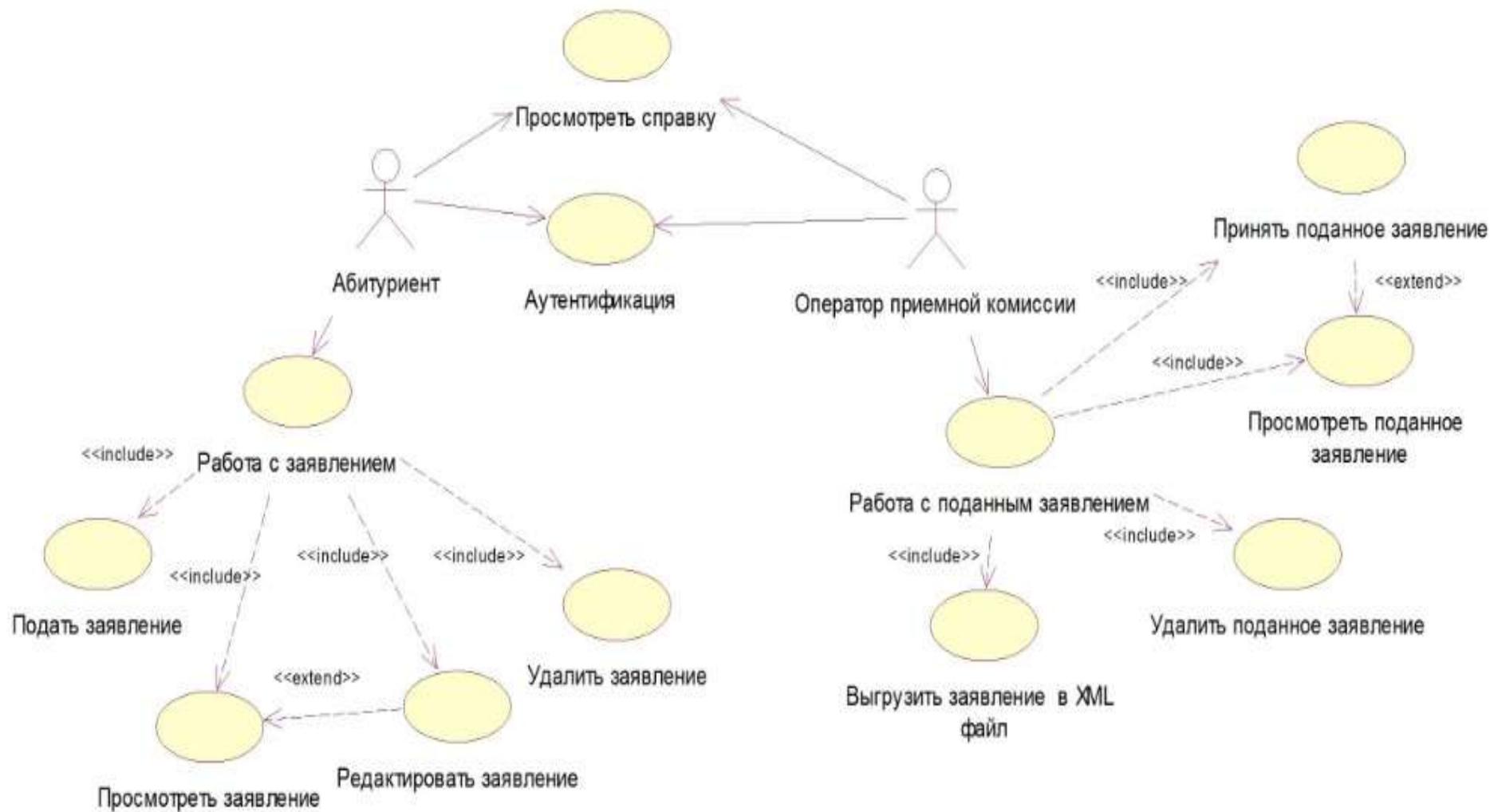


Рис. 1. Диаграмма прецедентов системы

Веб-страницы ASP.NET являются полностью объектно-ориентированными. На страницах ASP.NET с элементами HTML можно работать, используя свойства, методы и события.

Было принято разбить систему на структурные компоненты и показать связи (зависимости) между компонентами (см. рис. 2).

Контроллер является центральным компонентом в архитектуре MVC. Контроллер получает ввод пользователя, обрабатывает его и посылает обратно результат обработки, например, в виде представления. В данной системе контроллер Account отвечает за аутентификацию в системе и выдает права пользователям, прошедшим ее. Контроллер Manage отвечает за изменение параметров учетной записи. Контроллер Home отвечает за работу с абитуриентами. В нем описана логика подачи и изменения заявления. Контроллер Admin отвечает за работу с операторами приемной комиссии. В нем описана логика приема, удаления, редактирования заявления, а так же выгрузки заявления в XML файл.

Все сущности следует выделять в модели. В зависимости от поставленной задачи и сложности приложения можно выделить различное количество моделей.

В модели Statement содержится классы для работы с заявлением, в Serialization содержатся классы для сериализации заявления в XML файл, в AccountViewModels содержатся классы для регистрации, входа и восстановления пароля. В модели IdentityViewModels содержится класс контекста данных для пользователей. В модели ManageViewModels содержатся классы для восстановления пароля.

Хотя работа системы управляется главным образом контроллерами, но непосредственно пользователю приложение доступно в виде представления, которое и формирует внешний вид приложения. Представления – это файлы с расширением cshtml, которые содержат код пользовательского интерфейса в основном на языке html.

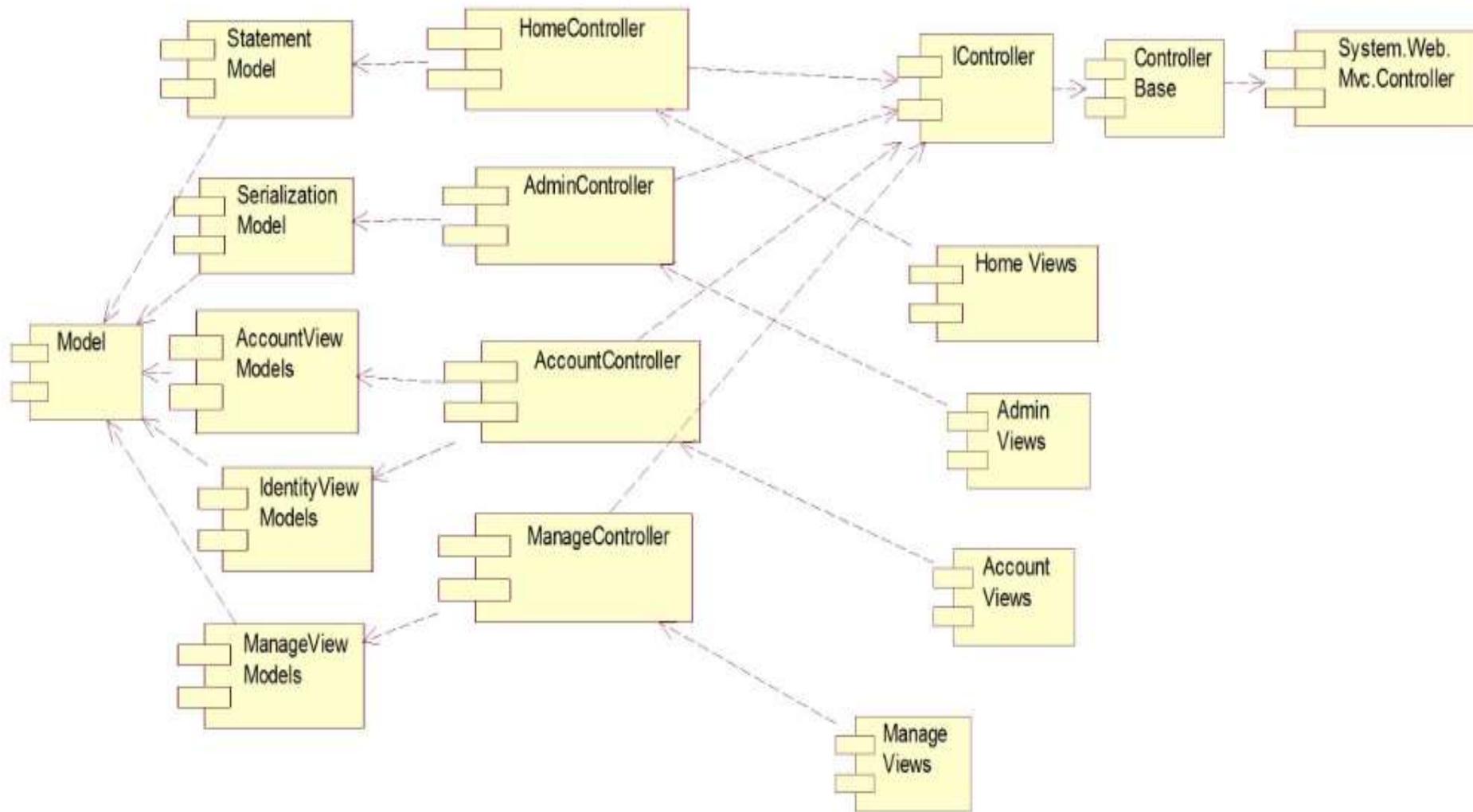


Рис. 2. Диаграмма компонентов

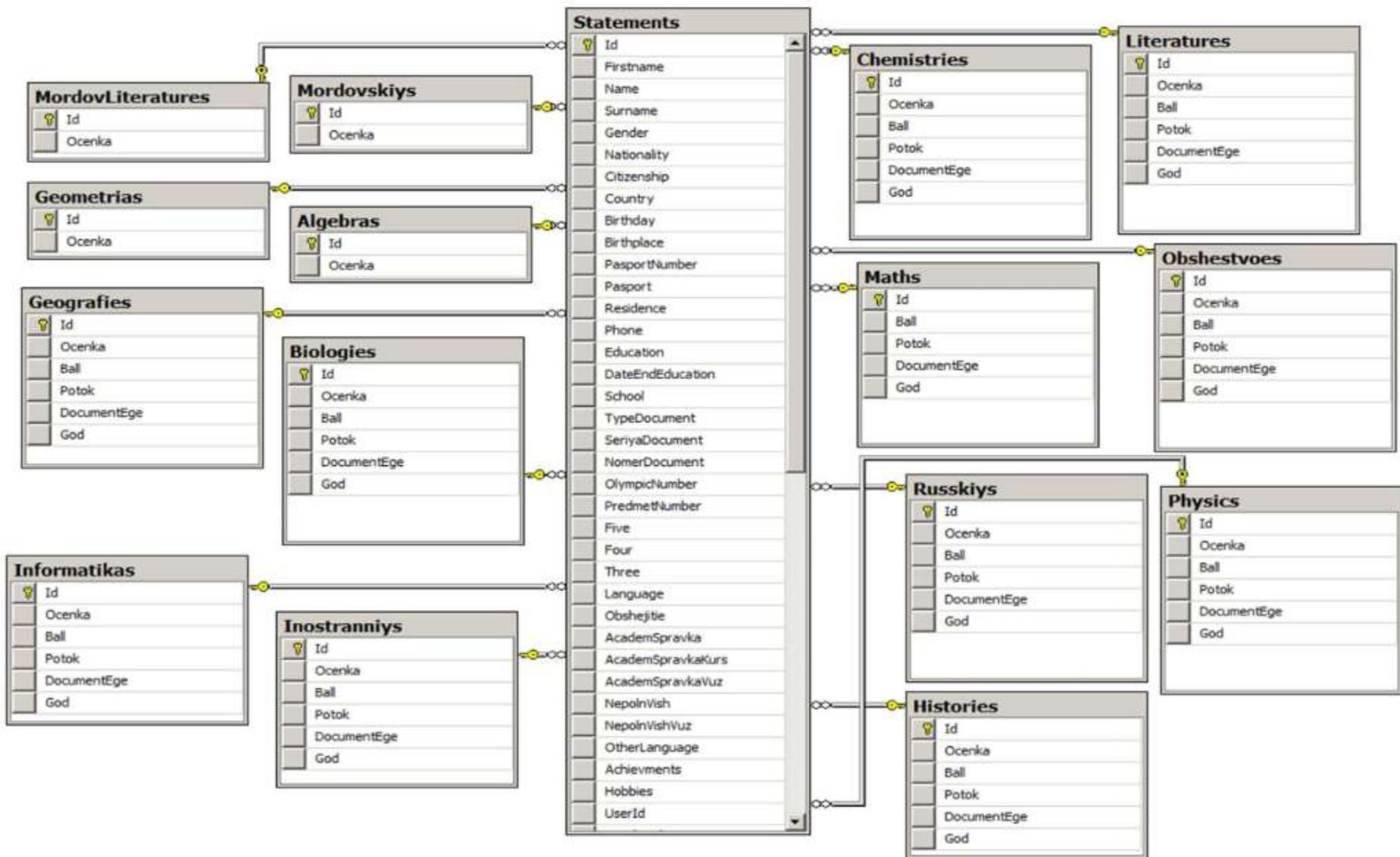


Рис. 3. ER-диаграмм

Одной из главных частей любой информационной системы является база данных. В ней хранится вся информация, необходимая для работы системы [4]. Структура базы данных показана с помощью ER-диаграммы (см. рис. 3). Для работы с данными в ASP .NET MVC рекомендуется использовать EntityFramework, хотя его использование необязательно и всецело зависит от предпочтений разработчика. EntityFramework позволяет вести все операции с базой данных через модель.

Система реализована с помощью объектно-ориентированного языка программирования C#. C# относится к семье языков с C-подобным синтаксисом [5-7].

### **Список использованных источников**

1. Закон Российской Федерации от 10.07.1992 г. № 3266-1 «Об образовании».
2. Бабич А. Введение в UML [Электронный ресурс]. Режим доступа: <http://www.intuit.ru/studies/courses/1007/229/info> (дата обращения: 03.12.2015)
3. Википедия – свободная энциклопедия [Электронный ресурс]. Режим доступа: <https://ru.wikipedia.org/wiki/Model-View-Controller> (дата обращения: 03.12.2015).
4. Базы данных (MSACCESS, MYSQL) / Аббакумов А.А., Акимов В.Л., Егунова А.И. и др. Саранск: Изд-во Средневолжского математического общества, 2011.
5. Александров Э.Э., Афонин В.В. Программирование на языке C в MicrosoftVisualStudio 2010. Саранск: Изд-во Мордов. ун-та, 2010. 424 с.
6. Александров Э.Э., Афонин В.В., Программирование на языке C в MicrosoftVisualStudio 2010 [Электронный ресурс]. Режим доступа: <http://www.intuit.ru/studies/courses/627/483/info>
7. Александров Э. Э., Афонин В.В. Введение в программирование на языке C: Учеб. пособие. Саранск: изд-во Мордов. Ун-та, 2009.