

УДК 004.4

## ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ АВТОМАТИЗАЦИИ РАБОТЫ ПРИЕМНОЙ КОМИССИИ УЧРЕЖДЕНИЯ ОБРАЗОВАНИЯ

*Славинский Г.А.. студент*

*Белорусский государственный университет информатики и радиоэлектроники,  
Институт информационных технологий,  
г. Минск, Республика Беларусь*

*Климов С.М. – ст. препод. каф. ИСиТ*

**Аннотация.** Цель проекта – разработка веб-приложения для автоматизации работы приемной комиссии учреждения образования. Проект предназначен для повышения уровня эффективности работы приемной комиссии и улучшения процесса получения, хранения и обработки данных абитуриентов. Веб-приложение ориентировано на потребности учреждения образования «Полоцкий государственный химико-технологический колледж». Результат применим в условиях приемных кампаний учреждений образования, ведущих набор по уровням профессионально-технического и среднего специального образований.

**Ключевые слова.** Автоматизация работы, приемная комиссия, фреймворк Django.

**Введение.** В ходе прохождения технологической практики в учреждении образования «Полоцкий государственный химико-технологический колледж», были выделены следующие трудности, возникающие в ходе приемной кампании:

- постоянная необходимость личного присутствия абитуриентов;
- большое число неравномерно поступающих обращений со стороны абитуриентов.

Также, была выделена возможность по улучшению деятельности приемной комиссии через предложение решения для работы с данными абитуриентов, предоставляющего как надежное хранилище, так и гибкий доступ к данному хранилищу.

Приемная кампания является сложным и ограниченным во времени процессом, в ходе которого обрабатываются большие объемы информации [1].

Многие высшие учебные заведения имеют системы для получения, обработки и хранения данных абитуриентов. При поступлении в Белорусский государственный университет информатики и радиоэлектроники каждый абитуриент должен создать личный кабинет, где следует заполнить все необходимые данные и составить заявление с помощью конструктора. Наличие аккаунта позволяет отслеживать статус документов и обращаться в приемную комиссию по возникающим вопросам [2].

Данные системы все более актуальны, независимо от ступени образования, достоинства данного подхода:

- экономия времени, посещение абитуриентом комиссии сводится к подаче оригиналов документов, проверке данных и подписи;
- централизованная система информирования, личный кабинет позволяет абитуриенту следить за ходом вступительной кампании, получить обратную связь от приемной комиссии.

Подобные системы позволяют предварительную подачу документов, но не отменяют необходимость формирования перечня оригинальных документов для подачи в приемную комиссию.

На основе рассмотренных языков (Python, PHP, C#), для разработки был выбран язык программирования Python, как функциональный и более простой в использовании. Из Python-фреймворков был выбран Django, реализующий базовую функциональность веб-приложения на базе шаблона проектирования MVC и содержащий множество функций, позволяющих создать прототип в кратчайшие сроки, включая средства взаимодействия с базой данных и инструменты отладки [3].

**Основная часть.** При обзоре аналогов программного средства, было рассмотрено как решение для организации данных абитуриентов в рамках файлов СУБД Microsoft Access, так и средство для комплексного управления деятельностью учреждения образования «1С: Колледж», у данных аналогов были выделены следующие достоинства:

- простота, интуитивно понятный интерфейс;
- большое количество материалов для выбранных платформ;
- автоматизация всех этапов деятельности приемной комиссии, включая отчетность.

Выявленные недостатки «1С: Колледж»:

- автоматизация деятельности приемной комиссия не является главной задачей решения;
- возможность выбора нескольких специальностей не актуальна для заказчика.

Выявленные недостатки решений на базе Microsoft Access:

- сложность с организацией доступа к данным;
- отсутствие разграничения доступа.

Главная задача приложения – автоматизация работы приемной комиссии с целью упрощения и качественного улучшения процесса приема абитуриентов. Приложение позволяет легко структурировать и систематизировать данные.

Входные данные предполагают:

— данные абитуриентов – данные, предоставляемые со стороны абитуриента, необходимые для формирования заявления (включают личные данные абитуриентов, данные документов об образовании и представителей абитуриентов).

— организационные данные – данные, предоставляемые со стороны учреждения образования, необходимые для организации процесса приема данных абитуриентов (включают информацию об изучаемых абитуриентами иностранных языках, возможных статусах представителей, актуальных специальностях, установленных при поступлении льготах и т.д.);

Выходные данные предполагают сформированные списки организационных данных, данных абитуриентов, заявлений и документов, предоставленных со стороны абитуриентов.

С целью точной спецификации функций была разработана контекстная диаграмма, представлена на рисунке 1.

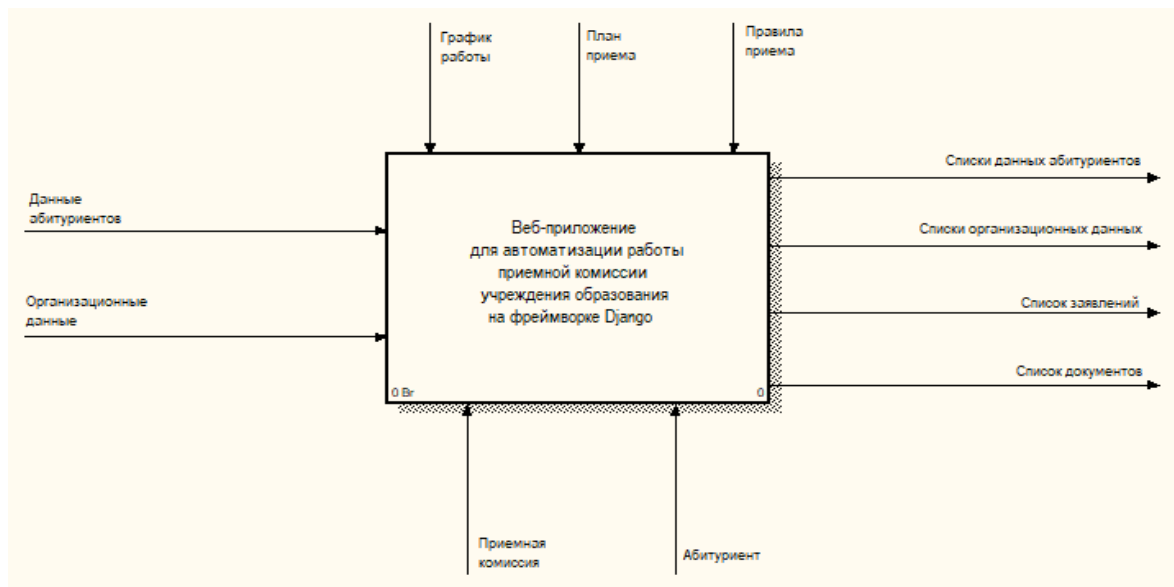


Рисунок 1 – Контекстная диаграмма

Далее была проведена декомпозиция основной конструкции, выполненная согласно принципу ограниченной сложности [4]. Результат декомпозиции представлен на рисунке 2.

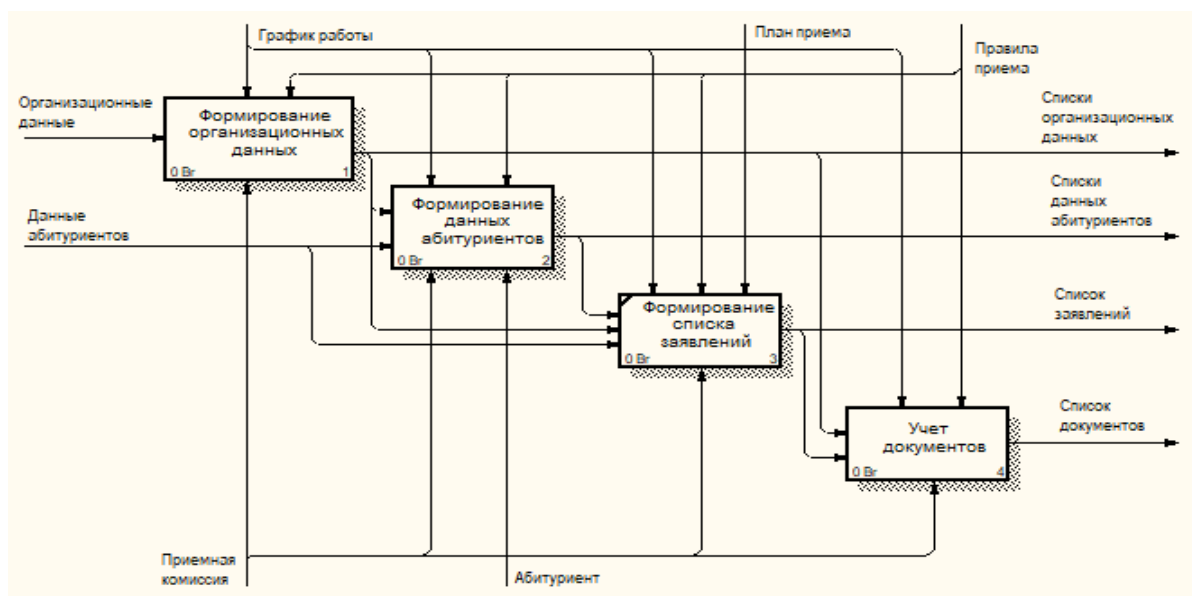


Рисунок 2 – Декомпозиция основной конструкции

Фреймворк Django предлагает готовую таблицу для данных пользователей, содержащую поля логина, пароля и электронной почты. Для реализации механизмов авторизации и регистрации использовались встроенные классы LoginView, AuthenticationForm и CreateView, UserCreationForm.

Все категории пользователей, кроме абитуриента, регистрируются через Django-панель.

MVC (Model – View – Controller) – набор архитектурных идей и принципов, сформированных норвежским ученым Трюгве Реенскауг, для построения сложных систем с пользовательским интерфейсом, возможности удобного и лаконичного определения именуется «паттерн» [5].

В фреймворке Django реализуется архитектурный паттерн MTV (Model – Template – View) являющийся модификацией MVC и включающий:

- url-диспетчер – определяет, какой ресурс должен обрабатывать url-запрос;
- model – описывает данные, используемые в приложении;
- template – логика представления, отображает пользователю результат, аналог View в MVC;
- view – получает и обрабатывает запрос, отправляет пользователю некоторый ответ, аналог Controller в MVC.

Функциональные возможности приложения предполагают формирование данных, а точнее – базовые функции работы с БД, реализованные на основе встроенных классов: ListView и DetailView (отображение данных), CreateView (создание), UpdateView (модификация), DeleteView (удаление).

Формирование данных возможно после прохождения авторизации и получения доступа к одной из подсистем, разграничение доступа реализовано через UserPassesTestMixin. Выделенные подсистемы представлены на диаграмме вариантов использования (рис. 3).

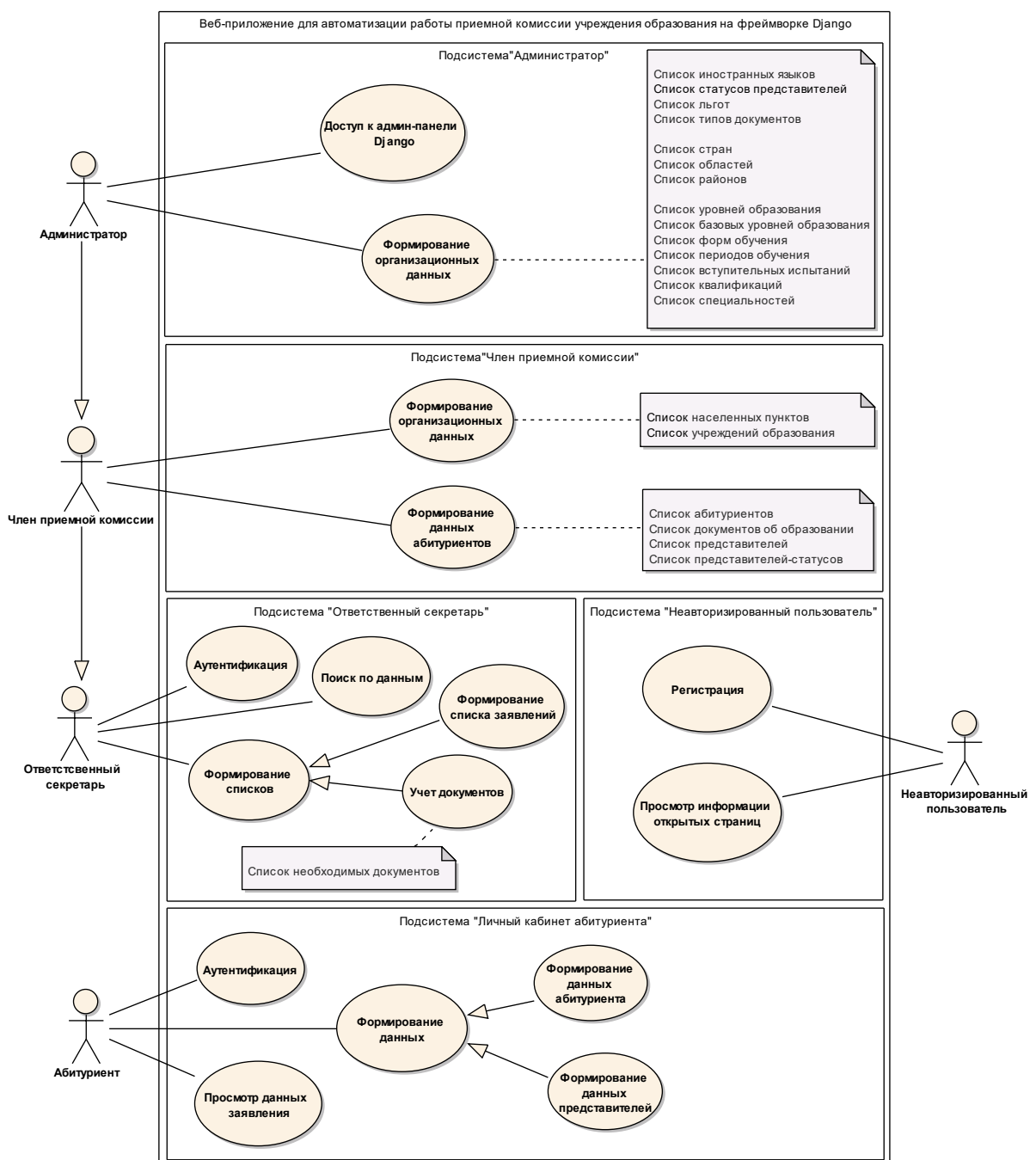


Рисунок 3 – Диаграмма вариантов использования

Перечень сформированных стран представлен на рисунке 4.

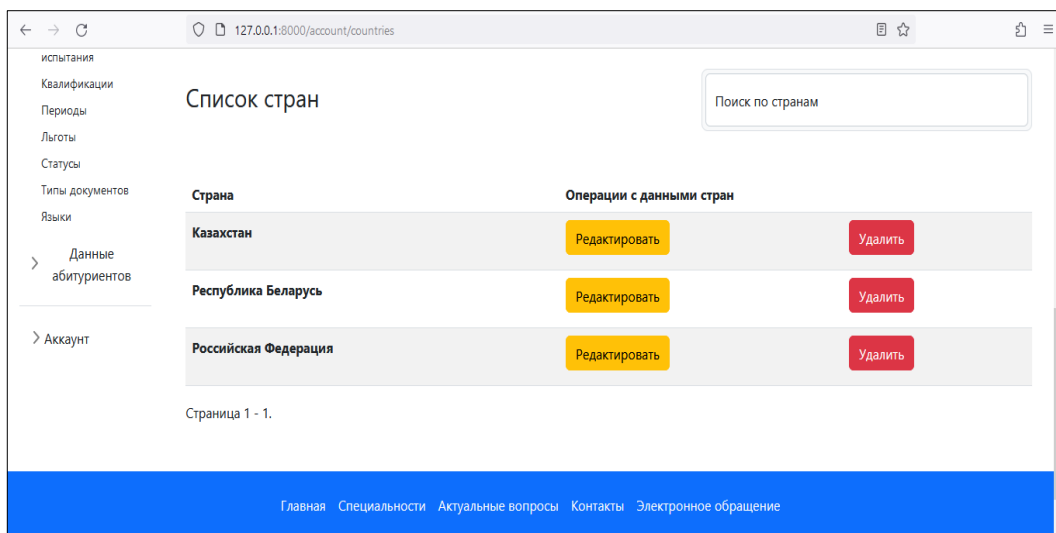


Рисунок 4 – Перечень сформированных стран, подсистема администратора

Добавление данных области, на основе перечня стран, представлено на рисунке 5.

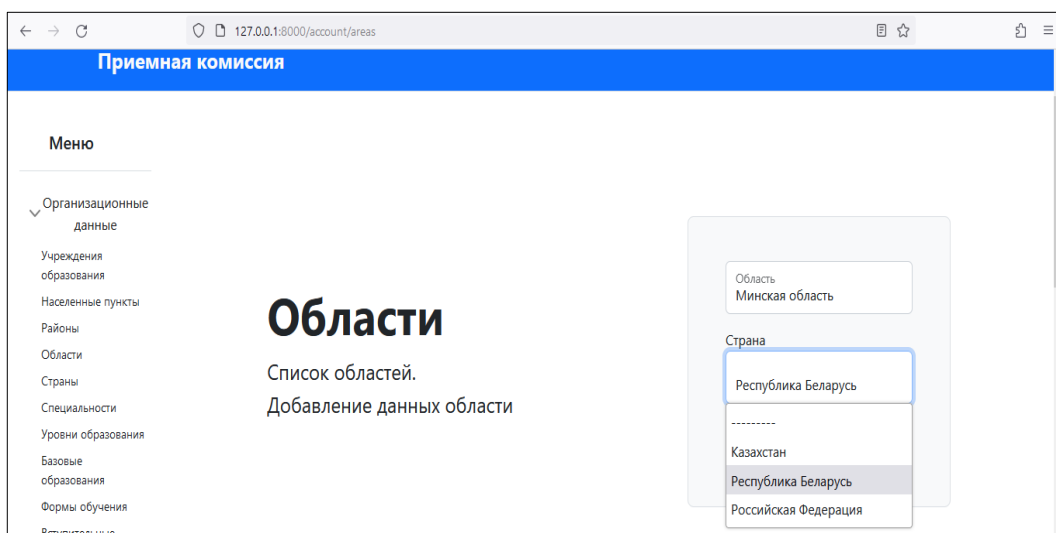


Рисунок 5 – Добавление данных области, подсистема администратора

Результат формирования данных – списки, включая списки абитуриентов, организационных данных, заявлений и поданных документов. Данные заявления представлены на рисунке 6.

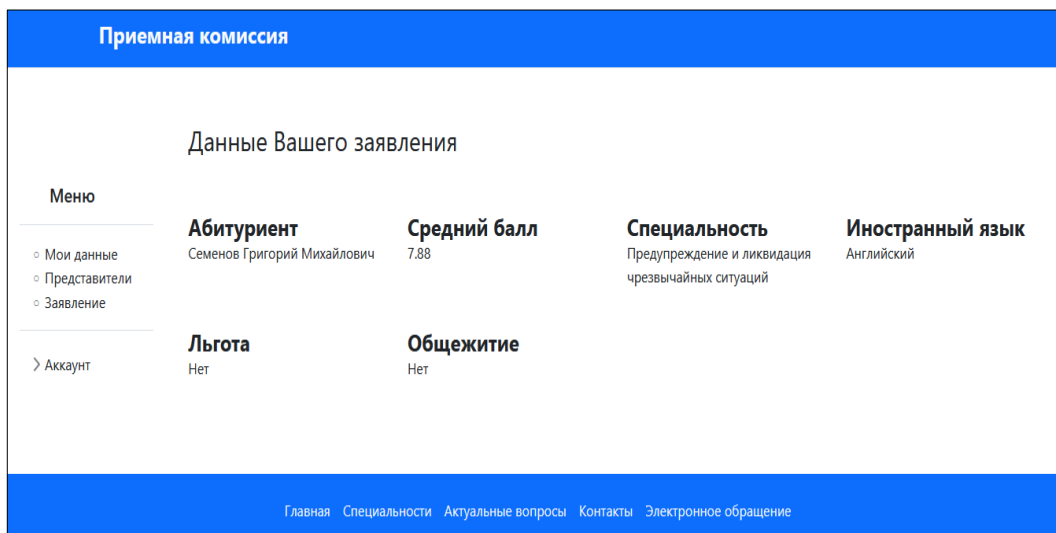


Рисунок 6 – Сформированное заявление, подсистема абитуриента

В случае, если абитуриент еще не обращался в приемную комиссию, то будет представлено уведомление о необходимости посещения приемной комиссии.

**Заключение.** Веб-приложение реализовано на Python-фреймворке Django с использованием инструмента Bootstrap 5, включающего HTML и CSS шаблоны, и реляционной системы управления базами данных MySQL. Для организации подстановки данных родительских таблиц в поля выбора использовался jQuery-плагин Select2.

В процессе разработки были учтены основные потребности в организации данных абитуриентов на время приемной кампании, реализованы базовые функции, используемые при работе с базами данных: создание, чтение, модификация, удаление.

Приложение отличается от аналогичных программных средств наличием широких функциональных возможностей и своей ориентированностью на автоматизацию работы приемной комиссии.

Технико-экономическое обоснование веб-приложения, в ценах на ноябрь 2022 года за счет средств заказчика, прогнозирует отпускную плановую цену в объеме около 13977,11 рублей.

Проект внедрен в Полоцкий государственный химико-технологический колледж.

Аргументами для внедрения приложения послужило улучшение процесса получения, хранения и обработки данных абитуриентов:

- интуитивно понятный и адаптивный пользовательский интерфейс;
- разграничение прав доступа, согласно выделенным категориям пользователей;
- организация клиент-серверного взаимодействия;
- наличие средств защиты информации от распространенных атак;
- автоматизация рутинных, массово представленных задач, какой, к примеру, является расчет среднего балла документа об образовании абитуриента.

От внедрения ожидается повышение уровня эффективности работы приемной комиссии колледжа. В дальнейшем, с целью оптимизации работы комиссии, предполагается представление приемной комиссии, как многоканальной системы массового обслуживания с неограниченной очередью, рассмотрение возможности внедрения в приложение функционала для оценки эффективности работы сотрудников.

**Список использованных источников:**

- 1.1. Батура, М.П. *Повышение эффективности работы приемной комиссии университета путем внедрения автоматизированной системы подачи заявлений и зачисления* / М.П. Батура, Б.В. Никульшин, М.В. Михневич, В.Г. Русин, Н.А. Дубко // *Информационные технологии и системы 2015 (ИТС 2015): материалы международной научной конференции (БГУИР, Минск, Беларусь, 28 октября 2015)* / редкол. : Л. Ю. Шилин [и др.]. – Минск: БГУИР, 2015. – С. 210-211.
2. Приемная кампания-2021 [Электронный ресурс]. – Режим доступа: <https://www.bsuir.by/ru/news/106894-priemnaya-kampaniya-2021-uproschenie-podachi-dokumentov-i-obnovlennye-spetsialnosti-v-bguir>. – Дата доступа: 10.10.2022.
3. Дронов, В. *Django: Практика создания Web-сайтов на Python* / В. Дронов. – СПб : Питер Ком, 2016. – 530 с.
4. Бахтизин, В. В. *Структурный анализ и моделирование в среде CASE-средства VPwin: учеб. пособие по курсу "Технология проектирования программ" для студ. спец. 40 01 01 "Программное обеспечение информационных технологий"* / В. В. Бахтизин, Л. А. Глухова. – Минск: БГУИР, 2002. – 44 с.: ил.
- Знакомство с паттерном MVC (Model-View-Controller) [Электронный ресурс]. – Режим доступа: <https://javarush.com/groups/posts/2536-chastjh-7-znakomstvo-s-patternom-mvc-model-view-controller>. – Дата доступа: 07.11.2022

UDC 004.4

## WEB APP FOR WORK AUTOMATION OF THE EDUCATIONAL ESTABLISHMENT ADMISSIONS COMMITTEE

Slavinsky G.A.

*Institute of Information Technologies of the Belarusian State University of Informatics and Radioelectronics,  
Minsk, Republic of Belarus*

*Klimov S.M. – Senior Lecturer*

**Annotation.** The goal of the project is to develop a web application for work automation of the educational establishment admissions committee. The project is designed to increase the work efficiency level of the admissions committee and improve the process of obtaining, storing and processing applicants' data. The web application is focused on the needs of the Polotsk State College of Chemical Technology. The result is applicable in the admission campaigns of educational establishments which carry out admission on the basis of the levels of professional-technical and secondary specialized education.

**Keywords.** Work automation, admissions committee, Django framework