МИНИСТЕРСТВО СВЯЗИ И ИНФОРМАТИЗАЦИИ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования

«БЕЛОРУССКАЯ ГОСУДАРСТВЕННАЯ АКАДЕМИЯ СВЯЗИ»

ФАКУЛЬТЕТ ЭЛЕКТРОСВЯЗИ

КАФЕДРА -----

ТЕМА

Пояснительная записка

к курсовой работе

по дисциплине

«Основы разработки и сопровождения Web - приложений»

Выполнил студент гр. Со841 Гусев Н.Н.

Руководитель Преподаватель

Минск 2020

# АННОТАЦИЯ

Данная курсовая работа посвящена разработке Telegram – бота, обеспечивающего конвертацию валютного курса относительно белорусского рубля, но основе валютного курса на текущий момент, установленного Национальным банком Республики Беларуси.

Ключевые слова: Web-приложение, модуль, программное обеспечение, telegram, API, тестирование, тестирование программного обеспечения.

В первом разделе рассмотрен вопрос проектирования и разработки Web-приложения, обоснование выбора стека технологий.

Во втором разделе описана работа созданного Web-приложения, описание его основных функций, а также цели его создания.

В третьем разделе представлены результаты тестирования разработанного Web-приложения.

Объем пояснительной записки составляет 19 листов, и она содержит 1 таблицу, 9 рисунков, и 3 источника литературы.

# СОДЕРЖАНИЕ

[Введение 4](#_Toc509745519)

[1 Тестирование программного обеспечения. Методики тестирования программного обеспечения 5](#_Toc509745520)

[1.1 Тестирование программного обеспечения, его цели и задачи 5](#_Toc509745521)

[1.2 Основные методики тестирование программного обеспечения 6](#_Toc509745522)

[2 Описание разработанного web-приложения. Обзор его компонентов 9](#_Toc509745523)

[2.1 Описание разработанного web-приложения 9](#_Toc509745524)

[2.2 Обзор web-приложения и его компонентов 9](#_Toc509745525)

[2.3 Описание основных функций разработанного web-приложения 15](#_Toc509745526)

[3 Описание тестирования разработанного web-приложения и его результаты 17](#_Toc509745527)

[Заключение 18](#_Toc509745528)

[Список использованных источников 19](#_Toc509745529)

# ВВЕДЕНИЕ

На сегодняшний день WEB – разработка занимает приоритетное направление, технологии которой способствующие покрыть задачи многих направлений.

На сегодняшний день тестирование программного обеспечения несомненно играет большую роль при создании программного продукта. Предусмотренное вовремя тестирование программного обеспечения может спасти от огромных затрат на доработку продукта. Именно поэтому в процессе реализации курсового проекта была поставлена цель не только разработать приложение, но и протестировать его.

Целью курсового проекта является практическое применение знаний, связанный с тестированием программного обеспечения, приобретённых в процессе обучения, а также приобретение новых умений в сфере разработки и тестирования программного обеспечения.

К основным задачам курсового проекта относится:

1. Выбор языка программирования, на котором будет написано приложение

2. Изучение его специфики для решения нужных задач

3. Написание Web-приложения

4. Тестирования разработанного Web-приложения.

Объектом курсового проекта является Web-приложение «Учёт успеваемости и посещаемости студентов», которое в процессе реализации проекта необходимо было разработать. В наше время данное приложение является весьма актуальным. Учитывая постепенное внедрение информационных технологий во все сферы жизни человека можно спрогнозировать использование подобных приложений во всех учебных заведения и учреждениях образования. Среди плюсов данного Web-приложения можно отметить:

1. сохранение древесных ресурсов, путём неиспользования бумажных носителей

2. система доступа, которая не позволит никому, кроме преподавателей исправлять данные

3. в отличии от бумажных носителей не имеет никаких проблем с редактированием информации.

# Проектирование и разработка WEB - приложения

* 1. **Этапы проектирования и разработки**

Проектирование корпоративного веб приложения, как и любого другого приложения, стоит начать с определения первоначальный целей и области решаемых задач. Создать реестр заинтересованных лиц.

На следующем этапе необходимо собрать требования к приложению, которое необходимо разработать. Уточнить цели и область решаемых задач и построить иерархическую структуру работ.

каждое web-приложение отправляет http запросы на web-сервер для получения полезных данных. Программа под управлением web-сервера использует ту или иную модель для хранения данных. В современном мире чаще всего используются базы данных, SQL или NoSQL.  
  
    Формально каждое web-приложение можно разбить на 3 взаимно независимые части.

1. Модуль, который исполняется WEB-браузером. Это приложение может быть написано на любом языке, который поддерживает браузер. Чаще всего используется язык JavaScript, как наиболее поддерживаемый и имеющий большую библиотечную поддержку. Это очень важно, так как позволяет существенно экономить бюджеты проектов.
2. Модуль, исполняемый на серверной стороне под управлением web-сервера. Это приложение может быть написано на любом языке, интерпретацию которого поддерживает выбранный Вами web-сервер. Последнее время, часто, в качестве языка программирования выбирается язык Java. Этот язык также имеет серьезную библиотечную поддержку.
3. База данных. В этой области так же существует достаточно широкий выбор. Есть промышленные базы данных, такие как Oracle, DB2, PostgreSQL. Есть легкие базы данных, такие как MySQL. База данных выбирается, основываясь на целях и области решаемых задач.

При построении архитектуры web — приложения необходимо максимально уменьшить зависимость между структурными единицами. В общем случае приложение состоит из трех структурных единиц:

1. Модуль, который работает под управлением браузера.
2. Модуль, который работает под управлением web -сервера.
3. База данных.

Для достижения цели максимальной независимости между структурными единицами, необходимо чтобы каждая структурная единица оперировала только необходимым ей набором данных. Рассмотрим более подробно.

Браузер — это прикладное программное обеспечение для просмотра web страниц.

HTML – это стандартный язык разметки документов. Большинство современных web-браузеров способны интерпретировать язык HTML.

Web сервер — это программное обеспечение, которое способно принимать HTTP запросы от клиентов, обрабатывать их и отправлять ответ в соответствии со стандартом протокола.

База данных — это представленная в объективной форме совокупность самостоятельных материалов, систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью ЭВМ.

Для минимизации зависимостей между «Браузером» и Web-сервером необходимо, чтобы язык разметки HTML был задействован только в браузере, а Web-сервер предоставлял интерфейс для получения необходимых данных для страницы.

Для решения этой задачи необходимо:

Определить цели и область решаемых задач, которые будут решаться в рамках создаваемого интерфейса.

1. Определить API серверной части.
2. Выбрать протокол взаимодействия между серверной и клиентской частью. Создание протокола удобнее всего выбрать на базе XML, так как большинство современных браузеров имеют встроенную поддержку этого языка.
3. Написать документ, в котором будет изложен протокол.
   1. **Обоснование выбора стека технологий**

Для реализации Back – End части курсового задания был выбран динамический, рефлективный, интерпретируемый высокоуровневый язык программирования Ruby. Фактически, Front – end части, в приложении, как таковой нет. Это обусловлено использованием продуманной системы API Telegram.

Ruby является мощным инструментом для разработки. К его плючам можно отснести:

1. Продуманный и очень лаконичный синтаксиссинтаксис, на котором код выглядит максимально читаемым. Можно писать действительно понятно и лаконично.
2. Огромное количество реализованных библиотек, подключение которых не вызывает проблем. Самый главный плюс ruby — это rails, лучший фреймворк. Четко реализованные паттерны, не то, что на node.js, крути как хочешь. Само написание кода напоминает английскую речь.

Можно сказать, что не все герои носят плащи. Ruby, изучив его плюсы, тому примеру. Для полной оценки языка программирования стоит разобрать так же и его минусы, которых, увы, не мало:

1. Требователен к ресурсам, всю историю разработки языка сопровождают серьезные оптимизации по потреблению памяти. В эталонной реализации интерпретатора (MRI) отсутствует реальная многопоточность с использованием нескольких ядер процессора (GIL).
2. Динамическая типизация. Но это сложно отнести к минусам, ведь она позволяет ускорить разработку, хоть и с потерей качества.
3. Язык практически не развивается на фоне его ближайших конкурентов (JavaScript и Python). Взрывной рост интереса к Ruby основывался в первую очередь на Ruby on Rails. Но сейчас, когда Rails-like фреймворки есть в каждом популярном ЯП, Ruby мало что может предложить.

Ruby был выбран на основе комьюнити, огромного количества качественно написанных туториалов и best practices; Rails фреймворка, делающего разработку максимально понятной от самого старта проекта, до деплоя; богатства подходов и реализаций для написания тестов: RSpec, MiniTest и так далее.

Для получения текущего курса валют был использован API Национального Банка Республики Беларусь. API реализует получения официального курса белорусского рубля по отношению к иностранным валютам, устанавливаемого Национальным банком Республики Беларусь. С сентября 2020 года Национальным Банком Республики Беларусь был осуществлен переход с HTTP на HTTPS. С момента перехода все сервисы интернет-сайта (новости в формате RSS, API, получение данных, используя XML, и др.) стали также доступны только по протоколу HTTPS.

Для связки Back – end части и Telegram было использовано BOT API Telegram и библиотека для Ruby – Telegram Bot API и HTTParty. Все запросы Telegram Bot API осуществляются через HTTPS.

Так же был предусмотрен прокси, в случае «технических проблем» у операторов и провайдеров связи и интернет.

# 2 ОПИСАНИЕ РАЗРАБОТАННОГО TELEGRAM БОТА

## **2.1 Описание разработанного Web-приложения**

Основной задачей данного telegram бота является упрощение получения курса валют и конвертация на основе белорусского рубля.

При создании данного электронного telegram бота были проанализированы различные способы реализации задания. Как результат, путём использования данного Web-приложения несколькими лицами, был выявлен наиболее комфортный для большинства пользователей интерфейс.

## **2.2 Обзор Web-приложения и его компонентов**

**2.2.1 Telegram Bot**

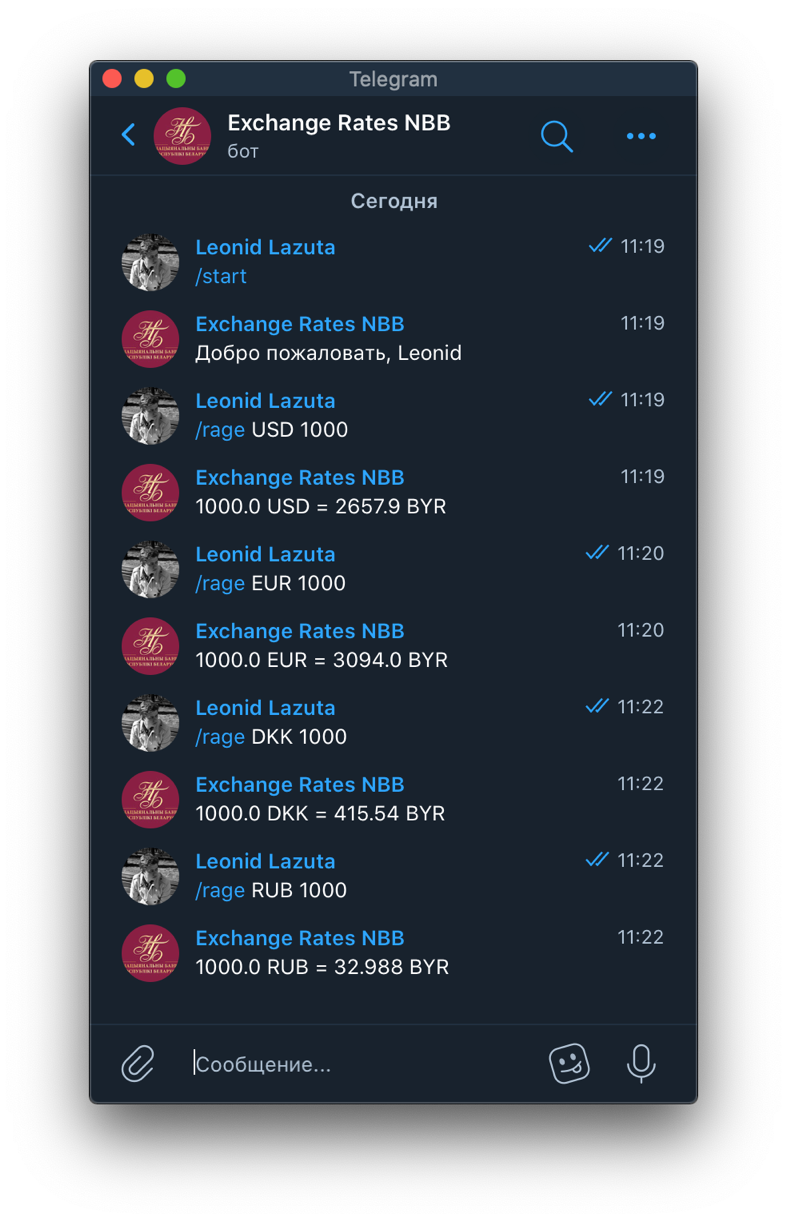


Рисунок 2.1 – Старнциа бота в Telegram

Бот поддерживает следующие команды для взаимодейсвтия с ним:

* /start – Запуск бота. Приветствие пользователя.
* /list – поддерживаемые курсы валют (Символьные коды валюты, обслуживающиеся Национальным Баноком РБ)
* /rage КОД\_ВАЛЮТЫ СУММА – конвертация указанной валюты в белорусккий рубль по курсу Национального Банка Республики Беларусь.

**2.2.2 Авторизация**

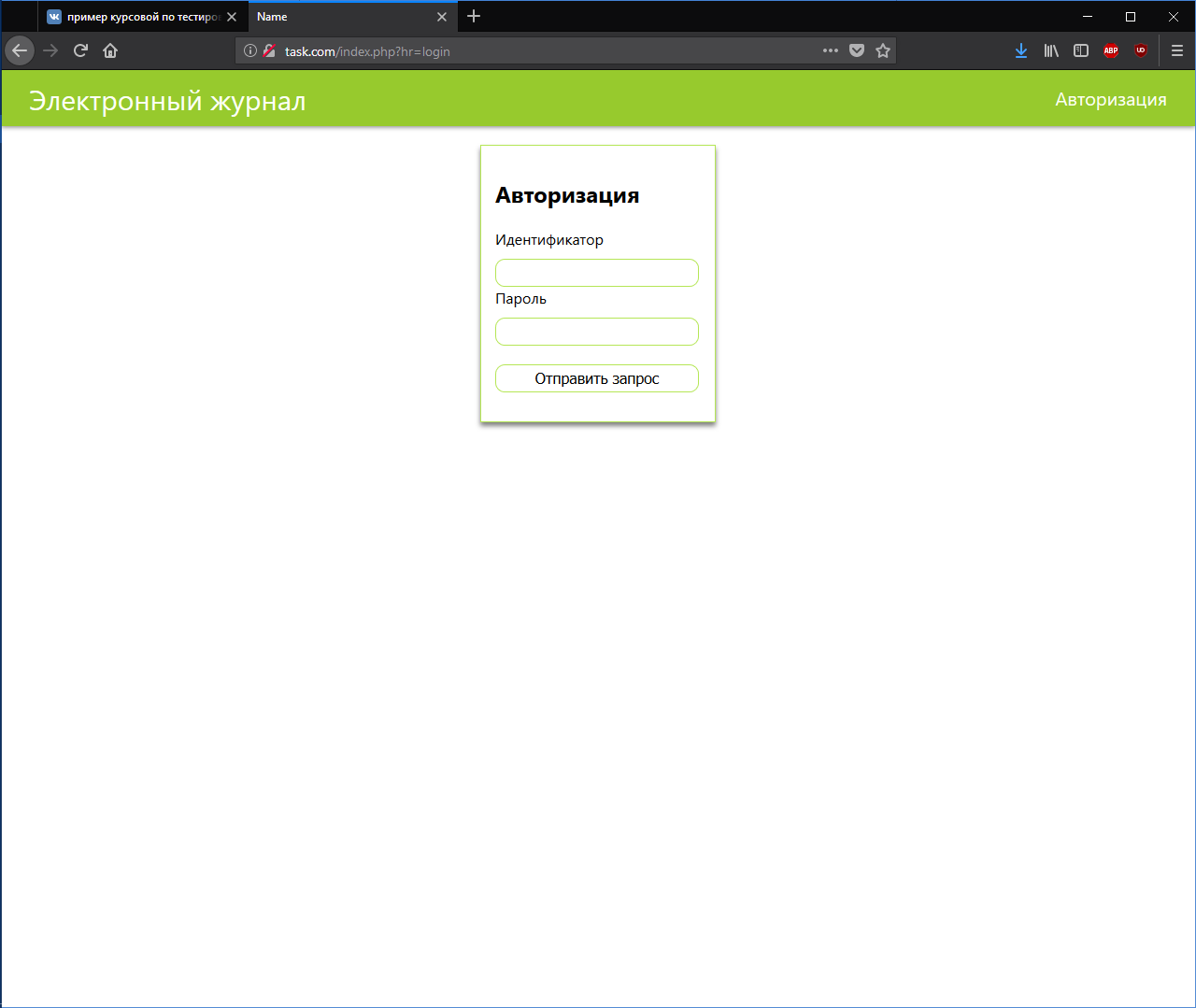


Рисунок 2.2 – Страница авторизации

1. Загрузка формы авторизации.

2. Поле «Идентификатор» допускает ввод только 6 символов(цифр).

3. Поле «Пароль» должно шифровать вводимые символы. Ограничений на количество вводимых символов нет.

4. Отправка формы. Форма должна обрабатываться сервером при условии валидного заполнения всех полей. В случае, если хоть одно поле заполнено не валидно или его заполонены отсутствует, то должна появляться ошибка.

5. В качестве тестового аккаунта был выделен аккаунт со следующими данными для входа. Пароль: 123; Идентификатор: 110011.

**2.2.3 Страница личный кабинета**

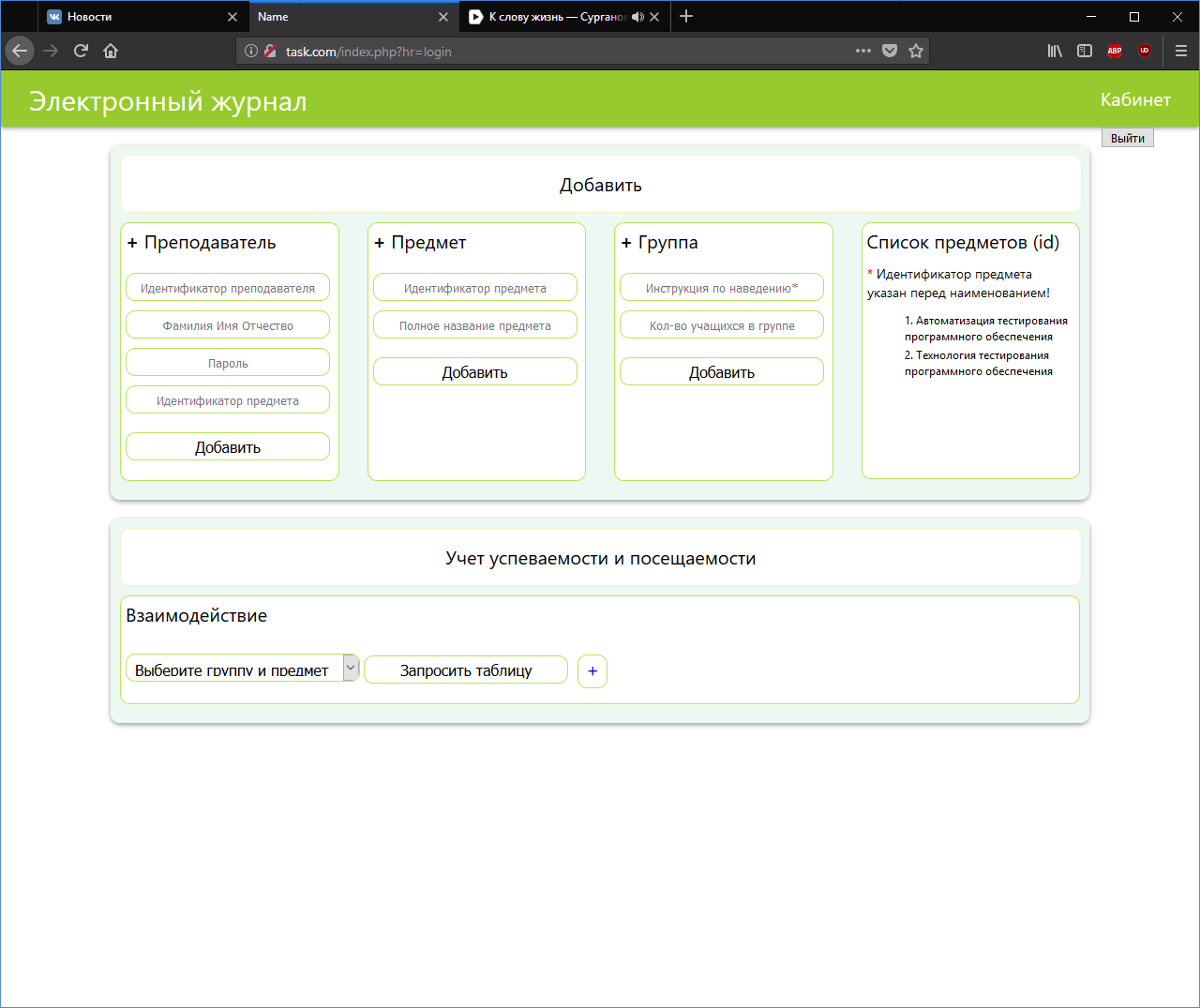


Рисунок 2.3 – Личный кабинет авторизированного пользователя

1. Загрузка страницы с 3 формами, одним списком и секцией для взаимодействия с таблицей.

2. Проверка всех действий на стороне сервера: создание, заполнение и редактирование БД.

3. Кнопка отправки формы должна срабатывать в случае валидного заполнения всех полей. В противном случае – вывод сообщения об ошибки.

**2.2.4 Модуль «Добавить преподавателя»**

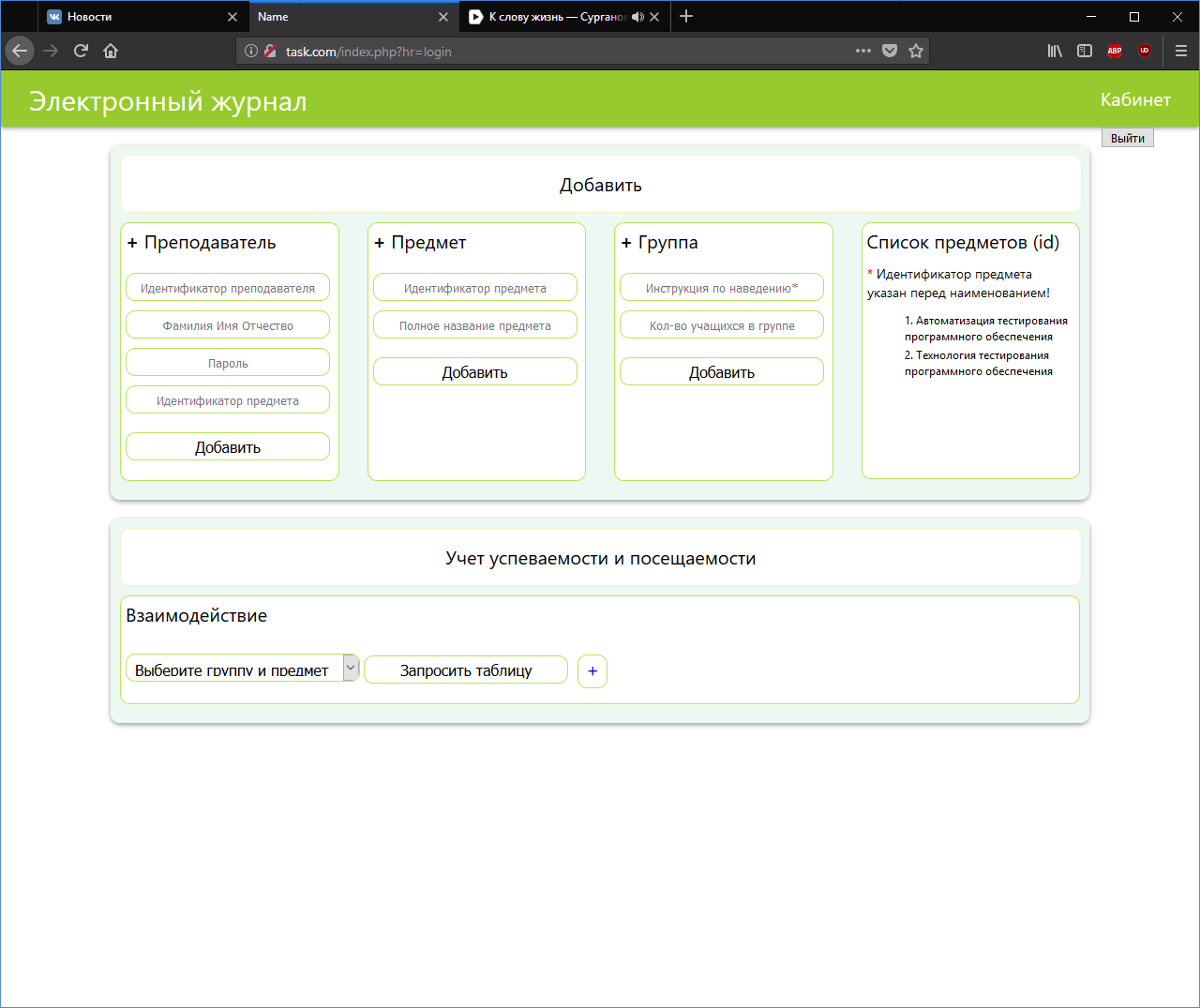


Рисунок 2.4 – Модуль добавления преподавателя

1. Поле "Идентификатор преподавателя". Доступен ввод только 6 символов (цифр).

2. Поле "ФИО". Доступен ввод ФИО в полной форме и спец. Символа «-».

3. Поле "Пароль". Вводимые значения шифруются.

4. Поле "Идентификатор предмета". Доступен ввод только от 1 до 3 символов (цифр)

5. Общее правило для отправки, указанное в модуле «Личный кабинет».

**2.2.5 Модуль «Добавить предмет»**

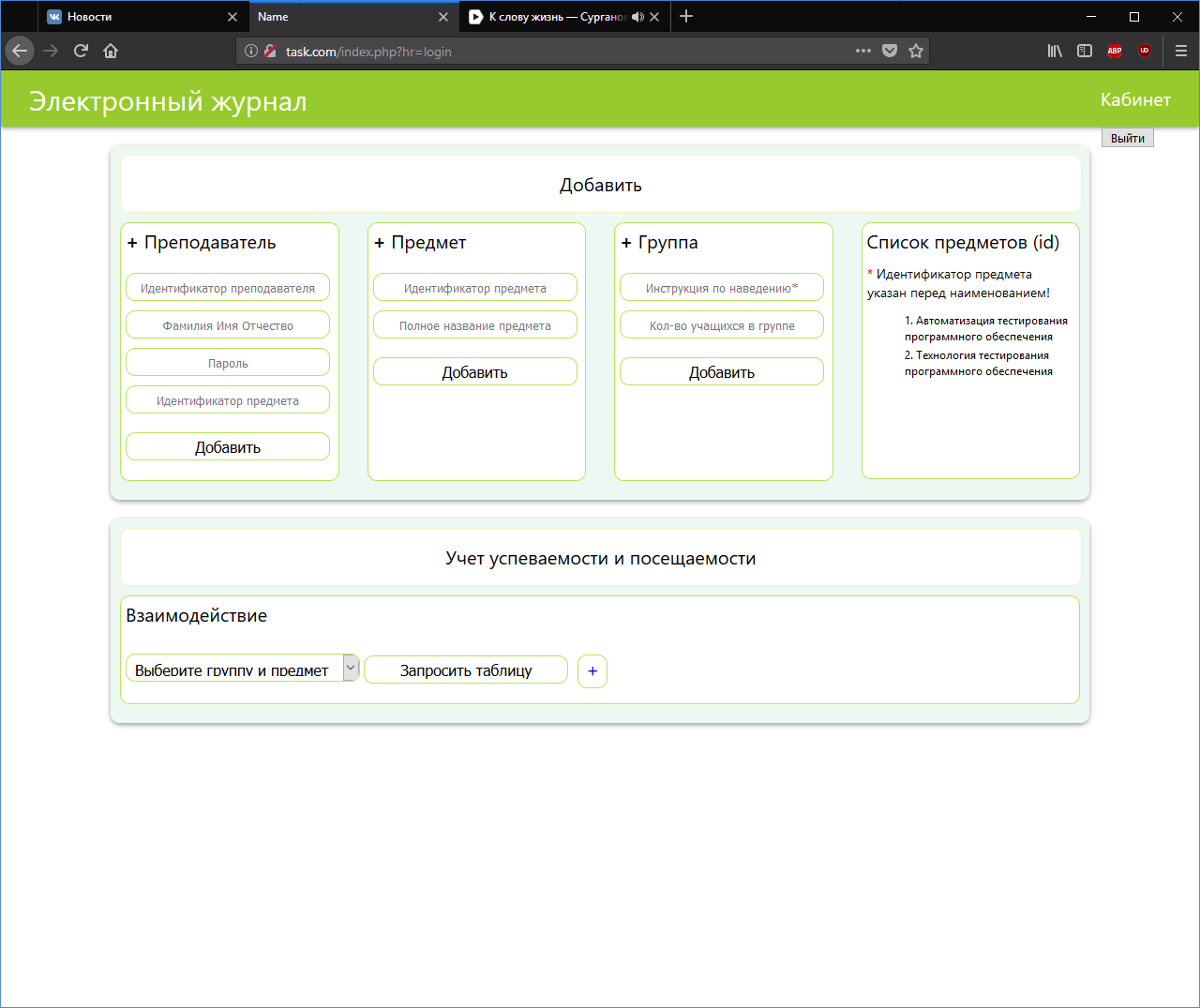


Рисунок 2.5 – Модуль добавления предмета

1. Поле "Идентификатор предмета". Доступен ввод только 2 символов (цифр)

2. Полное название предмета. Доступен ввод символов.

3. Общее правило для отправки, указанное в модуле «Личный кабинет».

**2.2.6 Модуль «Добавить группу»**

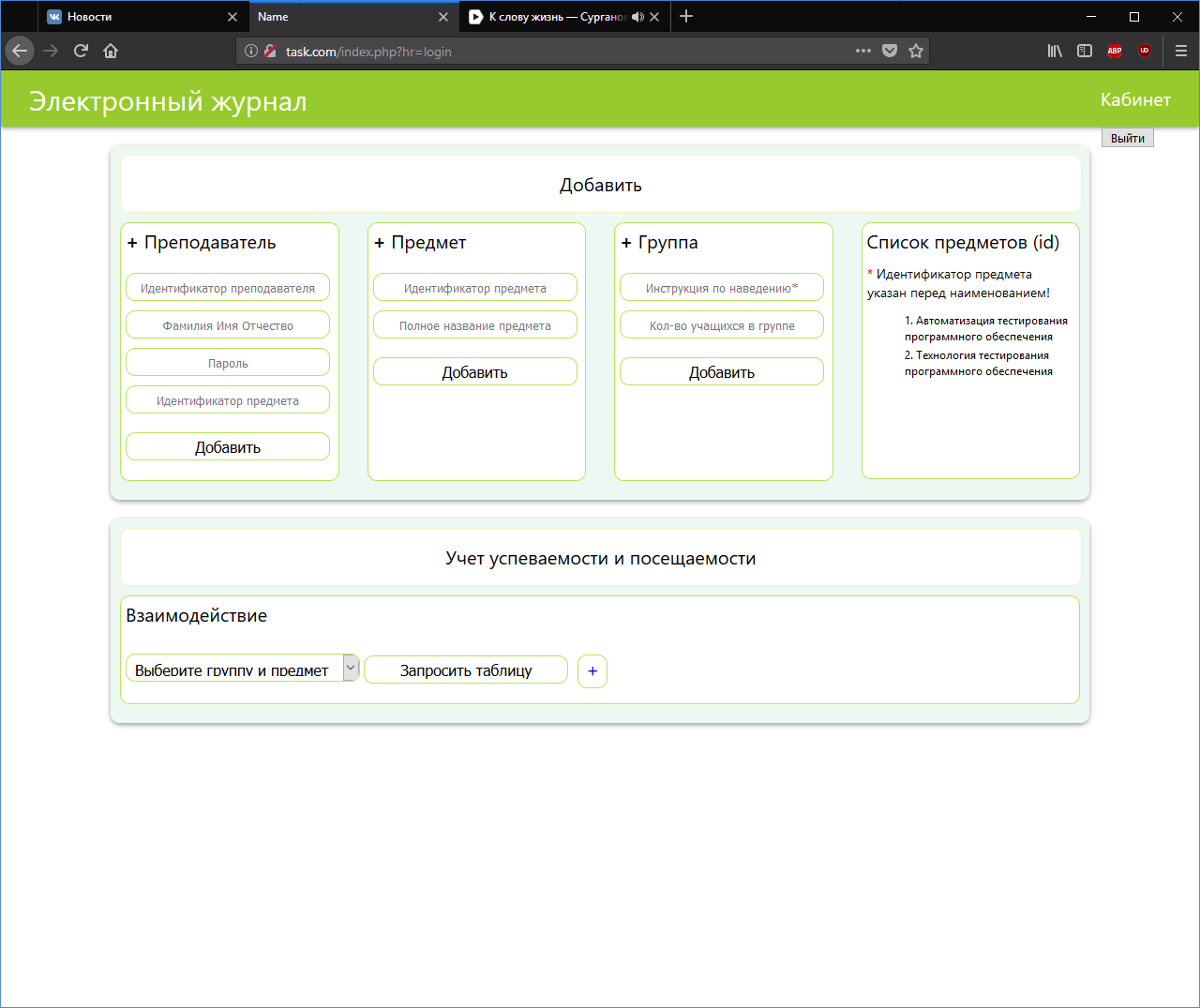


Рисунок 2.6 – Модуль добавления группы

1. Поле "Идентификатор группы". Инструкция заполнения поля показывается по наведению. Полное название предмета. Доступен ввод символов в форме «GROUP\_SUBJECT».

2. Поле "Кол-во учащихся в группе". Появляются ячейки, в зависимости от кол-ва записей (строк) исходной таблице.

3. Общее правило для отправки, указанное в модуле «Личный кабинет».

**2.2.7 Модуль «Список предметов (id)»**

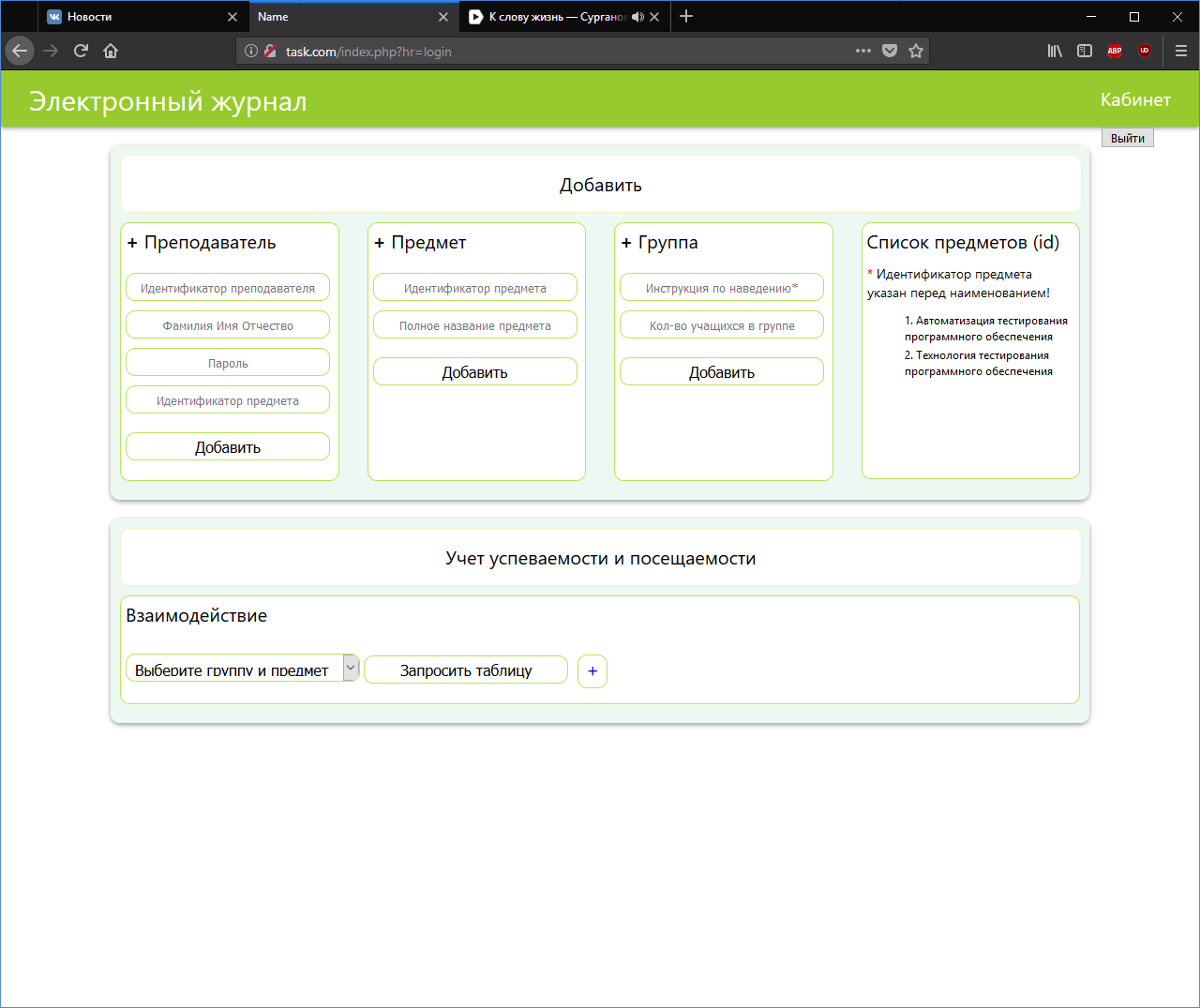


Рисунок 2.7 – Модуль список предметов

1. Добавить предмет можно в модуле "Добавить предмет".

2. Отображение предмета должно происходить сразу же, после добавления его в модуле "Добавить предмет".

3. Соответствие id наименованию предмета. Проверка через БД.

**2.2.8 Личный кабинет. Группа модулей «Взаимодействие»**

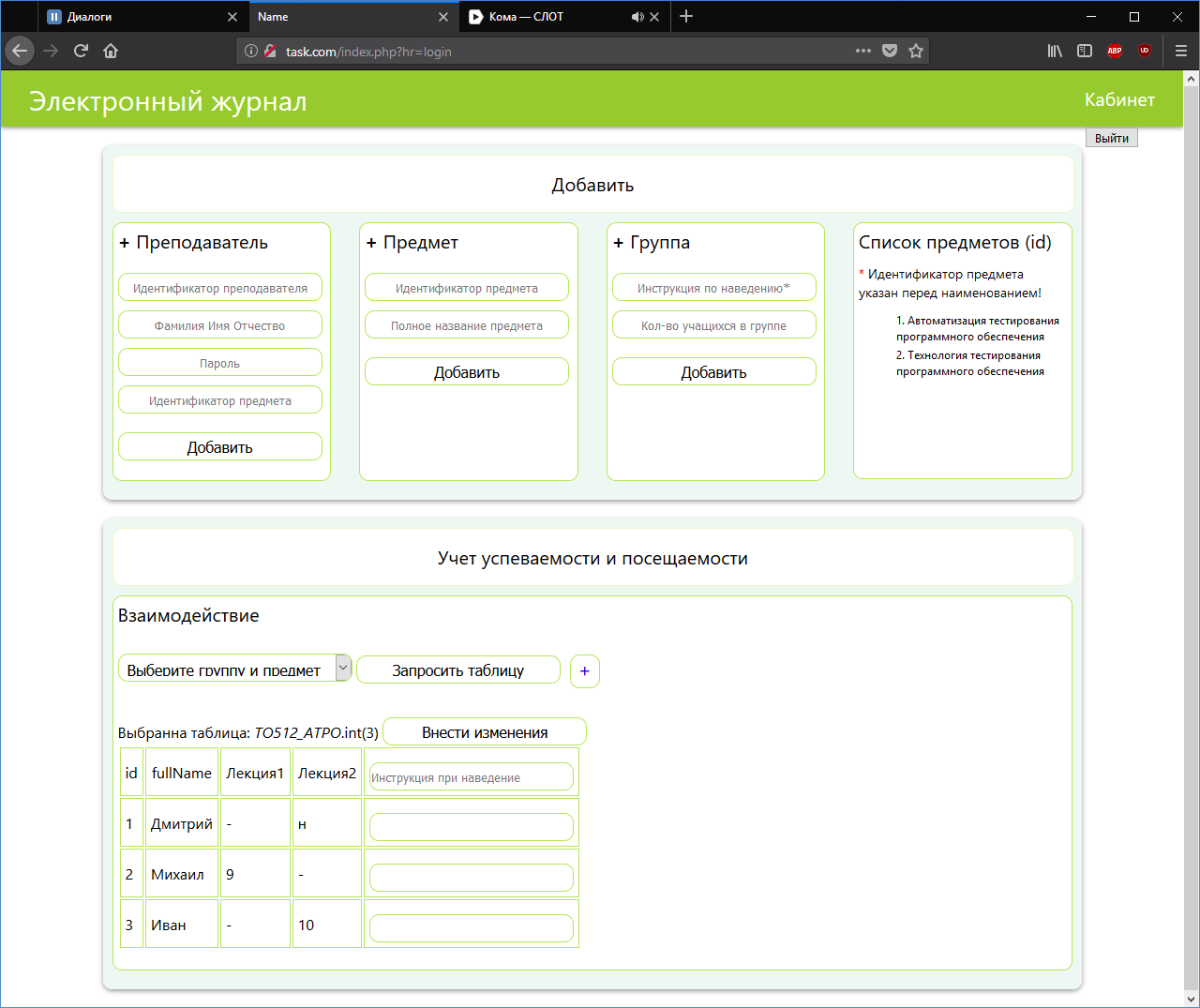


Рисунок 2.8 – Группа модулей «Взаимодействие»

1. Выбор таблиц и запрос ее. В случае выбора таблицы «Subject» или «teacher» - вывод сообщения о невозможности ответа.

2. Проверка редактирования таблицы, по средству нажатия на кнопку с «+», занесение в появившиеся ячейки значений и отправка по средству нажатия на «Внести изменения». По итогам появляется сообщение.

3. В случае, если поле было не заполнено, то на его месте должен появиться символ «-».

4. Название полей «Наименования занятия» должно заполняться строго по инструкции, которая будет появляться при наведении. В название поля должны отсутствовать пробелы.

**2.2.9 Выход с личного кабинета**

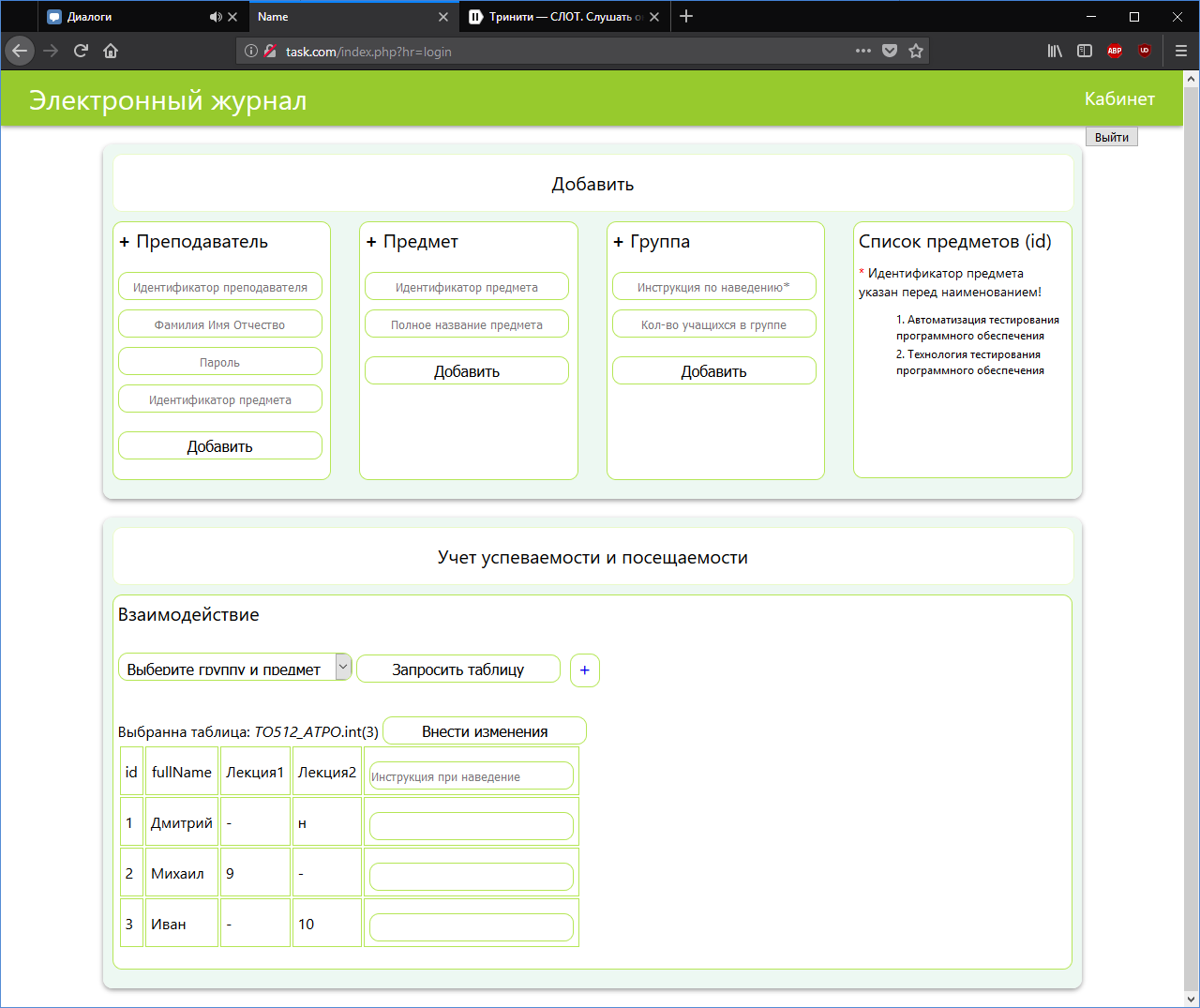


Рисунок 2.9 – Выход с личного кабинета

1. Нажатие на кнопку «выйти» должно привести к действию, после которого нас перенаправит на страницу авторизации.

## **2.3 Описание основных функций разработанного Web-приложения**

*Index.php*

Между тегами head заключены элементы наименования сайта во вкладке, подключение файлов JavaScript и CSS, а так же кодировка страницы и библиотека JQuery.

Тег body хранит отображаемую часть web-приложения.

Если пользователь авторизирован – выдоится текст-ссылка «Кабинет», если нет – «Авторизация»

<?php

if ($\_SESSION['name'] == ""){

echo '<a href="index.php?hr=login">Авторизация</a>';

} else {

echo '<a href="index.php?hr=login">Кабинет</a>';

echo'<form action="" method="post">

<input type="submit" name="do\_exite\_session" value="Выйти">

</form>';

}

?>

В случае, если пользователь нажимает на «Авторизация», то код подгружает страницу с авторизацией, в случае нажимает на «Кабинет» - подгружается личный кабинет.

<?php

if ($\_GET["hr"] == "login"){

require\_once 'login.php';

}

else if ($\_GET["hr"] == ""){

require\_once 'head.php';

}

?>

*Login.php*

Отвечает за загрузку формы авторизации. Производит проверку на волидность ввода значений в поля.

*Head.php*

Страница, предоставляющая не авторизированным пользователям просматривать успеваемость и посещаемость учащихся без внесения изменений в базу данных (далее – БД)

*Lk.php*

Страница, предоставляющая авторизированным пользователям возможность внесения изменений таблиц в БД, создание и редактирование их.

*Code.php*

Файл, содержащий функции для взаимодействия с БД и ее таблицами.

# 3 ОПИСАНИЕ ТЕСТИРОВАНИЯ РАЗРАБОТАННОГО WEB-ПРИЛОЖЕНИЯ И ЕГО РЕЗУЛЬТАТЫ

Предпочтение было отдано тестированию вручную. В силу небольшого размера самого Веб-приложения данный вид тестирования подходит лучше всего. Объектами тестирования выступали модули разработанного Веб-приложения и баги были выявлены в следствии взаимодействия с ними. Тестирование было произведено дважды по одному сценарию с целью устранения погрешностей в лице человеческой невнимательности. В ходе интенсивного взаимодействия с пользовательским интерфейсом были выявлены следующие дефекты:

Таблица 1 – Результаты тестирования

|  |  |
| --- | --- |
| Тестируемый модуль | Ошибка |
| 1 | 2 |
| Авторизация поле «идентификатор»;  Модуль «+ группа», поле «Идентификатор предмета» и «Идентификатор преподавателя»  Модуль «+ предмет», поле «Идентификатор предмета»  Модуль «+ група», поле «Кол-во учащихся в группе» | Допускается ввод букв, спец. Символов и иных символов. |
| Модуль «+ группа», поле «добавить группу» | Отсутствует допустимая форма ввода. |
| Группа модулей «Взаимодействие» при внесение изменений в таблицу: добавление пары. | Отсутствует форма для ввода занятия. В случае указания пробела форма сервер его не убирает. |

# ЗАКЛЮЧЕНИЕ

Сегодня каждый крупный разработчик должен помнить, что, забывая про тестирование программного обеспечения, или же, просто не обращая на него внимания, он фактически обрекает свой продукт на убытки. Ошибки всегда будут присутствовать в любом продукте, ведь человек не идеален. Но с помощью тестирования большую их часть можно избежать. Именно поэтому, тестирование и позволяет разработчикам быть уверенными в собственном программном обеспечении.

В процессе выполнения данного курсового проекта были изучены теоретические сведения в области тестирования программного обеспечения, было разработано и протестировано Web-приложение.

Перед приступлением к практической части были изучены теоретические сведения. Я ознакомился с понятием тестирования, его основными методами, а также с актуальностю его проведения в современном мире.

В процессе разработки Web-приложения были приобретены навыки по работе с такими языками программирования, как: PHP, CSS, HTML и JavaScript. При разработке Web-приложения все поставленные цели, такие как: создание возможности добавления пользователя, создание списка группы и авторизации были выполнены. Также были выполнены основные поставленные задачи.

Было выполнено тестирование разработанного Web-приложения. Тестирование было произведено в ручную, а его результаты были занесены в соответсвующую таблицу.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Фатон, Р. Тестирование программного обеспечения / Р. Фатон. — М. : Самс Паблишин, 2005. — 367 с.

2 Основы тестирования программного обеспечения [Электронный ресурс] — Режим доступа: <http://training.qatestlab.com>. — Дата доступа : *23.03.2018*

3 Методы тестирования программного обеспечения [Электронный ресурс]. – Режим доступа : <http://juice-health.ru/program/software-testing/495-software-testing-methods> — Дата доступа : 23.03.2018.