

РАЗРАБОТКА ПРОГРАММЫ "ЛИЧНЫЙ КАБИНЕТ" ДЛЯ ПРЕПОДАВАТЕЛЕЙ БГУ ИМЕНИ АЛЕКУ РУССО

Александр ПАРАХОНЬКО, магистр,
Бэлцкий государственный университет имени Алеку Руссо,
преподаватель, Т.Л. имени «Н.В. Гоголя»

Abstract: *The article discusses the necessity of developing a program "Personal Area" for professors of "Alecru Russo" Balti State University and its possibilities. The paper describes the optimizations used, which affect the design of the program and its performance.*

Keywords: *program, design, professor, component.*

1. Введение

Работа преподавателя очень сложная и ответственная. Она требует не только мастерства и высших знаний для преподавания некоторой дисциплины, но и времени и сил для составления, заполнения специальных документов. В БГУ имени «Алеку Руссо» уже не первый год преподаватели используют компьютерную программу «Notare studenti», разработанную департаментом «Компьютерных технологий», для выставления семестровых и годовых оценок студентам. Эти оценки собираются, хранятся и обрабатываются программой «Decanat». В этом году должна бы-ла «выйти в свет» программа для составления планов преподавателей и отчета, так необходимого кафедрам. В этой связи следует отметить две закономерности: 1 – с каждым годом растёт информатизация БГУ имени «Алеку Руссо» и 2 – рост числа программ с разными стилями дизайна, которые никак не связаны друг с другом. Поэтому была предложена идея объединить растущее количество программ, внедрить один стиль дизайна и сделать всё это максимально удобным для пользо-вателя – преподавателя. К тому же такая практика в нашем университете уже была внедрена для студентов в виде программы «Student».

Это и послужило предпосылками для разработки программы «Личный кабинет» – «Cadru Didactic».

2. Дизайн

В качестве примера был взят дизайн программы «Student», который был дополнен скрытой боковой панелью содержащей список всех компонентов программы (Рис. 1).



Рис. 1. Дизайн программы «Cadru Didactic».

Цветовое оформление программы включает два стиля: классика – белый фон и черный цвет текста (Рис. 2) и контраст – черный фон и белый цвет текста (Рис. 3) для тех, кто привык к стандартным типам дизайна, и тех, кто хочет попробовать что-то новое.



Рис. 2. Классический дизайн программы «Cadru Didactic».

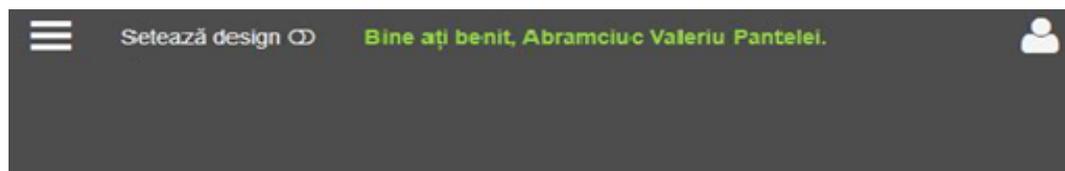


Рис. 3. Контрастный дизайн программы «Cadru Didactic».

Начальная веб-страница программы (Рис. 4) содержит «ободряющую» картинку, которая бы смогла снять некоторое напряжение преподавателя перед работой с программой.



Рис. 4. Начальная страница программы «Cadru Didactic».

Для устранения эффекта «загружающейся» веб-страницы был разработан механизм, который скрывает все подгружающиеся элементы и отображает пользователю в это время индикатор загрузки, который прекрасно подходит под дизайн программы (Рис. 5).

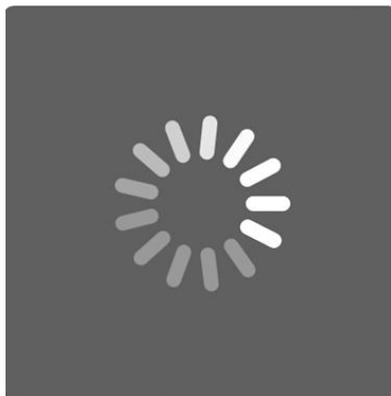


Рис. 5. Индикатор загрузки страницы.

Другим важным элементом дизайна является адаптивность программы к любому устройству. Это значит, что данной программой (некоторыми её компонентами) можно будет пользоваться прямо с мобильного телефона (планшета, смартфона и т.д.)

3. Компоненты программы «Cadru Didactic»

На данный момент программа содержит пять компонентов:

1. «Planuri» – раздел, который предназначен для создания личного учебного плана на текущий учебный год (Рис. 6 и 7). В этом разделе преподаватели должны выбрать из предложенных списков опорные пункты, которые они планируют осуществить, например: опорный пункт «книги с национальным ISBN» требует указания количества часов, ISBN, имена авторов, год публикации, типографию и т.д.

Table data from the screenshot:

Activitatea	Coefficient	Cantitatea planificată	Cantitatea efectuată	Ore planificate	Ore efectuate	Șterge item
Supravegherea evaluărilor curente (probe scrise) : 0,5 ore pentru un student, în cazul orelor individuale asistate de profesor	6.00	10	12	5.00	6.00	[Trash icon]
Modernizarea unei lucrări de laborator în acțiune : lucrare cu caracter de cercetare	5.00	0.13	0.17	4.00	5.00	[Trash icon]

Рис. 6. Компонент «Planuri». Классический дизайн.

Activitatea	Coefficient	Cantitatea planificată	Cantitatea efectuată	Ore planificate	Ore efectuate	Șterge item
Supravegherea evaluărilor curente (probe scrise) : 0,5 ore pentru un student, în cazul orelor individuale asistate de profesor	6.00	10	12	5.00	6.00	
Modernizarea unei lucrări de laborator în acțiune : lucrare cu caracter de cercetare	5.00	0.13	0.17	4.00	5.00	

Salvează

Рис. 7. Компонент «Planuri». Контрастный дизайн.

2. «Orar» – раздел, предназначенный для более удобного использования электронным расписанием с расширенными функциями. На сегодняшний день электронное расписание носит общий характер, т.е. имеет один и тот же набор функций как для преподавателей, так и для студентов, что вносит некоторые неудобства работы с программой. Новый подход к этому вопросу позволит не только специализировать расписание под конкретного преподавателя, но обогатить его следующими компонентами:

- автоматический выбор текущего дня и урока по времени и дате;
- удобный режим просмотра расписания по дням и по неделям;
- возможность выбора критерия просмотра расписания, к примеру, только по средам, только для какой-то группы и т.д.;
- быстрый переход на расписание определённой группы;
- автоматические уведомления при изменениях расписания через электронную почту и «Push уведомления» (Рис. 8).



Рис. 8. Пример некоторого «Push уведомления».

- возможность использования через телефон.

3. «Ordine» – раздел с перечнем всех приказов, касающихся преподавателя с возможностью полного просмотра и печатью;

4. «Notare Studenti» – раздел для оценивания студентов;

5. «Notificare» – раздел создания/ получения уведомлений. Такие уведомления, как и в случае с расписанием, будут распространяться через электронную почту и

«Push уведомления». На данный момент все новости публикуются на сайте университета или отправляются лично каждому преподавателю на почту. Это очень непрактично, поскольку лишь малое число преподавателей следят за всеми новостями на данном сайте. Раздел «Notificare» во много раз превосходит все указанные средства передачи уведомлений, поскольку ни один из них не обладает такой объёмной базой данных, как он. Более того, его режимы работы будут полезны всем пользователям программы. Планируется использовать три режима работы:

- a. простое уведомление – предоставляет возможность создания простого текстового сообщения с указанным периодом действия для всех пользователей (Рис. 9);

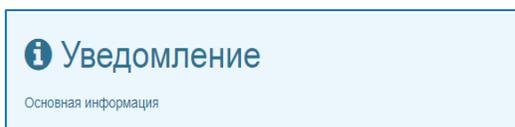


Рис. 9. Пример «простого» уведомления.

- b. приглашение – позволяет пригласить пользователя принять участие в некотором мероприятии, аналог Google – формам (Рис. 10); в отличие от предыдущего режима работы данные уведомления будут содержать кнопку «Принять участие» и возможность ввести некоторую дополнительную информацию, хотя данные о кафедре и факультете преподавателя можно будет получить по умолчанию. Это особенно полезно для организации круглых столов, практических семинаров и т.д.;

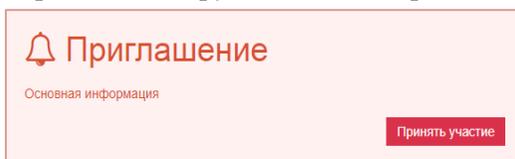


Рис. 10. Пример уведомления типа «приглашение».

- c. сообщение – аналог СМС сообщения, применяемый с целью простого уведомления конкретных преподавателей о необходимости подготовить ряд некоторых документов (Рис. 11), т.е. служит некоторым коротким напоминанием и возможностью выбрать конкретных адресатов из базы данных с фильтрами: кафедра, факультет.

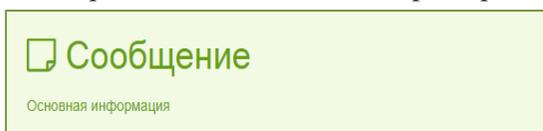


Рис. 11. Пример уведомления типа «сообщение».

4. Оптимизация

Для создания такого огромного проекта нельзя было обойтись без оптимизации, которая затронула все элементы программы. Разбор этих улучшений будем рассматривать на различных уровнях.

На уровне базы данных получение оптимального режима работы была построена соответствующая структура базы данных со всеми её связями и типами данных. Для увеличения скорости работы с данными были созданы хранимые процедуры [1, с. 1-6]. ХП хранятся и запускаются непосредственно на сервере. Они могут быть предварительно скомпилированы и проанализированы на сервере базы данных.

Это существенным образом отличается от обработки того же запроса со стороны клиента, где запрос разбирается драйверами базы данных, анализируется и оптимизируется (если это возможно) каждый раз, когда вызывается оператор запроса.

Это похоже на исполнение интерпретируемого языка (на стороне клиента) и компилируемого языка (на стороне сервера базы данных). А мы знаем, что скомпилированная программа будет работать быстрее.

Следующим уровнем оптимизации является сервер, а именно программный уровень – среда php, а не параметры и настройка компьютера-сервера. Программа личного кабинета построена на фреймворке «Laravel», который по умолчанию проводит множество оптимизаций, таких, как кэширование http-запросов, запросов к базе данных, сборку мусора, а также обеспечивает повышенную веб-безопасность посредством кодирования файлов cookie, CSRF-защиту, валидацию данных [2, с. 1-6].

На уровне клиента оптимизация обеспечивается посредством минимизации (компрессии) всех файлов Javascript и CSS, кэширования файлов и экономичному использованию кода выше упомянутых файлов.

5. Вывод

Разработка программы личного кабинета для преподавателей «Cadru didactic» позволит значительно упростить и сделать приятной выполнение своих обязанностей. Появление данной программы позволит нашим преподавателям всегда быть в курсе важных новостей, изменения расписания, ввод оценок и многое другое. Данный проект является инновационным не только благодаря внедренным идеям, но и технологиям, которые делают данную программу более надёжной, производительной и удобной.

Библиография:

1. TAYLOR, Ren. *Stored Procedures in MySQL and PHP*, 2014. (Электронный ресурс). Режим доступа: <https://www.sitepoint.com/stored-procedures-mysql-php/>
2. KAZANKOV, Vladimir. *The top 10 advantages of using laravel php framework*, 2017. (Электронный ресурс). Режим доступа: <https://belitsoft.com/laravel-development-services/10-benefits-using-laravel-php-framework>