

Санкт-Петербургский государственный университет
Фундаментальные информатика и информационные технологии
Информационные технологии

Клёнская Анастасия Александровна

Оптимизация личного кабинета абитуриента СПбГУ

Бакалаврская работа

Научный руководитель:
к. ф.-м. н., доцент Михайлова Е. Г.

Рецензент:
начальник отдела разработки и сопровождения ИС «Обучение»
Управления – службы инф. технологий СПбГУ
Павел Алексеевич Карпенко

Санкт-Петербург
2016

SAINT-PETERSBURG STATE UNIVERSITY

Fundamental Informatics and Information Technology
Information Technology

Anastasia Klyonskaya

Personal profile optimisation of St.
Petersburg University applicant

Bachelor's Thesis

Scientific supervisor:
Associate Professor Elena Mikhailova

Reviewer:
Head of the development and maintenance of IS "Education" Department
- St. Petersburg State University of Information Technology Services
Pavel Karpenko

Saint-Petersburg
2016

Оглавление

Введение	4
1. Цели и задачи	6
2. Описание личного кабинета поступающего	7
2.1. Описание программной реализации	8
2.2. Описание базы данных	9
3. Подходы для проведения реинжиниринга	10
3.1. Введение	10
3.2. Этапы реинжиниринга	10
3.3. Заключение	12
4. Реинжиниринг личного кабинета	13
4.1. Формализация претензий к системе	13
4.2. Анализ структуры системы	14
4.3. Альтернативы реинжиниринга системы	15
4.4. Реализация выбранной альтернативы	16
4.5. Обзор старой и новой версий экрана	18
5. Инструкция по внедрению в использование	21
Заключение	22
Список литературы	23

Введение

Ни для кого не секрет, что компьютеры и другие вычислительные устройства стали неотъемлемой частью жизни человека. Современные технологии позволяют не только общаться людям с разных уголков планеты, но и проводить сложные операции [8] с помощью очков дополненной реальности [6] или заменять органы или части тела аналогами, напечатанными на 3D-принтере [5]. Количество всевозможных программных комплексов неукоснительно растёт с каждым днём, также как и стремительно движется научно-технический прогресс; изменяются и требования к существующим системам. К тому же, не всегда сразу удаётся удовлетворить всем потребностям и желаниям покупателей, они постоянно меняются, при неизменных высоких требованиях к качеству продукта.

В связи с этим, актуальна проблема реинжиниринга, модернизации или перепроектирования программного обеспечения согласно новым требованиям пользователей. Разумеется, для принятия решения недостаточно лишь критики со стороны потребителей, необходимо учитывать множество факторов, начиная с денежной стороны вопроса и заканчивая вопросами, связанными с проблемами перехода со старой системы на новую. К сожалению, количество современной литературы, освещающей данную проблему, сравнительно невелико.

Темой данной бакалаврской работы стала "Оптимизация личного кабинета абитуриента СПбГУ". Сайт можно найти по адресу "<https://cabinet.spbu.ru>". Главной функцией кабинета является возможность подавать заявления и соответствующие документы для поступления по программам среднего профессионального образования, бакалавриата, магистратуры и аспирантуры, а также для переводов внутри СПбГУ, восстановлений и других действий, связанных с поступлением. Благодаря нему не только облегчается процесс поступления абитуриентов, которые живут далеко от города, но и значительно экономится время.

После детального исследования сайта мною были обнаружены неко-

торые проблемные области, которые необходимо модернизировать для улучшения не только удобства использования, но и для более корректной обработки загружаемого массива данных.

1. Цели и задачи

Целью данной работы является модернизация личного кабинета абитуриента СПбГУ для достижения более удобной работы с ним.

Для реализации заявленной цели были поставлены следующие задачи.

- Изучить методы проведения реинжиниринга
- Разработать прототип, устраняющий недостатки личного кабинета
- Реализовать прототип
- Составить инструкции для корректного внедрения в использование.

2. Описание личного кабинета поступающего

Личный кабинет поступающего расположен по адресу ["https://cabinet.spbu.ru"](https://cabinet.spbu.ru). Он предназначен для подачи заявлений для участия в конкурсе

- на образовательные программы общего образования,
- на образовательные программы подготовки специалистов среднего звена по специальностям среднего профессионального образования,
- на образовательные программы бакалавриата и специалитета,
- на образовательные программы магистратуры,
- по программам подготовки научно-педагогических кадров в аспирантуре,
- на образовательные программы ординатуры [10].

Кроме того, заявления принимаются не только на основной приём, но и для перевода внутри СПбГУ, восстановления или перевода в СПбГУ.

Для того чтобы подать заявление необходимо прежде всего зарегистрироваться в кабинете, подтвердить правильность электронной почты и зайти в кабинет с указанными данными при регистрации. Следующим шагом является заполнение анкеты согласно пунктам: личные данные, паспорт, контактная информация, образование, дополнительные данные, опыт работы, родители и льготы. После того как анкета заполнена появляется возможность в зависимости от уровня образования подать заявление для участия в интересующем конкурсе.

При подаче заявления указывается форма и основа обучения, выбираются направления и образовательные программы. Для каждого уровня поступления (например, магистратура, аспирантура) создаётся

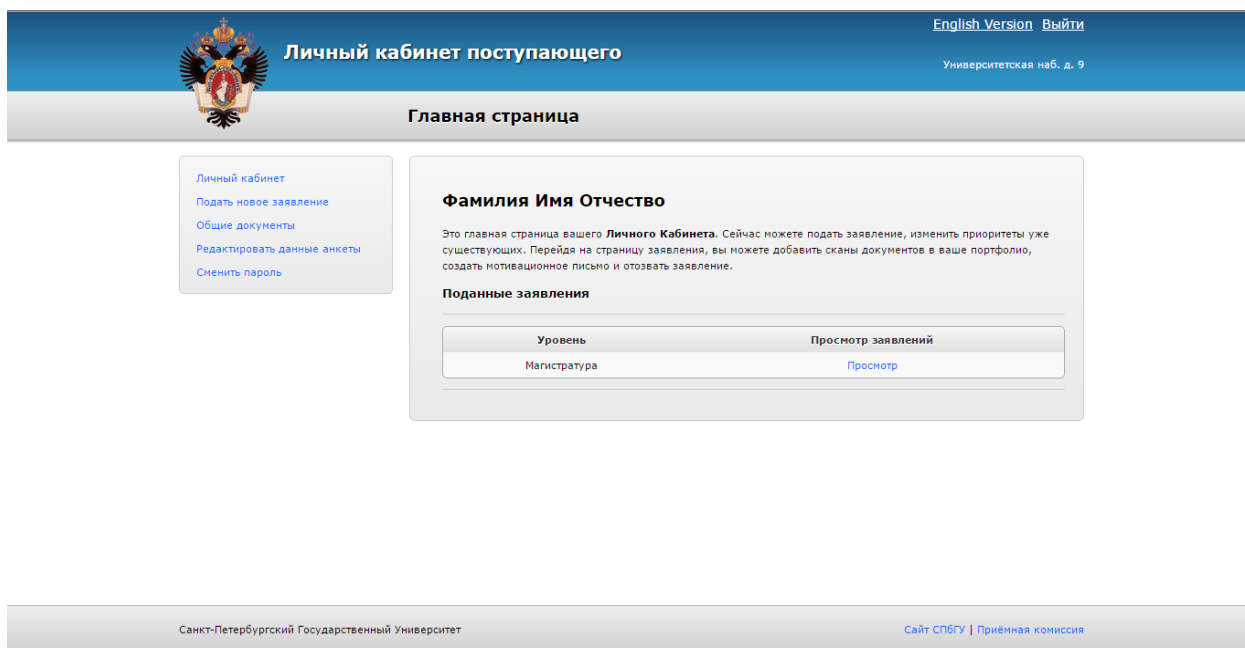


Рис. 1: Пример главной страницы после заполнения анкеты и подачи заявления для участия в конкурсе на образовательные программы магистратуры

только одно заявление, в котором указаны все желаемые направления и образовательные программы и их приоритет.

Документы, необходимые для поступления, могут прикрепляться к направлению, если это необходимо для данного уровня, либо непосредственно к конкретному заявлению, либо без привязки к нему во вкладке "Общие документы".

В кабинете также можно сменить пароль и редактировать данные анкеты. На рис. 1 представлен один из возможных экранов.

2.1. Описание программной реализации

Проект написан на платформе ASP.NET MVC 4 [2]. В её основе лежит взаимодействие трёх компонентов: контроллера(controllers), модели(models) и представления(views), каждый из которых выполняет свои специфические функции. Рассмотрим компоненты подробнее.

Модель – это набор классов, отображающих логику используемых данных.

Представление – это интерфейс приложения - например, html-страница, через которую пользователь, открывший данную страницу, взаимодействует с веб-приложением.

Контроллер представляет класс, с которого именно и начинается работа приложения. Он связывает между собой модель и представление. Получая данные, идущие от пользователя, контроллер согласно своей внутренней логики, если это требуется, обращается к модели и на её основе создаёт соответствующее представление.

Таким образом, подобное разделение позволяет легче поддерживать и модифицировать приложение [1].

В проекте используются такие языки, как C#, JavaScript, HTML, CSS, SQL кроме того задействована библиотека JavaScript – jQuery.

Согласно расчёту метрик, произведённому с помощью средств среды разработки Visual Studio, в решении содержится 27 775 строк кода, в том числе на контроллеры приходится 20 671 строк кода. Строками кода, в данном случае, не считаются пробелы, комментарии, скобки, объявление типов и пространств имён [3].

2.2. Описание базы данных

Для того чтобы локально запустить проект была развёрнута и подключена база данных. База состояла из 86 таблиц для хранения информации о данных пользователя, поданных заявлениях, прикреплённых документах, а также о данных для заполнения различных полей на сайте.

3. Подходы для проведения реинжиниринга

3.1. Введение

Реинжиниринг – сложный процесс, нацеленный на улучшение существующей системы, и для его успешного проведения необходимо иметь методологическую основу.

В настоящее время существует много сходных и смежных понятий, таких как модернизация, адаптация, рефакторинг, обратный инжиниринг, редизайн и другие. Они раскрывают понятие «реинжиниринг», а соотносимая с ними деятельность рассматривается либо как одна из форм реинжиниринга, либо как подпроцесс процесса реинжиниринга.¹

Согласно [7] понятие «реинжиниринг информационных систем» различными исследователями до сих пор трактуется по-разному, существует множество близких понятий, наличие которых приводит к появлению внешне отличающихся, но по сути схожих методов и технологий.

Поэтому, в связи с отсутствием общей методологии для проведения реинжиниринга, конкретнее определим начальные условия. В данной работе будем рассматривать уже выпущенную и работающую некоторое время в реальных условиях систему, к которой существуют известные претензии.

3.2. Этапы реинжиниринга

Опираясь на [9], были выделены следующие этапы для проведения реинжиниринга личного кабинета.

1. Формализация претензий к системе;
2. анализ структуры системы;
3. генерация альтернатив реинжиниринга системы;

¹Далее понятие "модернизация" будем считать синонимом реинжиниринга

4. выбор оптимальной альтернативы;
5. реализация выбранной альтернативы.

На первом этапе производится сбор и обработка претензий. Они могут поступать как от непосредственных пользователей системы, так и от руководства. Это могут быть замечания, касающиеся пользовательского интерфейса, быстродействия, неполноты функциональности и много другого. Недостатки системы можно выявить посредством анкетирования, опросов, наблюдения за работой пользователей, карты кликов и т.п. После сбора информации, необходимо её обработать и формализовать для выявления общих характеристик, которые помогут найти слабые места.

На втором этапе нужно исследовать структуру системы, то есть строение кода, базы данных, их взаимосвязь; построить модели, необходимые для полного понимания архитектуры всего проекта. Это важно не только для разработчиков, которым придётся работать с чужим кодом, но и для людей, которые сами принимали участие в написании данного программного обеспечения.

На следующем этапе порождается множество решений проблем, формализованных на первом этапе. Способы генерации можно разделить на три категории. К первой категории относятся решения, относящиеся к изменению системы в целом; ко второй – частные решения; к третьей – применение различных шаблонов. Первый способ наиболее трудно реализуемый на практике, для этого требуются высоко квалифицированные разработчики, способные учитывать и охватывать всевозможные нюансы системы. Второй способ предполагает разработку локальных улучшений. К последней категории относятся решения с применением уже существующих шаблонов, которые корректируются с учётом условий конкретного проекта.

На четвёртом этапе выбирается наиболее оптимальная альтернатива. Критерии оптимальности для каждой задачи выбираются индивидуально, особенно часто используемые критерии: стоимость, время, сложность реализации, насколько хорошо решаются выявленные про-

блемы. При проведении сравнения следует учитывать различные риски, так как запланированные оценки могут на практике отличаться от реальных. Одним из возможных исходов на этом этапе может стать решение о написании системы с нуля, если этот вариант в наибольшей степени удовлетворяет выбранным критериям.

После выбора альтернативы на последнем этапе происходит его реализация. Прежде всего необходимо составить технический проект, подробно описывающий необходимые технические решения. Это обусловлено тем, что при реализации в зависимости от сложности системы возникает ряд проблем, таких как переход со старой версии на новую, обучение пользователей и другие. Для успешного внедрения в использование все эти факторы надлежит учитывать.

3.3. Заключение

Таким образом, процесс реинжиниринга связан с определёнными рисками и для принятия решения о его проведения следует всесторонне рассмотреть этот вопрос. Одним из важных факторов является хорошо обученный, высоко квалифицированный персонал, поскольку не каждый сможет не только реализовать новые требования, но и не испортить уже существующую функциональность.

4. Реинжиниринг личного кабинета

На основе шагов, сформулированных в предыдущей главе, проведём модернизацию личного кабинета, описанного в главе 2.

4.1. Формализация претензий к системе

Прежде всего формализуем претензии к системе. Моим научным руководителем была описана проблема: в личном кабинете пользователям нужно при аналогичности необходимых документов для поступления на несколько направлений загружать документы несколько раз для каждого направления по отдельности. Зачастую из-за невнимательности или других причин абитуриенты загружают лишь один экземпляр документа, что понижает их баллы. Данный недостаток не только делает неудобной процедуру подачи заявлений, но и значительно затрудняет проверку документов членами приёмной комиссии.

Подробно проанализировав сайт и его функции, мною были отмечены пункты.

- Создаётся одно заявление на уровень.
- Для каждого уровня можно задавать несколько направлений.
- Документы привязываются либо к конкретному заявлению(уровню), либо к направлению (если это допускает выбранный уровень), либо добавляются во вкладке "общие файлы"(общие для всех заявлений).
- Абитуриенты могут подавать документы на несколько разных уровней и направлений внутри них.
- Для подачи одинаковых документов на разные направления приходится дублировать их загрузку.
- Значительно затруднена проверка загруженных документов, так как документы с различной степенью привязки расположены на разных экранах.

Таким образом, процесс подачи заявлений усложняется, возникают некоторые проблемы.

4.2. Анализ структуры системы

В главе 2 была в основном описана структура системы и используемые технологии, поэтому в данном пункте рассмотрим детали, непосредственно касающиеся описанной выше проблемы.

Рассмотрим интерфейс личного кабинета. Как видно на рисунке 1, общие для всех документы загружаются во вкладке "Общие документы". Для того чтобы загрузить документы общие для одного заявления, нужно нажать кнопку "Просмотр" у соответствующего заявления, внизу экрана будет поле для загрузки. И, наконец, для конкретного направления документы загружаются в окне, открываемом после нажатия "Просмотр/Добавить файлы" желаемого направления.

Теперь обратимся к базе данных. Для хранения документов используются такие таблицы, как `dbo.PersonFile`, `dbo.ApplicationFile`. В первой таблице хранятся общие для всех заявлений документы, они привязываются напрямую к пользователю через атрибут `PersonId`. Во второй таблице хранятся документы общие для одного заявления и для конкретного направления. Информация о выбранных направлениях находится в таблице `dbo.Application` и связывается в одно заявление с помощью атрибута `CommitId` и таблицы `dbo.ApplicaionCommit`. Таким образом, при привязке к конкретному направлению в `dbo.ApplicationFile` заполняются соответствующими данными поля `ApplicationId` и `CommitId`, а при привязке к заявлению заполняется только `CommitId`.

Со стороны программной реализации выделим контроллер `AbiturientNewController`, модели содержащиеся в `MyModels.cs` и представление `AddSharedFiles`, отвечающие за работу экрана "Общие документы".

4.3. Альтернативы реинжиниринга системы

В этой главе объединены этапы генерации и выбора альтернативы вместе.

Для генерации альтернатив учитывались такие факторы как, сложность перехода со старой версии на новую, время реализации, трудоёмкость. Финансовая сторона вопроса не рассматривалась.

В связи с этим вариант изменения системы в целом не был выбран. В ходе поиска решения была проведена встреча с разработчиками. Путём уточнения деталей с научным руководителем и разработчиками, было принято решение изменения вкладки "Общие документы" и разработан прототип, который можно видеть на рисунке 2. Прототип создан посредством инструментов Moqups [4].

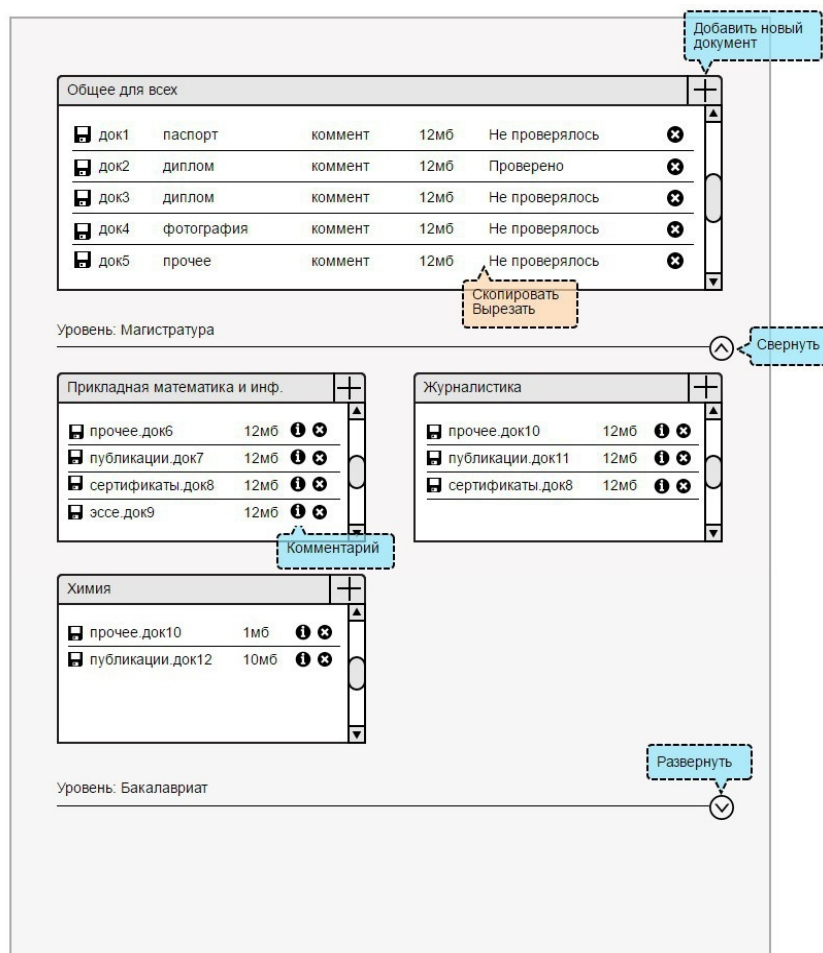


Рис. 2: Прототип нового экрана вкладки "Общие документы"

Согласно прототипу вместо загрузки одинаковых документов несколько раз, загружается один документ, а затем копируется в окно необходимого направления. Для этого нужно создать таблицу с ссылками на загруженный документ и заявление, чтобы избежать дублирования файла в базе данных. Преимуществом данного прототипа также является возможность видеть сразу все документы в одном месте, а не переключаться между несколькими вкладками. Загрузка документа происходит путём нажатия на символ "+" в углу каждого окна. Для каждого уровня создаётся новая область, которую можно скрыть при необходимости.

Таким образом, прототип решает описанную выше проблему. Учтя данный факт и другие преимущества, было принято решение реализовать данный прототип.

4.4. Реализация выбранной альтернативы

В ходе реализации прототипа была создана новая таблица `dbo.ReferenceOnFile`, код создания которой представлен в листинге 1. В ней хранятся ссылки на уже существующие файлы. То есть при копировании документа в окно другого направления в таблице создаётся строка с соответствующими файлу данными, `FileId` – идентификатор файла, `ApplicationId` – идентификатор нового направления, `CommitId` – идентификатор заявления, `PersonId` – идентификатор пользователя, `IsDeleted` – по умолчанию 0, после удаления ссылки – 1.


```

1 Create table ReferenceOnFile
2 (
3     Id uniqueidentifier NOT NULL,
4     FileId uniqueidentifier NOT NULL,
5     ApplicationId uniqueidentifier NOT NULL,
6     CommitId uniqueidentifier NOT NULL,
7     PersonId uniqueidentifier NOT NULL,
8     IsDeleted bit NOT NULL Default 0,
9     PRIMARY KEY (Id)
10 );

```

Listing 1: Инструкция для создания таблицы

Был изменён контроллер `AbiturientNewController`, а именно метод `AddSharedFiles()`, который стал на основе новой модели `AllFileModel` выводить документы согласно их принадлежности заявлению или направлению. Также создан новый метод `AddReferenceOnFile(string fileId, string applicationId, string commitId)`, который принимает на вход идентификаторы файла, направления и заявления и записывает новую строку в таблицу `dbo.ReferenceOnFile`.

Кроме того, изменён метод `DeleteFile(string id)` в `AbiturientController`, удаляющий файлы, привязанные к направлению. Добавлена проверка на ссылки на документы в таблице `dbo.ReferenceOnFile` для правильной работы удаления.

Соответственно было переписано представление `AddSharedFiles.aspx` и добавлены новые стили `mystyle.css` и изображения.

В процессе реализации прототипа было принято решение о незначительных его изменениях.

4.5. Обзор старой и новой версий экрана

В итоге реализации старый экран (рис. 3) заменился на новый (рис. 4).

Файл Файл не выбран

Настоятельно рекомендуем загружать каждый документ целым файлом в формате **PDF**.
Для загрузки фотографии можно использовать файлы JPG (JPEG, PNG).

Тип файла

Комментарий

Загруженные файлы:

Не отображать отказы от конкурса и другие файлы, защищенные от удаления

Просмотр	Имя файла	Тип файла	Комментарий	Размер	Статус	Удалить
	passport1.docx	Паспорт	(исходное название файла-Документ Microsoft Word.docx)	12,35 Kb	Не проверялось	
	drugoe1.docx	Прочие документы	(исходное название файла-Документ Microsoft Word.docx)	12,35 Kb	Не проверялось	

Информация для поступающих:

Рис. 3: Часть старого экрана вкладки "Общие документы"

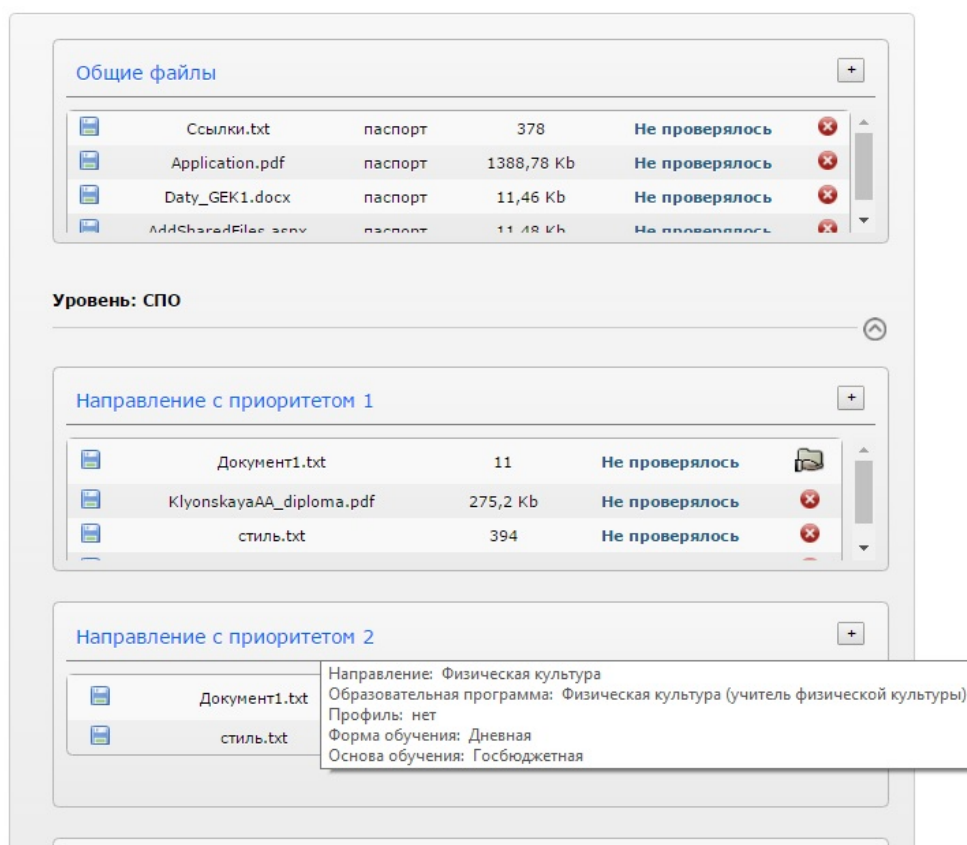


Рис. 4: Часть нового экрана вкладки "Общие документы"

Таким образом, интерфейс почти полностью преобразился. Теперь стало возможно копировать документы на другие направления, для этого нужно нажать на строку с документом, появится кнопка "Копировать", нажать на неё, затем выбрать окно с другим направлением щёлкнуть туда, появится кнопка "Вставить", нажать на неё, документ добавится на данное направление. Для загрузки нового документа нужно нажать "+" в углу окна появится форма (рис. 5). В окне "Общие файлы" находятся документы общие для всех заявлений. Файлы общие для одного заявления дублируются во всех окнах направлений данного заявления.

Также при создании нового заявления появляется новый блок с выбранными направлениями. Чтобы узнать подробную информацию о направлении, нужно навести на название окна (например: "Направление с приоритетом 2" рис. 4).

The image shows a web form for document upload. On the left, there are labels: 'Файл', 'Тип файла', and 'Комментарий'. To the right of 'Файл' is a button 'Выберите файл' and the text 'Файл не выбран'. Below 'Тип файла' is a dropdown menu with 'паспорт' selected. Below 'Комментарий' is a large empty text area. At the bottom left is a green button 'Отправить', and to its right is a blue link 'Закреть'.

Рис. 5: Форма для загрузки документов

Новый интерфейс не только решает описанную в 4.1 проблему, но и позволяет удобнее контролировать загружаемые документы, что в будущем уменьшит количество нежелательных ошибок.

5. Инструкция по внедрению в использование

Для того чтобы изменить существующий проект необходимо:

- в базе OnlinePriem2015 создать таблицу в соответствии с 1;
- добавить в проект классы AllFileModel, CommitFile, AppFile, ApplicationInfo в MyModels.cs;
- добавить в проект таблицу стилей mystyle.cs;
- добавить изображения для кнопки "свернуть";
- заменить представление AddSharedFiles.aspx на новое;
- в AbiturientNewController заменить метод AddSharedFiles() на новый и добавить метод
AddReferenceOnFile(string fileId, string applicationId, string commitId);
- в AbiturientController заменить метод DeleteFile(string id);
- при отправке данных на приложение членов приёмной комиссии делать выборку файлов с учётом новой таблицы.

Заключение

В ходе данной работы были выделены основные этапы проведения реинжиниринга системы. В результате соблюдения которых, был разработан и реализован прототип, повышающий удобство использования личного кабинета абитуриента СПбГУ. Составлены подробные инструкции изменения существующей системы для корректного внедрения в использование.

Таким образом, поставленные задачи и цели были достигнуты.

Список литературы

- [1] Metanit.com. Онлайн-книга Изучаем ASP.NET MVC 4. — 2015. — URL: <http://metanit.com/sharp/mvc> (дата обращения: 10.04.2016).
- [2] Microsoft. ASP.NET MVC 4. — URL: <http://www.asp.net/mvc/mvc4> (дата обращения: 10.04.2016).
- [3] Microsoft. New for Visual Studio 2008 – Code Metrics. — 2013. — URL: <https://blogs.msdn.microsoft.com/codeanalysis/2007/10/03/new-for-visual-studio-2008-code-metrics/> (дата обращения: 10.04.2016).
- [4] SRL Evercoder Software. Моqups. — URL: <https://moqups.com/>.
- [5] hobby3d.ru. 3D ПЕЧАТЬ И МЕДИЦИНА // hobby3d.ru. — 2015. — URL: <http://hobby3d.ru/news/3d-pechat-i-meditsina/> (дата обращения: 30.03.2016).
- [6] iCover. Пациент успешно перенес операцию на головном мозге, погрузившись в мир VR // Geektimes. — 2016. — URL: <https://geektimes.ru/company/icover/blog/271318/> (дата обращения: 30.03.2016).
- [7] Ахтырченко К.В., Сорокваша. Т.П. Методы и технологии реинжиниринга ИС. / Ed. by В.П. Иванников.
- [8] Носенкова Анастасия. Автоплан операции: интервью с разработчиками хирургической системы дополненной реальности СамГМУ // Голографика | дополненная и виртуальная реальность. — 2016. — URL: <http://holographica.space/articles/intervyu/autoplan-samgmu-2577> (дата обращения: 30.03.2016).
- [9] Реинжиниринг информационных систем // Пятифан. — 2013. — URL: <http://5fan.ru/wievjob.php?id=19283> (дата обращения: 16.04.2016).

- [10] СПбГУ. Личный кабинет поступающего. — URL: <http://spbu.ru/component/content/article?id=23369> (дата обращения: 10.04.2016).