

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет Компьютерных систем и сетей
Кафедра Программного обеспечения информационных технологий

К защите допустить:

Заведующая кафедрой ПОИТ

_____ Н. В. Лапицкая

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к дипломному проекту

на тему:

**ПРОГРАММНОЕ СРЕДСТВО СОЗДАНИЯ
ВЕБ-ПРИЛОЖЕНИЙ С ПОМОЩЬЮ ГОТОВЫХ
ГРАФИЧЕСКИХ КОМПОНЕНТОВ С
ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ WEBIX И ЯЗЫКА
ПРОГРАММИРОВАНИЯ JAVASCRIPT**

БГУИР ДП 1-40 01 01 01 ??? ПЗ

Студент

Н. А. Романчук

Руководитель

С. А. Медведев

Консультанты:

от кафедры ПОИТ

С. А. Медведев

по экономической части

В. А. Палицын

Нормоконтролёр

И. М. Марина

Рецензент

Минск 2019

СОДЕРЖАНИЕ

Введение	7
1 Анализ литературных источников и существующих аналогов . . .	10
1.1 Аналитический обзор литературных источников	10
1.2 Анализ существующих аналогов	10
1.3 Разработка и спецификации функциональных требований . .	15
2 Моделирование предметной области	17
3 Проектирование программного средства	18
4 Конструирование программного средства	19
5 Тестирование и проверка работоспособности ПС	20
6 Руководство по установке и использованию	23
7 Техничко-экономическое обоснование разработки и внедрения программного средства	24
7.1 Характеристика программного средства	24
7.2 Определение объема и трудоемкости ПС	24
7.3 Расчет сметы затрат	27
7.4 Оценка экономической эффективности применения ПС у пользователя	31
Заключение	35
Список использованных источников	36
Приложение А Фрагменты исходного кода	37

ОПРЕДЕЛЕНИЯ И СОКРАЩЕНИЯ

В настоящей пояснительной записке применяются следующие определения и сокращения.

Спецификация – документ, который желательно полно, точно и верифицируемо определяет требования, дизайн, поведение или другие характеристики компонента или системы, и, часто, инструкции для контроля выполнения этих требований [1].

ССДОАОАИИООИ – секта свидетелей драматического отличия аутентификации от авторизации и их обоих от идентификации.

ВВЕДЕНИЕ

С появлением Web-технологий компьютер начинают использовать совершенно новые слои населения Земли. Можно выделить две наиболее характерные группы, находящиеся на разных социальных полюсах, которые были стремительно вовлечены в новую технологию, возможно, даже помимо их собственного желания. С одной стороны, это были представители элитарных групп общества - руководители крупных организаций, президенты банков, топ-менеджеры, влиятельные государственные чиновники и т.д. С другой стороны, это были представители широчайших слоев населения - домохозяйки, пенсионеры, дети.

При появлении технологии Web компьютеры повернулись лицом к этим двум совершенно противоположным категориям потенциальных пользователей. Элиту объединяла одна черта – в силу высочайшей ответственности и практически стопроцентной занятости “большие люди” никогда не пользовались компьютером; типичной была ситуация, когда с компьютером работал секретарь. В какой-то момент времени они поняли, что компьютер им может быть полезен, что они могут результативно использовать то небольшое время, которое можно выделить на работу за компьютером. Они вдруг поняли, что компьютер - это не просто модная и дорогая игрушка, но инструмент получения актуальной информации для бизнеса. При этом им не нужно было тратить сколько-нибудь заметного времени, чтобы освоить технологию работы с компьютером (по сравнению с тем, как это было раньше).

Спектр социальных групп, подключающихся к сети Интернет и ищущих информацию в WWW, все время расширяется за счет пользователей, не относящихся к категории специалистов в области информационных технологий. Это врачи, строители, историки, юристы, финансисты, спортсмены, путешественники, священнослужители, артисты, писатели, художники. Список можно продолжать бесконечно. Любой, кто ощутил полезность и незаменимость Сети для своей профессиональной деятельности или увлечений, присоединяется к огромной армии потребителей информации во «Всемирной Паутине».

Web-технология полностью перевернула наши представления о работе с информацией, да и с компьютером вообще. Оказалось, что традиционные параметры развития вычислительной техники - производительность, пропускная способность, емкость запоминающих устройств - не учитывали главного «узкого места» системы - интерфейса с человеком. Устаревший механизм взаимодействия человека с информационной системой сдержи-

вал внедрение новых технологий и уменьшал выгоду от их применения. И только когда интерфейс между человеком и компьютером был упрощен до естественности восприятия обычным человеком, последовал беспрецедентный взрыв интереса к возможностям вычислительной техники.

С развитием технологий гипертекстовой разметки в Интернете стало появляться всё больше сайтов, тематика которых была совершенно различной – от сайтов крупных компаний, повествующих об успехах компании и её провалах, до сайтов маленьких фирм, предлагающих посетить их офисы в пределах одного города.

Развитие Интернет-технологий послужило толчком к появлению новой ветки в Интернете - Интернет-форумов. Стали появляться сайты, и даже целые порталы, на которых люди со всех уголков планеты могут общаться, получать ответы на любые вопросы и, даже, заключать деловые сделки.

Основные идеи современной информационной технологии базируются на концепции, согласно которой данные должны быть организованы в базы данных, с целью адекватного отображения изменяющегося реального мира и удовлетворения информационных потребностей пользователей.

Любая информационная система представляет собой программный комплекс, функции которого состоят в поддержке надежного хранения информации в памяти компьютера, выполнении специфических для данного приложения преобразований информации и/или вычислений, предоставлении пользователям удобного и легко осваиваемого интерфейса. С развитием и распространением сети Интернет информационные системы стали более интерактивными, масштабируемыми и доступными обычным пользователям.

Необходимость систем управления для владельцев сайтов начала проявляться в тот момент, когда количество материалов на веб-сайтах начало стремительно расти. Это привело к тому, что традиционные «ручные» технологии разработки и поддержки сайтов, когда сайт состоял из статических страниц и набора дополнительных специализированных скриптов, стали не успевать за быстро меняющимися условиями бизнеса. Ввод данных на сайт требовал (как минимум) знания технологий HTML/CSS верстки, изменения структуры сайтов были сопряжены с каскадным изменением большого количества взаимосвязанных страниц. Различные автоматизированные механизмы, вроде гостевых книг и новостных лент, внедренные на сайтах как отдельные скрипты и, как правило, написанные разными специалистами, перестали удовлетворять требованиям безопасности. На многих сайтах стали появляться коктейли из разных технологий и подходов к разработке, поэтому возникла потребность в стандартизации программных решений,

в разделении дизайна и содержимого на две независимые составляющие. CMS действительно разделяют сайты на две составляющие: дизайн (внешний вид сайта в целом, отдельных страниц, конкретных блоков информации) и контент. Дизайн сайта как правило «защит» в шаблоны и изменяется значительно реже, чем контент.

Система управления сайтами – это программный комплекс, позволяющий автоматизировать процесс управления как сайтом в целом, так и сущностями в рамках сайта: макетами страниц, шаблонами вывода данных, структурой, информационным наполнением, пользователями и правами доступа, а также по возможности предоставляющий дополнительные сервисы: списки рассылки, ведение статистики, поиск, средства взаимодействия с пользователями и т. д.

Обычно системы обновления делятся на две части: внешнюю – набор HTML-страниц, генерируемых при вызове страниц из браузера посетителя сайта и внутреннюю – систему администрирования. Обе части обычно используют общее хранилище данных, в роли которого, как правило, выступает реляционная база данных (иногда встречаются другие виды хранилищ, например, XML-документы или даже текстовые файлы).

В данном дипломном проекте реализуется программное средство создания веб-приложений с помощью готовых графических компонентов, которое способно интегрироваться в другие программные решения, может использоваться как самими разработчиками для создания пользовательского интерфейса веб-приложения, так и пользователями этого веб-приложения для конфигурации пользовательского интерфейса согласно собственным предпочтениям.

Темой разрабатываемого проекта является «Программное средство создания веб-приложений с помощью готовых графических компонентов».

Перечень задач, поставленных для решения в течение преддипломной практики представлен ниже:

1 АНАЛИЗ ЛИТЕРАТУРНЫХ ИСТОЧНИКОВ И СУЩЕСТВУЮЩИХ АНАЛОГОВ

1.1 Аналитический обзор литературных источников

Выбор литературных источников при подготовке к проектированию был обусловлен использованием определенных технологий, созданных для упрощения разработки веб-ориентированного программного обеспечения. В то же время, помимо печатных изданий, были использованы дополнительные ресурсы в сети Интернет.

Первым литературным источником, который использовался для получения информации по теме, стала книга автора Кайла Симпсона «You don't know JS». Эта книга подробно описывает возможности и внутреннее устройство языка программирования JavaScript, а также тонкости использования предоставляемых им инструментов и работы с ним в целом.

Вторым литературным источником, который использовался для получения информации по теме, стала книга автора Кайла Симпсона «Functional-Light JavaScript». Данный источник использовался с целью получения информации о функциональном подходе к программированию для применения их в разработке программного средства. Книга является вполне практико-ориентируемой - автор простым языком объясняет основные принципы функциональной парадигмы, не вдаваясь глубоко в теорию по этой теме, что позволяет быстро разобраться в материале и начать применять полученные знания на практике в разработке программного средства.

Третьим источником информации, который использовался для получения информации по теме, является документация технологии Webix - программного продукта, использовавшегося при разработке. Данная технология предоставляет большое количество уже спроектированных веб-компонентов со своей логикой работы и богатым инструментарием, что благоприятно сказывается на скорости разработки продуктов с использованием данной технологии.

Также стоит отметить сайт habrahabr.ru, который использовался для получения информации о REST (сокращение от англ. Representational State Transfer - «передача состояния представления») - архитектурном стиле взаимодействия компонентов распределенного приложения.

1.2 Анализ существующих аналогов

Целью данной курсовой работы является проектирование и разработка веб-сервиса для компоновки пользовательского интерфейса. Данное про-

граммное средство является интегрируемым сервисом, которое может использоваться как самими разработчиками для создания пользовательского интерфейса веб-приложения, так и пользователями этого веб-приложения для конфигурации пользовательского интерфейса согласно собственным предпочтениям. Программные средства такого рода называются конструкторами сайтов или CMS (сокращение от англ. Content Management System, - "система управления контентом сайта" или просто "система управления сайтом") и обладают следующим функционалом:

- предоставление полностью готовых шаблонов сайтов с готовым дизайном;
- предоставление готовых дизайнерских решений для пользовательского интерфейса;
- гибкая система управления сайтом.

Такие сервисы целесообразно использовать для:

- создания одностраничных (SPA – сокращение от англ. Single Page Application) под трафик с контекстной и таргетированной рекламой;
- визуализации идеи, чтобы впоследствии предоставить ее разработчикам и/или дизайнерам;
- быстрого запуска несложных проектов;
- тестирования идеи, чтобы понять, стоит ли тратить время и деньги на разработку;
- быстрого запуска несложных проектов;
- некоммерческих сайтов "для души".

Серьезные веб-проекты лучше создавать на зарекомендовавших себя CMS или "самописных" движках, заточенных под конкретные задачи. Это так, но в некоторых ситуациях такой подход слишком дорог, дорог и трудозатратен. С тем же WordPress нужно разбираться несколько недель. Если есть время и желание изучать тонкости самостоятельно или средства для оплаты услуг специалиста - отлично. В противном случае можно воспользоваться визуальными конструкторами. Это не панацея, есть проекты, которые невозможно реализовать без участия дизайнеров и программистов. Выбирать конструктор стоит исходя из конкретных задач. Некоторые отлично справляются с Landing Page, другие - подходят для создания многостраничных сайтов, третьи хорошо продвигаются в поиске. Давайте сравним популярные сервисы, чтобы понять в какой ситуации лучше использовать тот или иной продукт. При выборе инструмента для создания сайта нужно учитывать много параметров. Они зависят от типа ресурса и задач, которые он должен решать. Для удобства я составила сравнительную таблицу. В ней прописаны важные, на мой взгляд, характеристики и функциональные

особенности конструкторов, которые упомянуты в этом обзоре. Сравним сервисы по таким параметрам:

- типы сайтов - на какие ресурсы рассчитан функционал (визитка, лендинг, магазин и т.д.);
- уровень пользователей - опыт в разработке: новички, продвинутые пользователи, профессионалы;
- адаптивность шаблонов - наличие в каталоге макетов, адаптированных для мобильных устройств;
- количество готовых шаблонов - сколько в каталоге готовых макетов, за которые не придется платить отдельно;
- уровень пользователей - опыт в разработке: новички, продвинутые пользователи, профессионалы;
- адаптивность шаблонов - наличие в каталоге макетов, адаптированных для мобильных устройств;
- количество готовых шаблонов - сколько в каталоге готовых макетов, за которые не придется платить отдельно;
- уровень кастомизации шаблонов - возможность изменения элементов и дизайна в целом: высокая, средняя, низкая;
- возможность создать сайт с нуля - можно ли открыть пустой макет и собрать страницы из блоков и виджетов;
- обучающие материалы - информационная база по пользованию конструктором;
- возможность редактировать и добавлять код - добавлять свои элементы и редактировать стили через HTML и CSS;
- бесплатный тариф - наличие бесплатного тарифа. Ограничения прописаны в детальном обзоре каждого конструктора ниже;
- триал - наличие тестового периода с расширенным функционалом.
- техподдержка - язык и способы поддержки пользователей;
- минимальный тариф - стоимость минимального тарифа;
- способы оплаты - варианты оплаты тарифов и дополнительных услуг;
- интеграции - подключение к сторонним сервисам для расширения функционала сайта;
- домен - возможность подключить свой домен на бесплатном тарифе, а также условия, на которых он предоставляется бесплатно;
- SEO - возможности оптимизации сайта;
- импорт/экспорт товаров - способы загрузки большого количества товаров в каталог;
- интеграция с CRM - подключение к CRM для автоматического им-

порта заказов с сайта;

- интеграция с системами аналитики - возможность подключиться к сторонним сервисам аналитики;
- онлайн-оплата - платежные системы, которые можно подключить к сайту;
- интеграция с соцсетями - кнопки, виджеты и комментарии.

В качестве рассматриваемых аналогов будут выступать следующие решения:

- Tilda Publishing;
- LPgeneratorWIX.

Tilda Publishing

Tilda – интуитивный конструктор сайтов. Подходит для создания небольших проектов - информационных и корпоративных ресурсов, Landing Page и интернет-магазинов с десятком-другим позиций. Хотя для последних есть более удобные решения, за последний год Tilda добавила множество возможностей для этого типа сайтов. Появились полноценная корзина с вариантами доставки и оплаты, блоки карточек товаров со встроенными попапами, в которых отображаются увеличенные фотографии и расширенное описание, лейблы "хит акция и другие для визуального выделения товаров. Кроме того, сайт интегрируется с несколькими платежными системами, интернет-эквайерами и CRM. Заявки можно отслеживать во встроенном инструменте конструктора, экспортировать в Google Sheets или Telegram.

Функционал Tilda ориентирован на эффектное оформление лонгридов - стильная типографика, много блоков для комбинирования текстового, визуального и видео-контента. Возможности не ограничиваются готовыми шаблонами и блоками - и то, и другое можно разработать самостоятельно с нуля, используя конструкторы. Для тех, у кого на это нет времени и желания, в каталоге около 200 дизайнов, которые можно настроить под себя с помощью 450 блоков. Есть готовые макеты для пиццерий, антикафе, салонов красоты и других ниш. Для визуализации отдельных элементов сайта дизайнеры Тильды отрисовали целую библиотеку иконок под разные сферы бизнеса, которая регулярно пополняется.

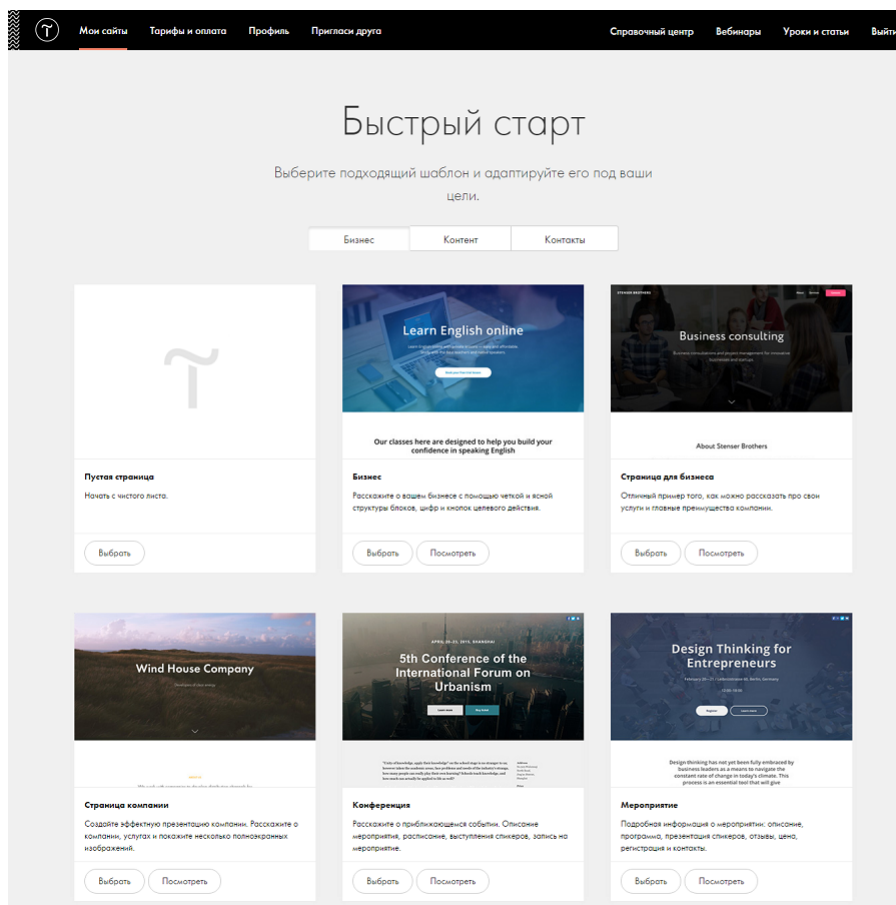


Рисунок 1.1 – Скриншот приложения "Tilda Publishing"

Достоинства:

- Большой выбор готовых шаблонов;
- Удобный интуитивно понятный интерфейс;
- Широкие возможности для кастомизации - цвета, шрифты, отступы, прозрачность, анимация;
- Стильные адаптивные шаблоны;
- Возможность отдельно настроить отступы для мобильных и десктопных версий;
- Возможность создать свой уникальный макет с нуля;
- Большой выбор модулей - текст, блоки преимуществ, формы заявок и обратной связи, опросов, онлайн-бронирования и т. д;
- Генератор UTM-меток;
- Обучающие статьи и видео-уроки по работе с сервисом;
- Хорошая типографика, можно подключать шрифты из Google Fonts, Турекит и собственной коллекции;
- Возможность добавить свои элементы с помощью HTML, CSS и JS;
- Интеграция с CRM, сервисами обратных звонков, онлайн-чатами и электронными платежными системами;

- Сайт можно перенести на другой хостинг или собственный сервер, а также интегрировать созданные на Tilda страницы с проектами на Битрикс, WordPress и другими;
- Встроенная аналитика, подключение Google Analytics и "Яндекс.Метрики";
- Отслеживание количества кликов по кнопкам, заполнения форм и других событий в "Метрике" и "Аналитике" по умолчанию, без настройки целей;
- Подключение на сайт ленты постов Instagram с автоматическим обновлением;
- Возможность настроить передачу данных в налоговую в соответствии с 54-ФЗ;
- Конструктор политики обработки персональных данных.
- Возможность сохранить и использовать для дальнейшей работы собственные шаблоны;
- Подключение HTTPS в интерфейсе конструктора.
- Функционал мультилендинга - можно создать динамический контент, чтобы внешний вид сайта и информация подстраивались под конкретного пользователя;
- Интеграция с популярными онлайн-банками - "Сбербанк "Альфа Банк" и "Тинькофф чтобы принимать платежи и получать деньги на расчетный счет;
- Функционал для верстки email-рассылок с последующей отправкой через MailChimp, UniSender или любой другой сервис с помощью экспорта кода.

Недостатки:

- Высокие тарифы;
- Для хорошего результата требуются определенные познания в веб-дизайне.

1.3 Разработка и спецификации функциональных требований

Основной целью создаваемого программного средства является предоставление возможности создать собственные веб-компоненты на основании предоставляемых платформой или разработчиком, который интегрирует данное программное средство в свой продукт; скомпоновать из готовых "односложных" компонентов более комплексные, задав им при этом логику взаимодействия. Пользователями данного программного средства могут быть как разработчики с целью более быстрой разработки интерфейса вза-

имодействия с пользователем с приложением, так и пользователи, которым разработчики, интегрирующие данное программное средство в свой продукт, пожелают предоставить возможность самим определить, как будет выглядеть для них дизайн веб-приложения. Программное средство предоставляет интерфейс для взаимодействия с ним, позволяя гибко себя сконфигурировать. Однако, существует ряд требований к разработчику, интегрирующему данное программное средство в свой продукт:

- Совместимость технологии, на которой осуществляется разработка продукта с технологией Webix (React, Angular, Ember.js);
- Разработчик должен задать место, куда будет интегрироваться программное средство;
- Разработчик должен предоставить способ получения конфигурационных объектов веб-компонентов, которые будут доступны пользователю изначально, в объект конфигурации (JSON, представляющий собой массив, содержащий объекты конфигураций компонентов или ссылку, по которой приложение сможет получить данный JSON).

Перечислим требования к программному средству:

- Стабильная работа;
- Высокая скорость вычислений;
- Простота интеграции в готовые программные решения;
- Дружелюбный интерфейс;
- Возможность кастомизации;
- Возможность протестировать макет в песочнице;
- Возможность сохранять итоговый объект конфигурации скомпонованного пользователем макета;
- Обработка исключительных ситуаций и вывод информации о любых ошибках пользователю.

В качестве системы контроля версий был выбран Git, а в качестве GUI интерфейса для работы с Git используется программное средство GitKraken, так как обладает удобным, доступным интерфейсом, позволяющим легко ориентироваться в возможностях данного программного средства.

2 МОДЕЛИРОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

3 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА

В некоторых проектах, по согласованию с руководителем, разрешается объединять разделы 3 и 4.

4 КОНСТРУИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА

Например, следующий JavaScript код

```
return <div>Hello{this.props.children}</div>;
```


5 ТЕСТИРОВАНИЕ И ПРОВЕРКА РАБОТОСПОСОБНОСТИ ПС

В данном разделе проведем динамическое ручное тестирование. В таблице 5.1 приведен список тестовых случаев, относящихся к позитивному тестированию, в таблице 5.2 – к негативному.

Таблица 5.1 – Тестовые случаи позитивного тестирования

Модуль (экран)	Описание тестового случая	Ожидаемые результаты	Тестовый случай пройден?
1	2	3	4
Аккаунт	<p>1. Регистрация. Предусловие: необходим существующий ящик электронной почты.</p> <p>1) Нажать кнопку «Регистрация» на главной странице ПС. 2) Ввести адрес электронной почты. 3) Ввести пароль "12345678". 4) Ввести пароль из предыдущего пункта в поле подтверждения пароля. 5) Нажать кнопку «Зарегистрироваться». 6) Проверить ящик электронной почты, дожидаясь получения электронного письма. 7) Перейти по ссылке из полученного письма.</p>	Отображается страница регистрации. На указанный адрес электронной почты приходит письмо со ссылкой. При переходе по ссылке появляется сообщение «Аккаунт подтвержден».	Да
Аккаунт	<p>2. Аутентификация. Предусловие: необходим зарегистрированный в системе аккаунт.</p> <p>1) Нажать кнопку «Вход» на главной странице ПС. 2) Ввести адрес электронной почты и пароль аккаунта. 3) Нажать кнопку «Войти».</p>	Отображается страница аутентификации. По нажатую кнопки «Войти» открывается главная страница зарегистрированного пользователя.	Да

Продолжение таблицы 5.1

1	2	3	4
Аккаунт	3. Редактирование профиля. Предусловие: необходимо произвести аутентификацию. 1) Нажать кнопку «Мой профиль». 2) Нажать кнопку «Редактировать». 3) Изменить значения полей имени, фамилии, отчества, описания. 4) Обновить страницу. 5) Нажать кнопку «Мой профиль».	Открывается страница профиля пользователя. Открывается страница редактирования личной информации. В профиле отображается новая информация.	Да

Таблица 5.2 – Тестовые случаи негативного тестирования

Модуль (экран)	Описание тестового случая	Ожидаемые результаты	Тестовый случай пройден?
1	2	3	4

6 РУКОВОДСТВО ПО УСТАНОВКЕ И ИСПОЛЬЗОВАНИЮ

7 ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТКИ И ВНЕДРЕНИЯ ПРОГРАММНОГО СРЕДСТВА

7.1 Характеристика программного средства

...

Разработки проектов программных средств связана со значительными затратами ресурсов. В связи с этим создание и реализация каждого проекта программного обеспечения нуждается в соответствующем технико-экономическом обосновании [2], которое и описывается в данном разделе.

7.2 Определение объема и трудоемкости ПС

Целесообразность создания ПС требует проведения предварительной экономической оценки. Экономический эффект у разработчика ПС зависит от объема инвестиций в разработку проекта, цены на готовый продукт и количества проданных копий, и проявляется в виде роста чистой прибыли.

Оценка стоимости создания ПС со стороны разработчика предполагает составление сметы затрат, вычисление цены и прибыли от реализации разрабатываемого программного средства.

Исходные данные, которые будут использоваться при расчете сметы затрат, представлены в таблице 7.1.

Перед определением сметы затрат на разработку программного средства необходимо определить его объём. Однако, на стадии ТЭО нет возможности рассчитать точные объёмы функций, вместо этого с помощью применения действующих нормативов рассчитываются прогнозные оценки.

В качестве метрики измерения объема программных средств используется строка их исходного кода (LOC – lines of code). Данная метрика широко распространена, поскольку она непосредственно связана с конечным продуктом, может применяться на всём протяжении проекта и, кроме того, может использоваться для сопоставления размеров программного обеспечения. Далее под строкой исходного кода будем понимать количество исполняемых операторов.

Расчет объема функций программного средства и общего объема приведен в таблице 7.2.

Исходя из определенной 3-ей категории сложности и общего объема ПС $V_0 = 28\,940$, нормативная трудоемкость $T_n = 520$ чел./д. [2, приложение 3]. Перед определением общей трудоемкости разработки необходимо определить несколько коэффициентов.

Таблица 7.1 – Исходные данные

Наименование показателя	Условное обозначение	Значение
Категория сложности		3
Дополнительный коэффициент сложности	$\sum_{i=1}^n K_i$	0,13
Степень охвата функций стандартными модулями	K_T	0,7
Коэффициент новизны	K_H	0,9
Количество дней в году	D_r	365
Количество праздничных дней в году	$D_{п}$	9
Количество выходных дней в году	$D_{в}$	103
Количество дней отпуска	D_o	21
Количество разработчиков	$Ч_p$	3
Тарифная ставка первого разряда, руб.	$T_{\text{ч}}^1$	265
Среднемесячная норма рабочего времени, ч.	Φ_p	168,3
Продолжительность рабочей смены, ч.	$T_{\text{ч}}$	8
Коэффициент премирования	K	1,3
Норматив дополнительной заработной платы	H_d	20%
Норматив отчислений в ФСЗН	$H_{\text{сз}}$	34%
Норматив отчислений по обязательному страхованию	$H_{\text{ос}}$	0,6%
Норматив расходов по статье «Материалы»	$H_{\text{мз}}$	5%
Норматив расходов по статье «Машинное время»	$H_{\text{мв}}$	15%
Понижающий коэффициент к статье «Машинное время»		0,5
Стоимость машино-часа, руб.	$\Pi_{\text{м}}$	0,8
Норматив расходов по статье «Научные командировки»	$H_{\text{рнк}}$	15%
Норматив расходов по статье «Прочие затраты»	$H_{\text{пз}}$	20%
Норматив расходов по статье «Накладные расходы»	$H_{\text{нр}}$	50%
Уровень рентабельности	$У_{\text{рп}}$	10%
Ставка НДС	$H_{\text{дс}}$	20%
Норматив затрат на освоение ПО	H_o	10%
Норматив затрат на сопровождение ПО	H_c	20%

Коэффициент сложности, который учитывает дополнительные затраты труда, связанные с обеспечением интерактивного доступа и хранения, и

Таблица 7.2 – Перечень и объём функций программного модуля

№ функции	Наименование (содержание)	Объём функции, LoC
101	Организация ввода информации	100
102	Контроль, предварительная обработка и ввод информации	500
109	Организация ввода/вывода информации в интерактивном режиме	190
111	Управление вводом/выводом	2600
204	Обработка наборов и записей базы данных	1900
207	Манипулирование данными	8000
208	Организация поиска и поиск в БД	7500
304	Обслуживание файлов	500
305	Обработка файлов	800
309	Формирование файла	1000
506	Обработка ошибочных и сбойных ситуаций	500
507	Обеспечение интерфейса между компонентами	750
601	Отладка прикладных программ в интерактивном режиме	4300
707	Графический вывод результатов	300
	Общий объем	28 940

поиска данных в сложных структурах [2, приложение 4, таблица П.4.2]

$$K_c = 1 + \sum_{i=1}^n K_i = 1 + 0,06 + 0,07 = 1,13, \quad (7.1)$$

где K_i — коэффициент, соответствующий степени повышения сложности за счет конкретной характеристики;

n — количество учитываемых характеристик.

Коэффициент K_t , учитывающий степень использования при разработке стандартных модулей, для разрабатываемого приложения, в котором степень охвата планируется на уровне около 50%, примем равным 0,7 [2, приложение 4, таблица П.4.5].

Коэффициент новизны разрабатываемого программного средства K_n примем равным 0,9, так как разрабатываемое программное средство принадлежит определенному параметрическому ряду существующих программных средств [2, приложение 4, таблица П.4.4].

Исходя из выбранных коэффициентов, общая трудоемкость разработ-

ки $T_o = T_n \cdot K_c \cdot K_t \cdot K_n = 520 \cdot 1,13 \cdot 0,7 \cdot 0,9 = 370$ чел./д.

Для расчета срока разработки проекта примем число разработчиков $Ч_p = 3$. Исходя из комментария к постановлению Министерства труда и социальной защиты Республики Беларусь от 05.10.16 №54 «Об установлении расчетной нормы рабочего времени на 2017 год» [3], эффективный фонд времени работы одного человека составит

$$\Phi_{эф} = D_{г} - D_{п} - D_{в} - D_{о} = 365 - 9 - 103 - 21 = 232 \text{ д.}, \quad (7.2)$$

где $D_{г}$ — количество дней в году;

$D_{п}$ — количество праздничных дней в году;

$D_{в}$ — количество выходных дней в году;

$D_{о}$ — количество дней отпуска.

Тогда трудоемкость разработки проекта

$$T_p = \frac{T_o}{Ч_p \cdot \Phi_{эф}} = \frac{370}{3 \cdot 232} = 0,53 \text{ г.} = 194 \text{ д.} \quad (7.3)$$

Исходя из того, что разработкой будет заниматься 3 человека, можно запланировать фонд рабочего времени для каждого исполнителя

$$\Phi_{пi} = \frac{T_p}{Ч_p} = \frac{194}{3} \approx 65 \text{ д.} \quad (7.4)$$

7.3 Расчет сметы затрат

Основной статьей расходов на создание ПО является заработная плата разработчиков проекта. Информация об исполнителях перечислена в таблице 7.3. Кроме того, в таблице приведены данные об их тарифных разрядах, приведены разрядные коэффициенты, а также по формулам 7.5 и 7.6 рассчитаны месячный и часовой оклады.

$$T_m = T_m^1 \cdot T_k, \quad (7.5)$$

$$T_{ч} = \frac{T_m}{\Phi_p}, \quad (7.6)$$

где T_m — месячный оклад;

T_m^1 — тарифная ставка 1-го разряда (положим ее равной 265 руб.);

T_k — тарифный коэффициент;

$T_{ч}$ — часовой оклад;

Φ_p — среднемесячная норма рабочего времени (в 2017 г. составляет 168,3 ч. [3]).

Таблица 7.3 – Работники, занятые в проекте

Исполнители	Разряд	Тарифный коэффициент	Месячный оклад, руб.	Часовой оклад, руб.
Руководитель проекта	17	3,98	1054,70	6,27
Ведущий инженер-программист	15	3,48	922,20	5,48
Инженер-программист II категории	11	2,65	702,25	4,17

Тогда основная заработная плата исполнителей составит

$$\begin{aligned}
 Z_o &= \sum_{i=1}^n T_{\text{чи}} \cdot T_{\text{ч}} \cdot \Phi_{\text{пи}} \cdot K = \\
 &= (6,27 + 5,48 + 4,17) \cdot 8 \cdot 65 \cdot 1,3 = 10\,761,92 \text{ руб.},
 \end{aligned}
 \tag{7.7}$$

где $T_{\text{чи}}$ — часовая тарифная ставка i -го исполнителя, руб.;
 $T_{\text{ч}}$ — количество часов работы в день;
 $\Phi_{\text{пи}}$ — плановый фонд рабочего времени i -го исполнителя, д.;
 K — коэффициент премирования (принятый равным 1,3).

Дополнительная заработная плата включает выплаты, предусмотренные законодательством о труде: оплата отпусков, льготных часов, времени выполнения государственных обязанностей и других выплат, не связанных с основной деятельностью исполнителей, и определяется по нормативу, установленному в организации, в процентах к основной заработной плате. Приняв данный норматив $H_d = 20\%$, рассчитаем дополнительные выплаты

$$Z_d = \frac{Z_o \cdot H_d}{100\%} = \frac{10\,761,92 \cdot 20\%}{100\%} = 2152,38 \text{ руб.}
 \tag{7.8}$$

Отчисления в фонд социальной защиты населения и в фонд обязательного страхования определяются в соответствии с действующим законодательством по нормативу в процентном отношении к фонду основной и дополнительной зарплат по следующим формулам

$$\begin{aligned}
 Z_{\text{сз}} &= \frac{(Z_o + Z_d) \cdot H_{\text{сз}}}{100\%}, \\
 Z_{\text{ос}} &= \frac{(Z_o + Z_d) \cdot H_{\text{ос}}}{100\%}.
 \end{aligned}
 \tag{7.9}$$

В настоящее время нормы отчислений в ФСЗН $H_{\text{сз}} = 34\%$ и в фонд

обязательного страхования $H_{oc} = 0,6\%$. Исходя из этого, размеры отчислений

$$Z_{cz} = \frac{(10\,761,92 + 2\,152,38) \cdot 34\%}{100\%} = 4\,390,86 \text{ руб.},$$

$$Z_{oc} = \frac{(10\,761,92 + 2\,152,38) \cdot 0,6\%}{100\%} = 77,49 \text{ руб.} \quad (7.10)$$

Расходы по статье «Материалы» отражают траты на магнитные носители, бумагу, красящие материалы, необходимые для разработки ПО определяются по нормативу к фонду основной заработной платы разработчиков. Исходя из принятого норматива $H_{мз} = 5\%$ определим величину расходов

$$M = \frac{Z_o \cdot H_{мз}}{100\%} = \frac{10\,761,92 \cdot 5\%}{100\%} = 538,10 \text{ руб.} \quad (7.11)$$

Расходы по статье «Машинное время» включают оплату машинного времени, необходимого для разработки и отладки ПО, которое определяется по нормативам на 100 строк исходного кода. Норматив зависит от характера решаемых задач и типа ПК; для текущего проекта примем $H_{мв} = 15\%$ [2, приложение 6]. Примем величину стоимости машино-часа $\Pi_m = 0,8$ руб. Тогда, применяя понижающий коэффициент 0,5, получим величину расходов

$$P_m = \Pi_m \cdot \frac{V_o}{100} \cdot H_{мв} = 0,8 \cdot \frac{28\,940}{100} \cdot 15\% \cdot 0,5 = 1\,736,40 \text{ руб.} \quad (7.12)$$

Расходы по статье «Научные командировки» определяются по нормативу в процентах к основной заработной плате. Принимая норматив равным $H_{рнк} = 15\%$ получим величину расходов

$$P_{нк} = \frac{Z_o \cdot H_{рнк}}{100\%} = \frac{10\,761,92 \cdot 15\%}{100\%} = 1\,614,29 \text{ руб.} \quad (7.13)$$

Расходы по статье «Прочие затраты» включают затраты на приобретение и подготовку специальной научно-технической информации и специальной литературы. Определяются по нормативу в процентах к основной заработной плате. Принимая норматив равным $H_{пз} = 20\%$ получим величину расходов

$$\Pi_3 = \frac{Z_o \cdot H_{пз}}{100\%} = \frac{10\,761,92 \cdot 20\%}{100\%} = 2\,152,38 \text{ руб.} \quad (7.14)$$

Затраты по статье «Накладные расходы», связанные с необходимо-

стью содержания аппарата управления, вспомогательных хозяйств и опытных (экспериментальных) производств, а также с расходами на общехозяйственные нужды, относятся к конкретному ПО по нормативу в процентном отношении к основной заработной плате исполнителей. Принимая норматив равным $H_{\text{нр}} = 50\%$ получим величину расходов

$$P_{\text{н}} = \frac{Z_o \cdot H_{\text{нр}}}{100\%} = \frac{10\,761,92 \cdot 50\%}{100\%} = 5380,96 \text{ руб.} \quad (7.15)$$

Общая сумма расходов по смете определяется как сумма вышерассчитанных показателей

$$\begin{aligned} C_{\text{п}} &= Z_o + Z_{\text{д}} + Z_{\text{сз}} + Z_{\text{ос}} + M + P_{\text{м}} + P_{\text{нк}} + P_{\text{з}} + P_{\text{н}} = \\ &= 28\,804,78 \text{ руб.} \end{aligned} \quad (7.16)$$

Рентабельность определяется из результатов анализа рыночных условий и переговоров с потребителями ПО. Исходя из принятого уровня рентабельности $Y_{\text{рп}} = 10\%$, прибыль от реализации ПО составит

$$P_o = \frac{C_{\text{п}} \cdot Y_{\text{рп}}}{100\%} = \frac{28\,804,78 \cdot 10\%}{100\%} = 2880,48 \text{ руб.} \quad (7.17)$$

На основании расчета прибыли и уровня себестоимости рассчитаем прогнозируемую цену программного средства без учета налогов

$$Ц_{\text{п}} = C_{\text{п}} + P_o = 28\,804,78 + 2880,48 = 31\,685,26 \text{ руб.} \quad (7.18)$$

Далее рассчитаем налог на добавленную стоимость

$$\text{НДС} = \frac{Ц_{\text{п}} \cdot \text{НДС}}{100\%} = \frac{31\,685,26 \cdot 20\%}{100\%} = 6337,05 \text{ руб.} \quad (7.19)$$

НДС включается в прогнозируемую отпускную цену

$$Ц_o = Ц_{\text{п}} + \text{НДС} = 31\,685,26 + 6337,05 = 38\,022,31 \text{ руб.} \quad (7.20)$$

Организация-разработчик участвует в освоении и внедрении ПО и несет соответствующие затраты, которые определяются по нормативу $H_o = 10\%$ от себестоимости ПО в расчете на три месяца

$$P_o = \frac{C_{\text{п}} \cdot H_o}{100\%} = \frac{28\,804,78 \cdot 10\%}{100\%} = 2880,48 \text{ руб.} \quad (7.21)$$

Кроме того, организация-разработчик осуществляет сопровождение ПО, которое также оплачивается заказчиком. Расчет осуществляется в соот-

ветствии с нормативом $H_c = 20\%$ от себестоимости ПО

$$P_c = \frac{C_{\pi} \cdot H_c}{100\%} = \frac{28\,804,78 \cdot 20\%}{100\%} = 5760,96 \text{ руб.} \quad (7.22)$$

Экономическим эффектом разработчика будет являться сумма прибыли с вычетом налога на прибыль

$$\Pi_{\text{ч}} = \Pi_o - \frac{\Pi_o \cdot H_{\pi}}{100\%} = 2880,48 - \frac{2880,48 \cdot 18\%}{100\%} = 2361,99 \text{ руб.} \quad (7.23)$$

7.4 Оценка экономической эффективности применения ПС у пользователя

В результате применения нового ПО пользователь может понести значительные капитальные затраты на приобретение и освоение ПО, доукомплектованием ЭВМ новыми техническими средствами и пополнение оборотных средств. Однако, если приобретенное ПО будет в достаточной степени эффективнее базового, то дополнительные капитальные затраты быстро окупятся.

Для определения экономического эффекта от использования нового ПО у потребителя необходимо сравнить расходы по всем основным статьям сметы затрат на эксплуатацию нового ПО с расходами по соответствующим статьям базового варианта. При этом за базовый вариант примем ручной вариант. Исходные данные для расчета приведены в таблице 7.4.

Общие капитальные вложения заказчика (потребителя) вычисляются следующим образом

$$\begin{aligned} K_o &= K_{\text{пр}} + K_{\text{ос}} + K_c = 38\,022,31 + 2880,48 + 5760,96 = \\ &= 46\,663,75 \text{ руб.} \end{aligned} \quad (7.24)$$

где $K_{\text{пр}}$ — затраты пользователя на приобретение ПО по отпускной цене;
 $K_{\text{ос}}$ — затраты пользователя на освоение;
 K_c — затраты пользователя на оплату услуг по сопровождению.

В качестве типичного примера использования разрабатываемого ПС предполагается сценарий ее использования ежедневно 4 раза в день, при этом предполагается снижение трудоемкости с $T_{c1} = 2$ чел./ч. до $T_{c2} = 0,1$ чел./ч.. Приняв среднемесячную заработную плату работника $Z_{\text{см}} = 716,5$ руб., рассчитаем экономию затрат на заработную плату в

¹⁾На февраль 2017 г. [4]

Таблица 7.4 – Исходные данные

Наименование показателя	Условное обозначение	Значение в базовом варианте	Значение в новом варианте
Затраты пользователя на приобретение ПО	$K_{пр}$	—	38 022,31 руб.
Затраты пользователя на освоение	$K_{ос}$	—	2880,48 руб.
Затраты пользователя на сопровождение	K_c	—	5760,96 руб.
Трудоемкость на задачу, чел./ч.	T_{c1}, T_{c2}	2	0,1
Средняя зарплата ¹⁾ , руб.	$З_{см}$	716,5	716,5
Количество выполняемых задач	A_1, A_2	1460	1460
Время простоя сервиса, мин. в день	Π_1, Π_2	50	10
Стоимость одного часа простоя, руб.	$C_{п}$	79,8	79,8

расчете на одну задачу $C_{зе}$ и за год C_3

$$C_{зе} = \frac{З_{см} \cdot (T_{c1} - T_{c2})}{\Phi_p} = \frac{716,5 \cdot (2 - 0,1)}{168,3} = 8,09 \text{ руб.}, \quad (7.25)$$

$$C_3 = C_{зе} \cdot A_2 = 8,09 \cdot 1460 = 11\,811,4 \text{ руб.}, \quad (7.26)$$

где Φ_p — среднемесячная норма рабочего времени, ч.

Экономия с учетом начислений на зарплату

$$C_n = C_3 \cdot \frac{100\% + K}{100\%} = 11\,811,4 \cdot \frac{100\% + 1,3}{100\%} = 11\,964,95 \text{ руб.}, \quad (7.27)$$

где K — норматив начислений на зарплату, руб.

Экономия за счет сокращения простоев сервиса

$$C_c = \frac{(\Pi_1 - \Pi_2) \cdot D_{рг} \cdot C_{п}}{60} = \frac{(50 - 10) \cdot 300 \cdot 79,8}{60} = 15\,960,00 \text{ руб.}, \quad (7.28)$$

где $D_{рг}$ — плановый фонд работы сервиса, д.

Тогда общая годовая экономия текущих затрат, связанных с использо-

ванием нового ПО

$$C_o = C_n + C_c = 11\,964,95 + 15\,960,00 = 27\,924,95 \text{ руб.} \quad (7.29)$$

Внедрение нового ПО позволит пользователю сэкономить на текущих затратах, то есть получить на эту сумму дополнительную прибыль. Для пользователя в качестве экономического эффекта выступает лишь чистая прибыль. Принимая размер ставки налога на прибыль $H_n = 18\%$ получим

$$\Delta\P_{\text{ч}} = C_o - \frac{C_o \cdot H_n}{100\%} = 27\,924,95 - \frac{27\,924,95 \cdot 18\%}{100\%} = 22\,898,46 \text{ руб.} \quad (7.30)$$

В процессе использования нового ПО чистая прибыль в конечном итоге возмещает капитальные затраты. Однако полученные при этом суммы прибыли и затрат по годам приводят к единому времени – расчетному году (за расчетный год принят 2017-й год) путем умножения результатов и затрат за каждый год на коэффициент дисконтирования α . При расчете используются следующие коэффициенты: 2017 г. – 1, 2018 г. – 0,8696, 2019 г. – 0,7561, 2020 г. – 0,6575. Все рассчитанные данные экономического эффекта сводятся в таблицу 7.5.

В результате технико-экономического обоснования применения программного средства были получены следующие значения показателей эффективности:

- чистая прибыль разработчика составит $\Pi_{\text{ч}} = 2361,99$ руб.;
- затраты заказчика окупятся уже на четвертом году использования;
- экономическая эффективность для заказчика, выраженная в виде чистого дисконтированного дохода, составит 5618,02 руб. за четыре года использования данного ПС; более высокий прирост прибыли заказчик получит по истечению данного срока.

Полученные результаты свидетельствуют об эффективности разработки и внедрения проектируемого программного средства.

Таблица 7.5 – Расчет экономического эффекта от использования нового ПО

Наименование показателя	2017 г.	2018 г.	2019 г.	2020 г.
<i>Результаты</i>				
Коэффициент приведения	1	0,8696	0,7561	0,6575
Прирост прибыли за счет экономии затрат (П _ч), руб.	—	22 898,46	22 898,46	22 898,46
То же с учетом фактора времени, руб.	—	19 912,50	17 313,53	15 055,74
<i>Затраты</i>				
Приобретение ПО (К _{пр}), руб.	38 022,31	—	—	—
Освоение ПО (К _{ос}), руб.	2880,48	—	—	—
Сопровождение ПО (К _с), руб.	5760,96	—	—	—
Всего затрат (К _о), руб.	46 663,75	—	—	—
То же с учетом фактора времени, руб.	46 663,75	—	—	—
<i>Экономический эффект</i>				
Превышение результата над затратами, руб.	—46 663,75	19 912,5	17 313,53	15 055,74
То же с нарастающим итогом, руб.	—46 663,75	—26 751,25	—9437,72	5618,02

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] ISTQB glossary – Specification [Электронный ресурс]. — Электронные данные. — Режим доступа: <http://glossary.astqb.org/search/specification>. — Дата доступа: 25.03.17.

[2] Палицын, В.А. Техничко-экономическое обоснование дипломных проектов: Метод. пособие для студ. всех спец. БГУИР. В 4-х ч. Ч. 4: Проекты программного обеспечения / В.А. Палицын. — Мн : БГУИР, 2006. — 76 с.

[3] Пещенко, Е.А. Производственный календарь на 2017 год [Электронный ресурс]. — Электронные данные. — Режим доступа: <http://www.mintrud.gov.by/system/extensions/spaw/uploads/files/Kommetarij-2017-RV.pdf>. — Дата доступа: 06.04.17.

[4] Белстат. О начисленной средней заработной плате работников в феврале 2017 г. [Электронный ресурс]. — Электронные данные. — Режим доступа: http://www.belstat.gov.by/ofitsialnaya-statistika/solialnaya-sfera/trud/operativnaya-informatsiya_8/o-nachislennoi-srednei-zarabotnoi-plate-rabotnikov/o-nachislennoy-sredney-zarabotnoy-plate-rabotnikov-v-fevrale-2017-nbsp-g/. — Дата доступа: 07.04.17.

ПРИЛОЖЕНИЕ А

(обязательное)

Фрагменты исходного кода

```
<!-- index.html -->
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8" />
  <link rel="stylesheet" href="./node_modules/semantic-ui-css/semantic.min.
    css"/>
</head>
<body>
  <div id="root"></div>

  <!-- Dependencies -->
  <script src="./node_modules/react/dist/react.js"></script>
  <script src="./node_modules/react-dom/dist/react-dom.js"></script>

  <!-- Main -->
  <script src="./dist/bundle.js"></script>
</body>
</html>
```