

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
Кафедра «Интеллектуальные информационные технологии»

«К защите допускаю»

Заведующий кафедрой

_____ В. А. Головки

“ ____ ” _____ 2017 г.

**Разработка программно-информационного обеспечения для под-
держки процессов управления проектами**

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К ДИПЛОМНОМУ ПРОЕКТУ

ДП.АС36.120006-07 81 00

Листов 69

Руководитель

Г. Л. Муравьев

Выполнил

А. С. Бурштын

Консультанты:

по основным разделам

Г. Л. Муравьев

по экономическому

А. К. Крамаренко

по ЕСПД

А.Л. Михняев

Рецензент

А.А. Козинский

2017

АННОТАЦИЯ

69 с./ 16с., 54 рис., 22 табл., 10 ист. лит., 3 прил., 6 л. граф. матер.

В дипломном проекте разработано программно-информационное обеспечение для поддержки процессов управления проектами. Система представляет собой web-приложение, которое работает на устройствах с наличием браузера и доступа в интернет. Приложение предназначено для управления проектами в небольших It-компаниях.

Система обеспечивает ведение базы данных, учет и анализ проектов, организацию и управление проектами, генерирует отчеты, управляет доступом и авторизацией.

Приведены результаты проектирования программной части системы и информационной части, а также пользовательского интерфейса.

Составлен акт о внедрении системы в процесс функционирования, система находится на стадии тестирования

Представлены результаты тестирования системы, а также результаты оценки экономической эффективности системы.

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”

Факультет _____ ЭИС _____ Кафедра _____ ИИТ _____

УТВЕРЖДАЮ

Зав. кафедрой _____

(подпись)

« » _____ 2017 г.

ЗАДАНИЕ

по дипломному проектированию

Студенту БУРШТЫНУ Александру Сергеевичу

1. Тема проекта Разработка программно-информационного обеспечения для поддержки процессов управления проектами

(Утверждена приказом по вузу от 21.04.2017 № 485-с)

2. Сроки сдачи студентом законченного проекта 10.06.2017 г.

3. Исходные данные к проекту _____

Объект – проекты в части автоматизации процессов их регистрации, планирования, управления ресурсами, контроля исполнения и т.д.

Цель автоматизации – сокращение затрат на учет и обработку данных по управлению проектами; повышение оперативности доступа, достоверности и полноты данных; повышение эффективности принимаемых решений

Особенности разработки: - построение системы на базе веб-технологий; - максимальное использование типовых подходов, компонентов; - взаимодействие с сервером APACHE2, средств PHP 5.6, MYSQL 5.5; - обеспечение единообразной схемы учета данных; - обеспечение приемлемого быстродействия

Система должна обеспечивать:

- ведение базы данных (данные о сотрудниках, проектах, работах, ресурсах и др.);
- комплекс задач по учету и анализу данных о проектах и ресурсах (фиксация, анализ и т.д.);
- комплекс задач по организации и планированию проектов;
- комплекс задач по управлению проектами (ведение, контроль и т.д.);
- генерацию отчетов, формирование выходных документов, оперативной отчетности (включая отчеты за заданный период времени, в разрезе отдельных сотрудников, проектов, подразделений и предприятия в целом)
- комплекс задач, обеспечивающих управление доступом, сохранность, восстанавливаемость информации, авторизацию пользователей, управление профилями пользователей

4. Содержание расчетно-пояснительной записки (перечень подлежащих разработке вопросов)

Введение

1. Системный анализ и постановка задачи

2. Проектирование системы

3. Реализация и испытание системы

4. Технико-экономическое обоснование

5. Энерго- и ресурсосбережение

Заключение

Список сокращений

Список использованных источников

Приложение А Текст программы

Приложение Б Инструкция по установке и проверке системы

Приложение В Описание применения системы

5. Перечень графического материала (с точным указанием обязательных чертежей и графиков)

1. Постановка задачи (плакат – формат А1)
2. Структура системы (плакат – формат А1)
3. Схема работы системы (чертеж “схема работы системы” – формат А1)
4. Схема программы (чертеж “схема программы” – формат А1)
5. Структура пользовательского интерфейса (плакат – формат А1)
6. Результаты испытаний (плакат – формат А1)

6. Консультанты по проекту (с указанием относящихся к ним разделов проекта)

доцент Муравьев Г. Л.	– по основным разделам
ст. преподаватель Михняев А. Л.	– нормоконтроль
ст. преподаватель Крамаренко А. К.	– экономический раздел

7. Дата выдачи задания 30.03.2017 г.

8. Календарный график работы над проектом на весь период проектирования (с указанием сроков выполнения и трудоемкость отдельных этапов)

Раздел 1:	30.03 – 13.04:	20%
Раздел 2:	14.04 – 04.05:	30%
Раздел 3:	05.05 – 25.05:	30%
Раздел 4, 5:	26.05 – 01.06:	10%
Оформление проекта:	02.06 – 10.06:	10%

Руководитель

(подпись)

Задание принял к исполнению (дата)

(подпись студента)

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1 СИСТЕМНЫЙ АНАЛИЗ И ПОСТАНОВКА ЗАДАЧИ	7
1.1 Описание предметной области.....	7
1.2 Обзор аналогичных приложений	10
1.3 Обоснование необходимости разработки системы.....	12
1.4 Постановка задачи на разработку системы.....	13
2 ПРОЕКТИРОВАНИЕ СИСТЕМЫ.....	14
2.1 Проектирование структуры системы.....	14
2.2 Проектирование программного обеспечения системы.....	16
2.3 Проектирование информационного обеспечения системы.....	19
2.4 Проектирование пользовательского интерфейса системы.....	26
3 РЕАЛИЗАЦИЯ И ИСПЫТАНИЕ СИСТЕМЫ	36
3.1 Выбор средств реализации системы	36
3.2 Программная реализация	38
3.3 Испытание системы.....	44
4 ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ	55
4.1 Исходные данные для осуществления расчета.....	55
4.2 Расчет объема функций программного обеспечения.....	55
4.3 Расчет полной себестоимости программного обеспечения	57
4.4 Расчет отпускной цены и прибыли программного обеспечения	62
5 ЭНЕРГО– И РЕСУРСОСБЕРЕЖЕНИЕ.....	64
5.1 Ресурсосбережение.....	64
5.2 Энергосбережение	65
ЗАКЛЮЧЕНИЕ.....	67
СПИСОК СОКРАЩЕНИЙ	68
СПИСОК ЛИТЕРАТУРЫ	69
ПРИЛОЖЕНИЕ А ТЕКСТ ПРОГРАММЫ	
ПРИЛОЖЕНИЕ Б ИНСТРУКЦИЯ ПО УСТАНОВКЕ И ПРОВЕРКЕ СИСТЕМЫ	
ПРИЛОЖЕНИЕ В ОПИСАНИЕ ПРИМЕНЕНИЯ СИСТЕМЫ	

					ДП.АС36.120006 – 07 81 00		
Изм.	Лист	№ докум.	Подпись	Дата			
Разработ.	Бурицын А.С.				Разработка программно-информационного обеспечения для поддержки процессов управления проектами. Пояснительная записка	Лит.	Лист
Проверил	Муравьев Г.Л.						4
							69
Н. контр.	Михняев А.Л.					УО «БрГТУ»	
Утв.	Головки В. А.						

ВВЕДЕНИЕ

Сегодня, технологии веб разработки позволяют создавать проекты различного спектра, сложности и направления. Они постоянно совершенствуются и развиваются, новые улучшаются, старые забываются. Происходит так называемый процесс эволюции технологий. На сегодняшний день управление проектами неотъемлемая часть работы веб студий различного масштаба. Многие считают такие системы удобными, потому, что они позволяют экономить время на распределение ресурсов, контролю финансов и отслеживанием активности. Существует уже готовые решения, но в некоторых специфичных ситуациях требуются оригинальные решения для таких задач. Веб-инструменты в наше время позволяют реализовывать проектных разных сложностей.

Объектом разрабатываемое системы являются проекты, в части автоматизации процессов, их регистрации, планирования, управления ресурсами.

Целью автоматизации является сокращение затрат на учет и обработку данных по управлению проектами, повышение оперативности доступа, достоверности и полноты данных, повышение эффективности принимаемых решений.

Система должна обеспечивать:

- ведение базы данных (данные о сотрудниках, проектах, работах, ресурсах и др.);
- комплекс задач по учету и анализу данных о проектах и ресурсах (фиксация, анализ и т.д.);
- комплекс задач по организации и планированию проектов;
- комплекс задач по управлению проектами (ведение, контроль и т.д.);
- генерацию отчетов, формирование выходных документов, оперативной отчетности (включая отчеты за заданный период времени, в разрезе отдельных сотрудников, проектов, подразделений и предприятия в целом)
- комплекс задач, обеспечивающих управление доступом, сохранность, восстанавливаемость информации, авторизацию пользователей, управление профилями пользователей.

Дополнительные требования к системе:

- санкционированный доступ к базе данных
- интеграция в сервер на базе Apache
- функционирование в составе сервера;
- тип базы данных - реляционная, доступ с использованием phpmyadmin или любой другой системы управления базами данных mysql;

Для разработки системы необходимо:

- обследовать предметную область;

- выбрать концепцию построения системы и средств разработки;
- разработать структуру системы, интерфейс, отчетные формы, модули ПО, спроектировать БД;
- провести тестирование, испытание системы, документирование результатов разработки.

1 СИСТЕМНЫЙ АНАЛИЗ И ПОСТАНОВКА ЗАДАЧИ

1.1 Описание предметной области

Предметной областью данной работы является поддержка процессов управления проектами. Поддержка представляется собой, обеспечение системой необходимых инструментов для работы над проектами в сфере ИТ, в, относительно, небольшой корпоративной структуре. Система должна обеспечить функционалом и поддерживать процессы для нормальной работы над проектами. Таким образом, система имеет необходимые возможности для обеспечения работы всех процессов управления над проектами.

Основные виды ИТ-проектов:

- проекты разработки и развития программного обеспечения;
- проекты внедрения информационных систем;
- инфраструктурные и организационные проекты.

Особенности проектов разработки и развития программного обеспечения:

- Разработка программного обеспечения осуществляется в рамках методологий, методов и подходов программной инженерии.

- Программная инженерия (Software Engineering)— это инженерная дисциплина, которая связана со всеми аспектами производства ПО от начальных стадий создания спецификации до поддержки системы после сдачи в эксплуатацию.

- Модель программного процесса — это упрощенное описание программного процесса, представленное с некоторой точки зрения. Модели всегда являются упрощениями.

- Метод программной инженерии — это структурный подход к созданию ПО, нацеленный на создание эффективного продукта наиболее прибыльным (рентабельным, cost-effective) путем. Практически все методы построены на идее создания графических моделей системы с последующим использованием этих моделей в качестве спецификации или архитектуры системы.

Основные фазы программного процесса:

					ДП.АС36.120006 – 07 81 00	Лист
						7
Изм	Лист	№ докум	Подп.	Дата		

- Создание спецификации ПО – что система должна делать и ограничения на разработку.
- Разработка ПО – производство программной системы.
- Тестирование ПО (включает в себя validation и verification) – проверка того, что клиент хочет именно того, что прописано в спецификации, и что система соответствует спецификации.
- Развитие или эволюция ПО (software evolution) – изменение ПО в ответ на изменение внешних требований.

Типы моделей программного процесса:

- Модель технологического процесса (workflow model) — показывает последовательность действий, наряду со входами, выходами и зависимостями.
- Модель потоков данных (data flow or activity model) — представляет процесс в виде набора действий, каждый из которых выполняет некоторое преобразование данных. В этой модели действия могут быть более низкого уровня, чем в предыдущей модели.
- Модель роль/действие (role/action model) — показывает роли людей, участвующих в программном процессе, а также действия, за которые они отвечают.

10 основных областей знаний программной инженерии:

- Software requirements – программные требования;
- Software design – дизайн (архитектура);
- Software construction – конструирование ПО;
- Software testing – тестирование;
- Software maintenance – эксплуатация (поддержка) ПО;
- Software configuration management – конфигурационное управление;
- Software engineering management – управление проектами ПИ;
- Software engineering process – процессы ПИ;
- Software engineering tools and methods – инструменты и методы ПИ;
- Software quality – качество ПО.

Особенности проектов внедрения информационных систем:

- Корпоративные информационные системы управления (интегрированные системы управления предприятием, ИСУП на основе ERP) – мощнейший инструмент и жизненная необходимость для большинства организаций.

- На практике применяются: стратегия «большого взрыва», «шаг за шагом» или пилотное внедрение.

- Программно-зависимые поэтапные модели (например, ValueSAP — целостный подход, объединяющий в комплексной инфраструктуре методы, инструменты и опыт компании SAP).

В данной системе проект, набор задач, над которыми нужно выполнить действия, связанные одной предметной областью.

Под процессами понимаются любые действия происходящие в системе управления, такие как регистрация, удаление и ведение проекта, контроль времени, управление коэффициентами, ставками и прочими внутри проектными транзакциями.

Под управлением имеется в виду обеспечение создания проекта, подключение исполнителя и менеджера к нему, выставление суммы и времени на проект, отслеживание выполненных задач, создание отчета по задачам, привязка клиента к проекту. Управление должно включать распределение доступа к разделам системы, т.е. отношение администратор-пользователь. Также, контроль над всеми пользователями системы, закрепленными за ними проектами и всеми клиентами компании.

В данный период времени, системы поддержки процессов управления проектами имеют широкий профиль и набор функций, которые зачастую не нужны для конкретных компаний, здесь же мы разрабатываем систему, которая выполняет основные задачи для поддержки этих процессов.

Каждая такая система имеет отношение администратор-пользователь, т.к. система устанавливается на сервер(желательно локальный сервер компании), регистрация проходит в обычном режиме, далее администратор подтверждает регистрацию через заполнения соответствующего поля в СУБД.

Система обеспечивает, как говорилось ранее, создание редактирование и удаление проекта, пользователя и клиента. Администратору доступны разделы с отчетом по рабочему времени, установлению денежных коэффициентов и балансу предприятия. Пользователь, в свою очередь, по выполнению задачи заполняет форму, которую отслеживает менеджер или администратор проекта. Привилегии доступны и в случае с просмотром информации по конкретным клиентам, где есть все, что нужно знать об исполнителе.

Все данные отображаются в, удобного вида, окнах, которые имеют конкретную цель и, каждому разделу и функции, ставится в соответствие сущность.

Заполнение информации осуществляется в обычных веб-формах с предельно понятным пользовательским интерфейсом. Что должно избавить от трудностей в освоении системы.

Для решения такой задачи, лучше всего, подходит веб-средства, что избавляет нас от привязки к локальным системам, и обеспечению доступа к системе из любой точки. Также, выбранные технологии построения приложений, должны быть наиболее популярными, для расположения на практически любом сервере.

1.2 Обзор аналогичных приложений

На рынке не могло не существовать подобных решений, из-за их популярности и необходимости в работе над проектами, приведем основные и самые популярные на рынке, на данный момент. Но все они в основном имеют большой набор избыточных инструментов для анализа и аналитики, что в работе небольших студий приводит к избыточности функционала и замедляет время работы над проектами, такие инструменты имеют свои плюсы и свои минусы, а у нас стоит цель обеспечить самый необходимый функционал в легкодоступном виде.

-Redmine [Рэдмайн] — открытое серверное веб-приложение для управления проектами и задачами (в том числе для отслеживания ошибок). Redmine написан на Ruby и представляет собой приложение на основе широко известного веб-фреймворка Ruby on Rails. Распространяется согласно GNU General Public License.

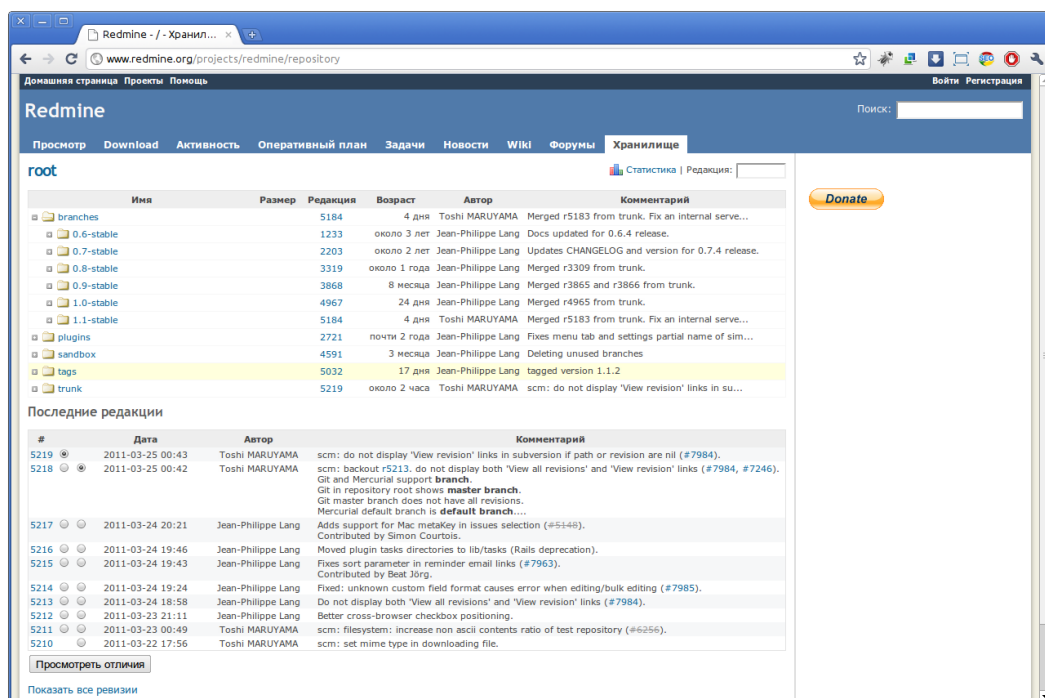


Рисунок 1.1 – Система Redmine

-gemini

Требуют кредитную карточку для.

Управление багами, фичерами, user stories, много полей, можно добавлять свои, можно свой порядок состояний бага. Баг может зависеть от других багов. Есть интеграция с github.

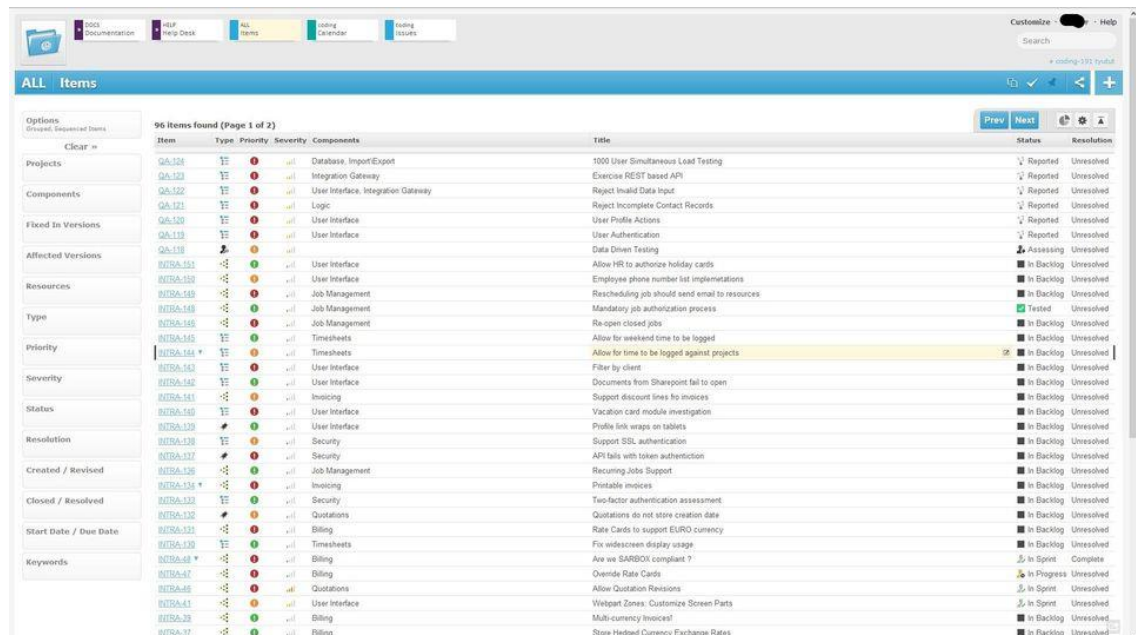


Рисунок 1.2 – Система gemini

-YouTrack

коммерческая система отслеживания ошибок, программное обеспечение для управления проектами, разработанное компанией JetBrains. YouTrack поддерживает поисковые запросы, автодополнение, манипуляцию с наборами задач, настройку набора атрибутов задачи, создание пользовательских рабочих процессов и реализует подход, основанный на преимущественном использовании клавиатуры. YouTrack интегрируется с системами управления версиями, такими как Subversion, CVS, Rational ClearCase, Visual SourceSafe, Mercurial, Git и Perforce с помощью TeamCity. Также имеется возможность интеграции с сервисом GitHub. Есть возможность интеграции с TeamCity, IntelliJ IDEA, TestLink, TestRail, поддержка аутентификации пользователей с помощью Google Account, Yahoo!, OpenID и LDAP. YouTrack может импортировать задачи, созданные в других системах отслеживания ошибок: JIRA, FogBugz, Bugzilla, MantisBT, Trac, CSV-файлов и багтрекера Google Code. Пользовательский интерфейс YouTrack основан на технологии AJAX и позволяет производить управление как с помощью клавиатуры, так и с помощью мыши. Поиск задач осуществляется с помощью поисковых запросов, выполняемых в единой строке поиска.

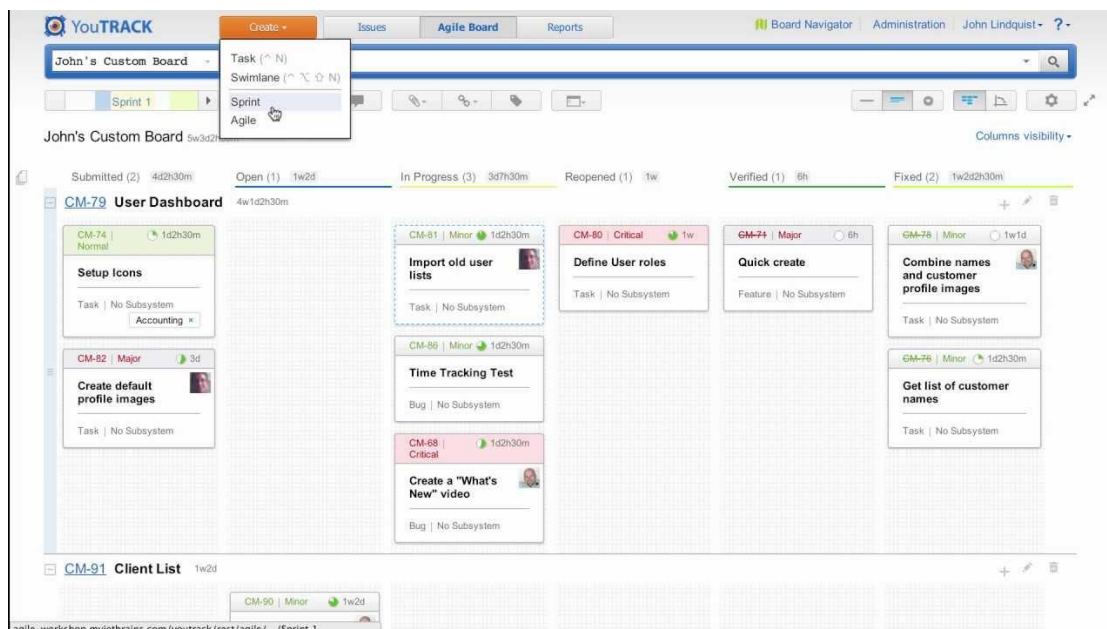


Рисунок 1.3. – Система YouTrack

1.3 Обоснование необходимости разработки системы

Создание подобной системы обусловлено нехваткой именно аналогов такой системы. Есть широкий спектр приложений, которые обеспечивают подобным функционалом, но зачастую они имеют более широкий его набор, что является подводным камнем таких систем. В нашем случае, хотим добиться простоты интерфейса и набор минимального и самого необходимого функционала, для работы над проектами. В аналогичных системах, рассмотренных в предыдущем пункте работы, огромный набор дополнительных функций, что, в случае небольших подразделений, который работают над проектами в сфере IT, лишь затрудняет работу, увеличивает время на адаптацию пользователей системы, изучение функционала и внедрению в работу. На подключение и отключение модулей может уйти огромное количество времени. Еще одним минусом является высокая нагрузка на сервер, при включении всего функционала таких систем, т.к. обычно подобные корпоративные решения, для управления над внутренними проектами, устанавливаются на сервер компании, а он чаще не подразумевает высоких нагрузок со стороны систем, расположенных на нем. Другие системы, так называемые, облачные, имеют свой огромный минус, это стоимость, чаще они стоят денег, что для начинающих компаний не является лучшим решением, а также, требуется значительное количество времени для настройки и поддержки таких систем. Зачастую они тоже являются не лучшими в плане юзабилити интерфейсов.

Таким образом, считаю целесообразным разработку такой системы, что удовлетворяла бы основным требованиям по управлению проектами.

1.4 Постановка задачи на создание системы

Объект – проекты в части автоматизации процессов их регистрации, планирования, управления ресурсами, контроля исполнения и т.д.

Цель автоматизации – сокращение затрат на учет и обработку данных по управлению проектами; повышение оперативности доступа, достоверности и полноты данных; повышение эффективности принимаемых решений.

Особенности разработки: - построение системы на базе веб-технологий;

- максимальное использование типовых подходов, компонентов;

- взаимодействие с сервером APACHE2, средств PHP 5.6, MYSQL 5.5;

- обеспечение единообразной схемы учета данных;

- обеспечение приемлемого быстродействия

Система должна обеспечивать:

- ведение базы данных (данные о сотрудниках, проектах, работах, ресурсах и др.);

- комплекс задач по учету и анализу данных о проектах и ресурсах (фиксация, анализ и т.д.);

- комплекс задач по организации и планированию проектов;

- комплекс задач по управлению проектами (ведение, контроль и т.д.);

- генерацию отчетов, формирование выходных документов, оперативной отчетности (включая отчеты за заданный период времени, в разрезе отдельных сотрудников, проектов, подразделений и предприятия в целом)

- комплекс задач, обеспечивающих управление доступом, сохранность, восстанавливаемость информации, авторизацию пользователей, управление профилями пользователей.

2 ПРОЕКТИРОВАНИЕ СИСТЕМЫ

2.1 Проектирование структуры системы

Общая структура системы изображена на рисунке 2.1



Рисунок 2.1 – Схема системы

Шаблон MVC описывает простой способ построения структуры приложения, целью которого является отделение логики от пользовательского интерфейса. В результате, приложение легче масштабируется, тестируется, сопровождается и конечно же реализуется.

В архитектуре MVC модель предоставляет данные и правила бизнес-логики, представление отвечает за пользовательский интерфейс, а контроллер обеспечивает взаимодействие между моделью и представлением.

Модель — содержит бизнес-логику приложения и включает методы выборки (это могут быть методы ORM), обработки (например, правила валидации) и предоставления конкретных данных, что зачастую делает ее очень толстой, что вполне нормально.

Модель не должна напрямую взаимодействовать с пользователем. Все переменные, относящиеся к запросу пользователя должны обрабатываться в контроллере.

Модель не должна генерировать HTML или другой код отображения, который может изменяться в зависимости от нужд пользователя. Такой код должен обрабатываться в видах.

Одна и та же модель, например: модель аутентификации пользователей может использоваться как в пользовательской, так и в административной части приложения. В таком случае можно вынести общий код в отдельный класс и наследоваться от него, определяя в наследниках специфичные для подприложений методы.

Вид — используется для задания внешнего отображения данных, полученных из контроллера и модели.

Виды содержат HTML-разметку и небольшие вставки PHP-кода для обхода, форматирования и отображения данных.

Не должны напрямую обращаться к базе данных. Этим должны заниматься модели.

Не должны работать с данными, полученными из запроса пользователя. Эту задачу должен выполнять контроллер.

Может напрямую обращаться к свойствам и методам контроллера или моделей, для получения готовых к выводу данных.

Виды обычно разделяют на общий шаблон, содержащий разметку, общую для всех страниц (например, шапку и подвал) и части шаблона, которые используют для отображения данных выводимых из модели или отображения форм ввода данных.

Контроллер — связующее звено, соединяющее модели, виды и другие компоненты в рабочее приложение. Контроллер отвечает за обработку запросов пользователя. Контроллер не должен содержать SQL-запросов. Их лучше держать в моделях. Контроллер не должен содержать HTML и другой разметки. Её стоит выносить в виды.

В хорошо спроектированном MVC-приложении контроллеры обычно очень тонкие и содержат только несколько десятков строк кода. Чего, не скажешь о Stupid Fat Controllers (SFC) в CMS Joomla. Логика контроллера довольно типична и большая ее часть выносится в базовые классы.

Модели, наоборот, очень толстые и содержат большую часть кода, связанную с обработкой данных, т.к. структура данных и бизнес-логика, содержащаяся в них, обычно довольно специфична для конкретного приложения.

Типичную последовательность работы MVC-приложения можно описать следующим образом:

- При заходе пользователя на веб-ресурс, скрипт инициализации создает экземпляр приложения и запускает его на выполнение.
- При этом отображается вид, скажем главной страницы сайта.
- Приложение получает запрос от пользователя и определяет запрошенные контроллер и действие. В случае главной страницы, выполняется действие по умолчанию (index).
- Приложение создает экземпляр контроллера и запускает метод действия,
- в котором, к примеру, содержатся вызовы модели, считывающие информацию из базы данных.
- После этого, действие формирует представление с данными, полученными из модели и выводит результат пользователю.

2.2 Проектирование программного обеспечения системы

Организация корневого каталога проекта.

Корневой каталог исходного кода системы представлен на рисунке 2.2.

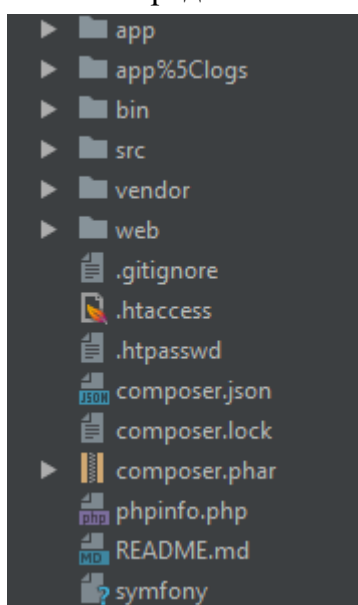


Рисунок 2.2 - Корневой каталог проекта

App- содержит файлы фреймворка, а также параметры конфигурации сервера

App/cache – хранит кэш приложения

App/config – конфигурация приложения, доступ к БД и прочие настройки фреймворка

App/logs – файлы-логи, логи работы приложения

App/resources – файлы где находятся сторонние модули приложения, у нас один такой модуль FOSUSERBUNDLE – для организации работы с пользователями

App/resources/view – базовые файлы отображения информации, которые организуют работу головы, подвала, меню приложения

Bin – внутренний каталог фреймворка

Src – основной каталог функционала приложения

Command – каталог содержащий функции, которые выполняются в автоматическом режиме

SetRateCommand.php – Автоматическое обновление ставки

Controller- контроллеры приложения

DefaultController.php – Основной контроллер приложения, где находятся обработчики всех моделей, используемых в приложении

Entity – каталог, содержащий все пользовательские классы, используемые в приложении, все они описывают сущности, представленные в следующем разделе работы

Balanse.php – класс баланса

Clients.php – класс клиентов

Currency.php – класс валют

Finance.php – класс финансов

Manager.php – класс менеджеров

Projects.php – класс проектов

StatusTask.php – класс статусов задач

Task.php – класс задач

User.php – класс пользователей

Repository – классы, перенаправляющие сущности, имеют названия, как в предыдущем разделе, имеют пустую структуру.

Resurses – здесь находится функционал для отображения и стили приложения. Название файла, описывает какой объект он показывает. Представлено на рисунке 2.3

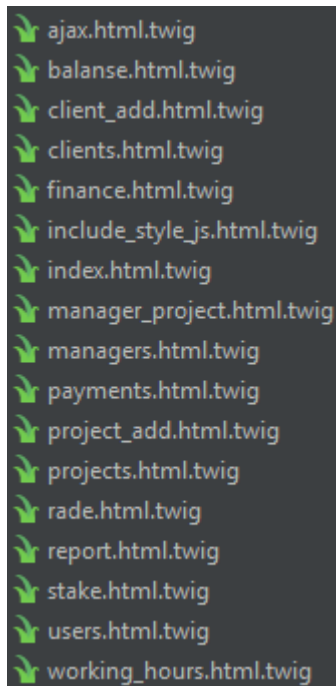


Рисунок 2.3 – Содержание каталога Resurses

Vendor – каталог фреймворка.

Web - каталог фреймворка.

UML — язык графического описания для объектного моделирования в области разработки программного обеспечения, моделирования бизнес-процессов, системного проектирования и отображения организационных структур.

UML является языком широкого профиля, это — открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML-моделью. UML был создан для определения, визуализации, проектирования и документирования, в основном, программных систем. UML не является языком программирования, но на основании UML-моделей возможна генерация кода.

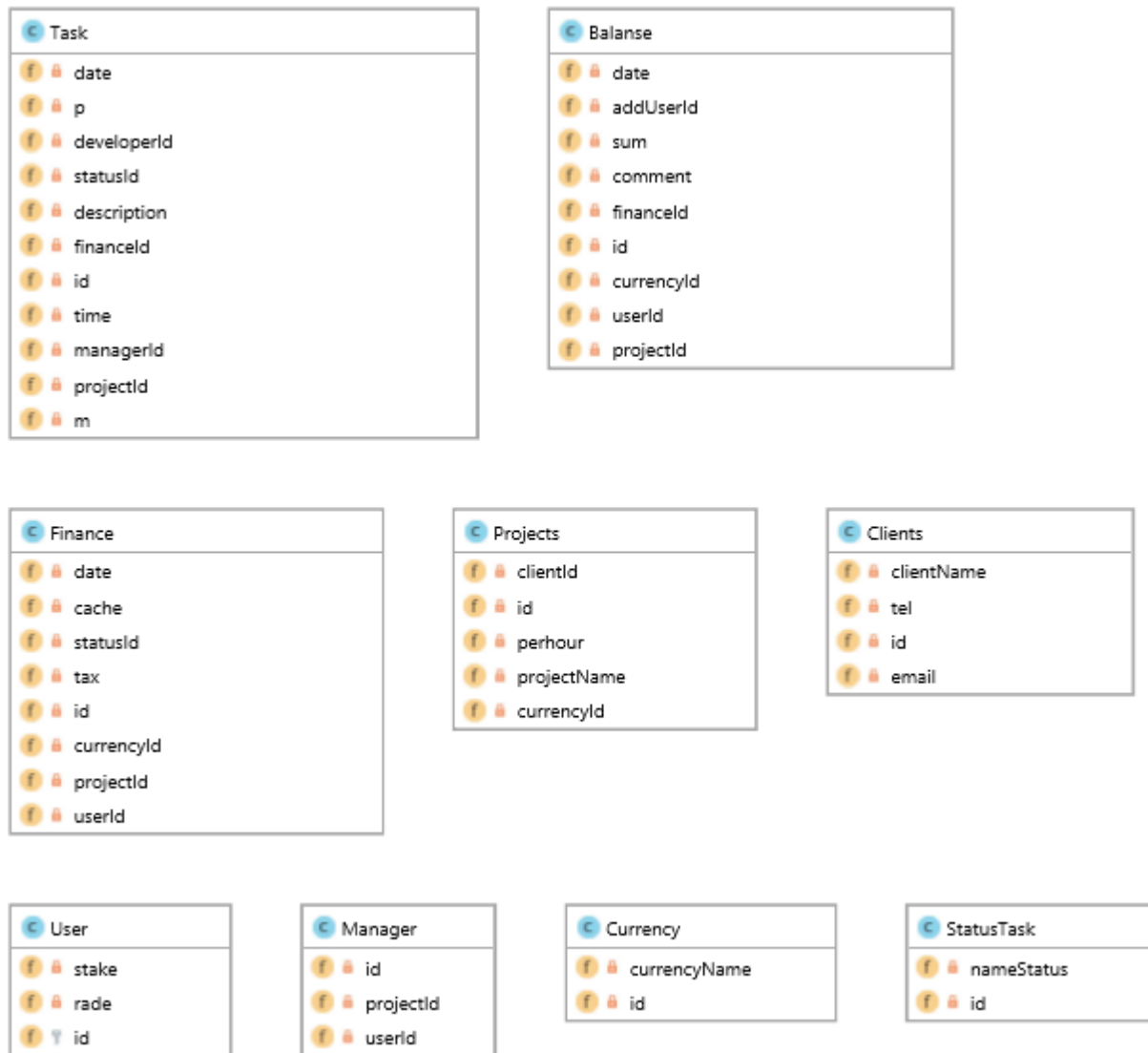


Рисунок 2.4 – UML-диаграмма

2.3 Проектирование информационного обеспечения системы

Информационное обеспечение приложения будет представлять собой базу данных. База данных будет содержать девять сущностей (таблиц). В логическую модель базы данных включены такие таблицы, как:

- баланс (balance);
- клиенты (clients);
- валюта (currency);
- финансы (finance);
- пользователи (fos_user);

- менеджеры (manager);
- проекты (projects);
- статус задачи(status_task);
- задачи (task).


Описание сущности balanse представлено в таблице 2.1.

Таблица 2.1 – Описание сущности balanse

Имя сущности	Описание сущности	Псевдонимы	Особенности использования
balanse	Объект, содержащий информацию о балансе: id пользователя, сумму, id валюты, дату комментариев, id финансов, id, id проекта, id пользователя добавившего сумму	Баланс	Используется для хранения и предоставления данных о балансе

Свойства сущности выступают в роли атрибутов в базе данных, они представлены в таблице описания атрибутов по сущности balanse (таблица 2.2).

Таблица 2.2 – Описание атрибутов по сущности balanse

#	Имя	Тип данных	Длина/Знач...	Беззна...	Разреш...	Zerofill	По умолчанию
 1	id	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT
2	sum	DOUBLE		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Нет значения по ...
3	currency_id	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Нет значения по ...
4	date	DATE		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	NULL
5	comment	VARCHAR	255	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	NULL
6	finance_id	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
7	user_id	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Нет значения по ...
8	project_id	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
9	add_user_id	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL

Описание сущности clients представлено в таблице 2.3.

Таблица 2.3 – Описание сущности clients

Имя сущности	Описание сущности	Псевдонимы	Особенности использования
clients	Объект, содержащий информацию о клиентах: id, имя, e-mail и телефон	Clients	Используется для хранения и предоставления данных о клиентах

Свойства сущности выступают в роли атрибутов в базе данных, они представлены в таблице описания атрибутов по сущности clients (таблица 2.4).

Таблица 2.4 – Описание атрибутов по сущности clients

#	Имя	Тип данных	Длина/Знач...	Беззна...	Разреш...	Zero fill	По умолчанию
1	id	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT
2	client_name	VARCHAR	255	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Нет значения по ...
3	email	VARCHAR	255	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	NULL
4	tel	VARCHAR	255	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	NULL

Описание сущности currency представлено в таблице 2.5.

Таблица 2.5 – Описание сущности currency

Имя сущности	Описание сущности	Псевдонимы	Особенности использования
currency	Объект, содержащий информацию о валюте: id, имя валюты	Валюта	Используется для хранения и предоставления данных о валюте

Свойства сущности выступают в роли атрибутов в базе данных, они представлены в таблице описания атрибутов по сущности currency (таблица 2.6).

Таблица 2.6 – Описание атрибутов по сущности currency

#	Имя	Тип данных	Длина/Знач...	Беззна...	Разреш...	Zero fill	По умолчанию
1	id	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT
2	currency_name	VARCHAR	255	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Нет значения по ...


Описание сущности finance представлено в таблице 2.7.

Таблица 2.7 – Описание сущности finance

Имя сущности	Описание сущности	Псевдонимы	Особенности использования
finance	Объект, содержащий информацию о финансах: id, id проекта, cache, id валюты, дата, статус, тариф, id пользователя	Финансы	Используется для хранения и предоставления данных о финансах

Свойства сущности выступают в роли атрибутов в базе данных, они представлены в таблице описания атрибутов по сущности finance (таблица 2.8).

Таблица 2.8 – Описание атрибутов по сущности finance

#	Имя	Тип данных	Длина/Знач...	Беззна...	Разреш...	Zerofill	По умолчанию
 1	id	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT
2	project_id	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Нет значения по ...
3	cache	DOUBLE		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Нет значения по ...
4	currency_id	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Нет значения по ...
5	date	DATE		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Нет значения по ...
6	status_id	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Нет значения по ...
7	tax	DOUBLE		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
8	user_id	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL




Описание сущности fos_user представлено в таблице 2.9.

Таблица 2.9 – Описание сущности fos_user

Имя сущности	Описание сущности	Псевдонимы	Особенности использования
fos_user	Объект, содержащий информацию о пользователях: id, имя, почта, подтверждение, пароль, последний вход, токен, запрос на восстановление, роль, коэффициент, ставка	Пользователь и	Используется для хранения и предоставления данных о пользователях

Свойства сущности выступают в роли атрибутов в базе данных, они представлены в таблице описания атрибутов по сущности fos_user (таблица 2.10).

Таблица 2.10 – Описание атрибутов по сущности fos_user

#	Имя	Тип данных	Длина/Знач...	Беззна...	Разреш...	Zerofill	По умолчанию
 1	id	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT
2	username	VARCHAR	255	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Нет значения по ...
 3	username_canonical	VARCHAR	255	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Нет значения по ...
4	email	VARCHAR	255	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Нет значения по ...
 5	email_canonical	VARCHAR	255	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Нет значения по ...
6	enabled	TINYINT	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Нет значения по ...
7	salt	VARCHAR	255	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Нет значения по ...
8	password	VARCHAR	255	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Нет значения по ...
9	last_login	DATETIME		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	NULL
10	locked	TINYINT	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Нет значения по ...
11	expired	TINYINT	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Нет значения по ...
12	expires_at	DATETIME		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	NULL
13	confirmation_token	VARCHAR	255	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	NULL
14	password_requested_at	DATETIME		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	NULL
15	roles	LONGTEXT		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Нет значения по ...
16	credentials_expired	TINYINT	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Нет значения по ...
17	credentials_expire_at	DATETIME		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	NULL
18	rade	DOUBLE		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	100
19	stake	DOUBLE		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL

Описание сущности manager представлено в таблице 2.11.

Таблица 2.11 – Описание сущности manager

Имя сущности	Описание сущности	Псевдонимы	Особенности использования
manager	Объект, содержащий информацию о менеджерах: пользователь, экзамен, вопрос, статус, ответ пользователя.	Менеджеры	Используется для хранения и предоставления данных о менеджерах.

Свойства сущности выступают в роли атрибутов в базе данных, они представлены в таблице описания атрибутов по сущности manager (таблица 2.12).

Таблица 2.12 – Описание атрибутов по сущности manager

#	Имя	Тип данных	Длина/Знач...	Беззна...	Разреш...	Zero fill	По умолчанию
1	id	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT
2	project_id	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Нет значения по ...
3	user_id	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Нет значения по ...

Описание сущности projects в таблице 2.13.

Таблица 2.13 – Описание сущности projects

Имя сущности	Описание сущности	Псевдонимы	Особенности использования
projects	Объект, содержащий информацию о проектах: id, имя, часы, id валюты, id клиента	Проекты	Используется для хранения и предоставления данных о проектах

Свойства сущности выступают в роли атрибутов в базе данных, они представлены в таблице описания атрибутов по сущности projects (таблица 2.14).

Таблица 2.14 – Описание атрибутов по сущности users

#	Имя	Тип данных	Длина/Знач...	Беззна...	Разреш...	Zero fill	По умолчанию
1	id	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT
2	project_name	VARCHAR	255	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Нет значения по ...
3	perhour	DOUBLE		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Нет значения по ...
4	currency_id	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Нет значения по ...
5	client_id	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Нет значения по ...

Описание сущности status_task в таблице 2.15.

Таблица 2.15 – Описание сущности ststus_task

Имя сущности	Описание сущности	Псевдонимы	Особенности использования
ststus_task	Объект, содержащий информацию о задачах: id, имя	Статус	Используется для хранения и предоставления данных о статусе задачи

Свойства сущности выступают в роли атрибутов в базе данных, они представлены в таблице описания атрибутов по сущности ststus_task (таблица 2.16).

Таблица 2.16 – Описание атрибутов по сущности ststus_task

#	Имя	Тип данных	Длина/Знач...	Беззна...	Разреш...	Zerofill	По умолчанию
1	id	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT
2	name_status	VARCHAR	255	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Нет значения по ...

Описание сущности task в таблице 2.17.

Таблица 2.17 – Описание сущности task

Имя сущности	Описание сущности	Псевдонимы	Особенности использования
Task	Объект, содержащий информацию о задачах: id, дата, время, описание, id проекта, id исполнителя, id менеджера, id статуса, id финансов	Задачи	Используется для хранения и предоставления данных о задачах

Свойства сущности выступают в роли атрибутов в базе данных, они представлены в таблице описания атрибутов по сущности task (таблица 2.18).

Таблица 2.18 – Описание атрибутов по сущности task

#	Имя	Тип данных	Длина/Знач...	Беззна...	Разреш...	Zerofill	По умолчанию
1	id	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT
2	date	DATE		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Нет значения по ...
3	time	DOUBLE		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Нет значения по ...
4	description	VARCHAR	2048	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Нет значения по ...
5	project_id	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Нет значения по ...
6	developer_id	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Нет значения по ...
7	manager_id	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Нет значения по ...
8	status_id	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Нет значения по ...
9	finance_id	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
10	m	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
11	p	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL

БД в нормальной форме изображена на рисунке 2.3.

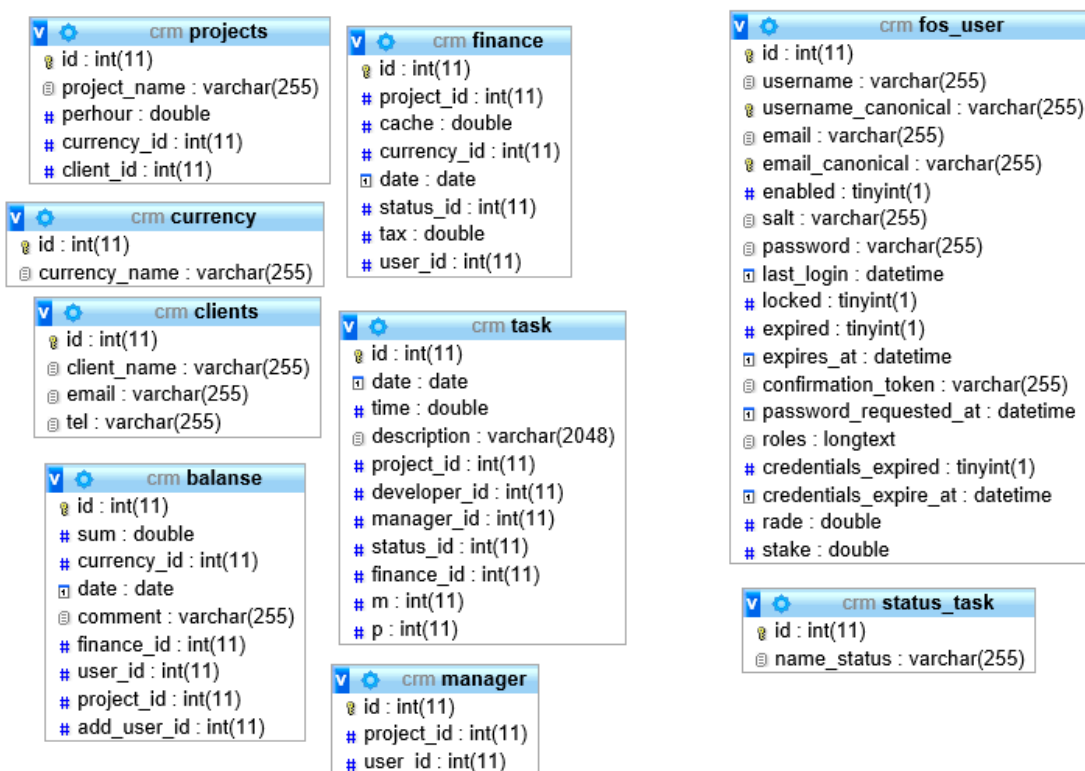


Рисунок 2.4 – БД в нормальной форме

2.4 Проектирование пользовательского интерфейса системы

В пользовательской части приложения предполагается наличие пользовательского интерфейса, посредством которого пользователи будут взаимодействовать с системой.

При открытии сервиса будет открываться страница входа/регистрации. На данной странице расположены две кнопки: «Вход» и «Регистрация». По умолчанию будет доступна форма для входа. На которой присутствуют поля: «Логин», «Пароль» и кнопка «Войти». Прототип этой страницы изображен на рисунке 2.4.

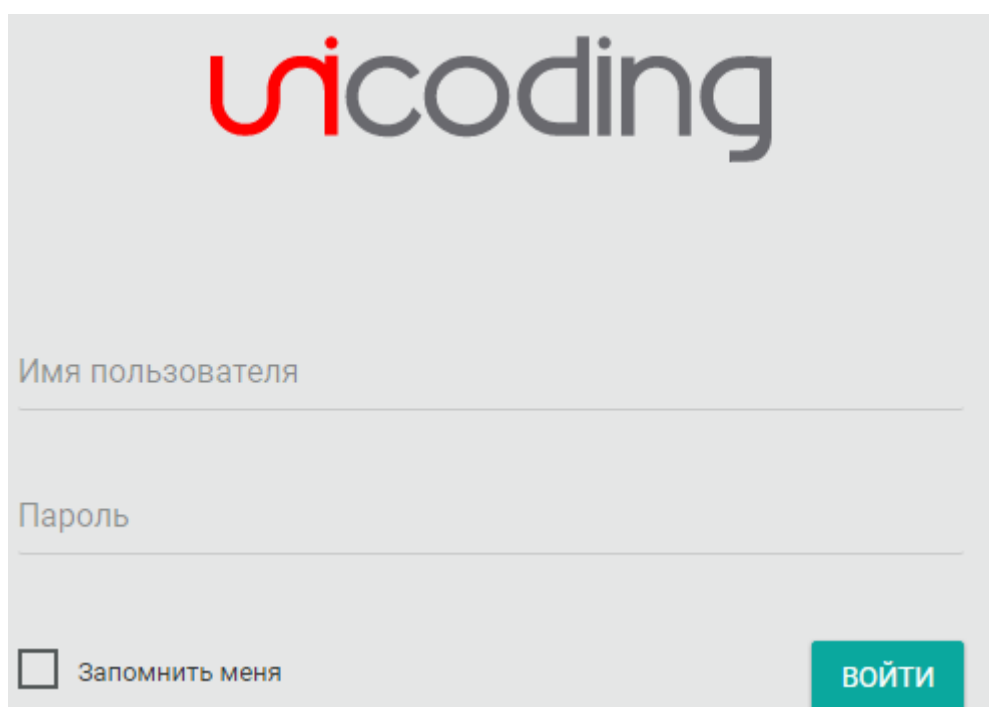
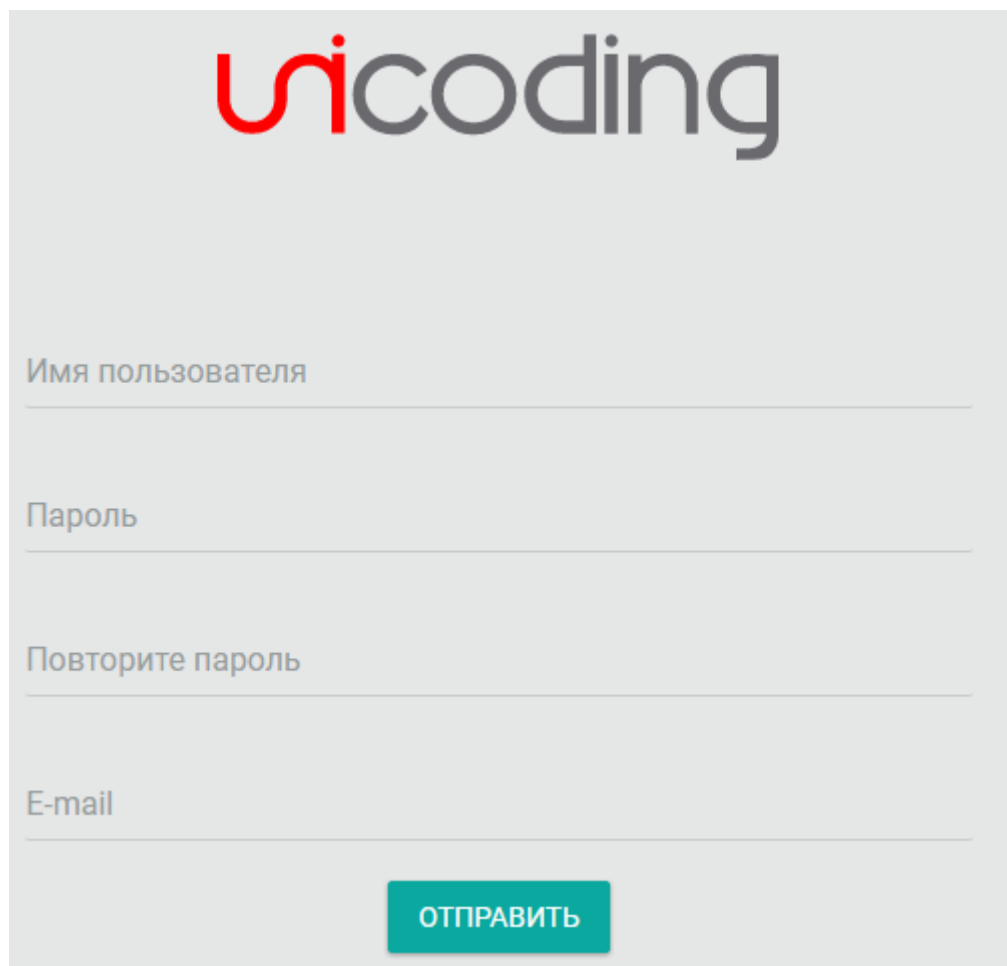


Рисунок 2.5 – Форма входа

При нажатии на кнопку «Регистрация» форма входа будет заменяться на форму регистрации. На которой присутствуют поля: «Имя», «Email», «Пароль», «Повторите пароля» и кнопка «Отправить». Прототип этой страницы изображен на рисунке 2.5.



unicoding

Имя пользователя

Пароль


Повторите пароль

E-mail

ОТПРАВИТЬ

Рисунок 2.6 – Форма регистрации

После входа в систему пользователь попадает в свой кабинет, где есть возможность перейти к доступным функциям. Первая из которых поле с выводом баланса на рисунке 2.7.



0 USD 0 RUB 0 BLN

Рисунок 2.7 – Баланс

Немного ниже расположена форма добавления записи по проекту, прототип на рисунке 2.8.

Добавить запись

Дата

2017-05-23

Время

Проект

proect_new

▼

Описание

ДОБАВИТЬ

Рисунок 2.8 – Добавить запись

Следующие поля задачи и таблица менеджера. Прототип этих полей изображен на рисунке 2.9.

Задачи

10

▼

показать строк

Q

Проект

↕

Дата

↕

Часы

↕

Описания

↕

Менеджер

↕

Статус

↕

↕

Нету данных в таблице

Показано 0 всего 0 из 0 полей

< >

Таблица менеджера

10

▼

показать строк

Q

Проект

↕

Стоимость часа

↕

Выполненные

↕

Выставленные

↕

1234567890	141 USD	22,00	0,00	📁 📄
123456789	15 BLN	8,00	0,00	📁 📄
21	21 USD	11,18	0,00	📁 📄

Показано 1 всего 3 из 3 полей

< 1 >

Рисунок 2.9 – Задачи и таблица менеджера

После нажатия на кнопку «Добавить клиента», пользователь попадет непосредственно на страницу добавления клиента. Прототип этой страницы изображен на рисунке 2.10.

Прототип страницы «Добавить клиента» имеет заголовок «Добавить клиента» на teal-фонной панели. Ниже расположены три текстовых поля: «Имя клиента», «E-mail» и «Телефон». В нижнем правом углу находится кнопка «ДОБАВИТЬ».

Рисунок 2.10 – Добавить клиента

Следующим пунктом меню является пункт «Добавить проект». Прототип этой страницы изображен на рисунке 2.11.

Прототип страницы «Добавить проект» имеет заголовок «Добавить проект» на teal-фонной панели. Ниже расположены пять текстовых полей: «Имя клиента» (с значением «cl1»), «Название проекта», «Стоимость часа», «Валюта» (с значением «USD») и «Имя менеджера» (с значением «albu»). В нижнем правом углу находится кнопка «ДОБАВИТЬ».

Рисунок 2.11 – Добавить проект

Все предыдущие функции пользователя, входят и в функционал администратора, внесем дополнения в интерфейс.

На главной странице добавляется таблица администратора. Прототип этой страницы изображен на рисунке 2.12.

Таблица Администратора

20

▼

показать строк

Q

Columns

Проект	⇅	Стоимость часа	⇅	Менеджер	⇅	Выполненные	⇅	Выставленные	▼
project_new		1001123 USD		alex		54,98		0,00	
зе		12211 RUB		alex		12,00		0,00	
гш		501 USD		alex		0,00		10,00	
test		60 BLN		alex		22,00		0,00	
test		700 RUB		alex		0,00		5,00	
1234567890		141 USD		albu		22,00		0,00	
123456789		15 BLN		albu		8,00		0,00	
21		21 USD		albu		11,18		0,00	

Показано 1 всего 8 из 8 полей

<

1

>

Рисунок 2.12 – Таблица администратора

Следующее дополнение в меню «Управление менеджерами». Прототип этой страницы изображен на рисунке 2.13.

Управление менеджерами

Проект	Ник	Действие
987	albu	
pr5	albu	
1234567890	albu	
гшг98	albu	
123456789	albu	

Рисунок 2.13 – Управление менеджерами

При выборе пункта «Проекты», попадаем на страницу где видим все проекты и баланс суммарный. Прототип этой страницы изображен на рисунке 2.14.

369.00 USD	1,308.00 RUB	123.00 BLN
Проекты		
10	показать строк	Q
Проекты	Стоимость часа	Валюта
progest_new	1001123	USD
ze	12211	RUB
гш	501	USD
test	60	BLN
test	700	RUB

Рисунок 2.14 – Проекты

При выборе пункта «Баланс» попадаем на страницу с суммарным балансом. Прототип этой страницы изображен на рисунке 2.15.

В верхней части страницы мы можем увидеть контейнеры, подобные тем, которые находятся на главной странице. Однако, это общий баланс зачислений системой на данном предприятии, после выполнения задачи, менеджер выставляет в систему время, затраченное на выполнение задания, проект, в рамках которого выполнялась данная задача, а также дата выполнения. Описание предназначено для того, чтобы менеджер мог конкретизировать, уточнить, над чем он конкретно работать. Это поле можно использовать, для полного описания области выполнения и любых комментариев, которые поймет руководитель проекта. Менеджер проекта в свою очередь, увидев отчет по выполненному заданию, может начислить финансовые бонусы, для исполнителя. Таким образом, администратор, который открывает раздел системы «Баланс» видит, какое количество средств, в конкретную дату ушла и какому пользователю, и за какой комментарий к своей задаче, он получил вознаграждение. Также в системе предусмотрено будущее расширение функционала, и как доказательство введен новый раздел «Номер задачи», где в перспективе будут отображаться номера задач, соответственно. Данное описание, доказывает объективность такого решения в интерфейсе

51,74 USD	10 000,00 RUB	500,00 BLN
-----------	---------------	------------

Баланс

10 ▾ показать строк

Q

Сумма	Должность	Дата	Комментарий	Номер задачи	Имя
0,52	USD	2017-03-29	auto		alex
0,52	USD	2017-03-29	auto		alex
0,70	USD	2017-03-28	auto		alex
50,00	USD	2017-02-23	good		user1
500,00	BLN	2017-02-23	ok		albu
10 000,00	RUB	2017-02-23	great		123

Показано 1 всего 6 из 6 полей

< 1 >

Рисунок 2.15 – Баланс

При выборе пункта «Пользователи» администратор попадает на страницу с информацией о всех пользователях. Прототип этой страницы изображен на рисунке 2.16

Список пользователей

Пользователи
123
123789
albu
alex
sfd
user1

Рисунок 2.16 – Пользователи

При выборе пункта «Коэффициенты» администратор попадает на страницу с информацией о зарплатных коэффициентах каждого пользователя. Прототип этой страницы изображен на рисунке 2.17.

Коэффициенты	
123	133
123789	56
albu	50
alex	70
sfd	
user1	15

Рисунок 2.17 – Коэффициенты

При нажатии на пункт «Ставки» администратор попадает на страницу просмотром и изменением зарплатных ставок. Прототип этой страницы изображен на рисунке 2.18.

Как и в разделе «Коэффициенты», в разделе «Ставка» видно, что каждому пользователю системы предусмотрен некий числовой параметр. В данном случае, это фиксированная зарплатная ставка каждого работника предприятия. Этот параметр позволяет разработать модуль, который в каждый новый рабочий день месяца будет начислять фиксированное значение к балансу исполнителя. Что избавит менеджера проекта, от ежедневного ручного отчисления каждому исполнителю задач по проекту. Он сможет заниматься только бонусными отчислениями, которые предусмотрены и описаны выше, где говорилось о балансе. То есть, система сама будет рассчитывать количество рабочих дней каждого месяца и распределять по ним ежедневные отчисления. В любой момент ставку, как и коэффициент можно изменять, повышая заработную плату исполнителя. Данное описание, доказывает объективность такого решения в интерфейсе.

Ставка
123
1233
123789
56
albu
12
alex
12
sfd
user1
8990

Рисунок 2.18 – Ставка

При выборе пункта «Клиенты» администратор попадает на страницу с формой для просмотра и отображения и редактирования всех клиентов и их проектов. Прототип этой страницы изображен на рисунке 2.19.

Клиент	Email	Телефон	Проекты
cl1	cl1@bla.bla	+123456789	<ul style="list-style-type: none"> pr1 pr2 987 pr5 project_new ze 1234567890 456 123132вп 45 12 73
cl10	cl10	34583475654	<ul style="list-style-type: none"> гш 123456789

Рисунок 2.19 –Клиенты

Рабочее время

Фильтр:

2017

ИЗМЕНИТЬ

10

▼

показать строк

Q

Номер месяца	Имя пользователя	Кол-во часов
Март	alex	84.18
Февраль	albu	30.33
Февраль	123	95.98

Рисунок 2.20 – Рабочее время

3. РЕАЛИЗАЦИЯ И ИСПЫТАНИЕ СИСТЕМЫ

3.1 Выбор средств реализации системы

Язык гипертекстовой разметки HTML. HTML (Hyper Text Markup Language) - язык гипертекстовой разметки, основной строительный блок веб-страниц, используется для создания и визуального представления веб-страниц. Он определяет содержание страницы, но не её функциональность.

Язык описания внешнего вида документа CSS. HTML добавляет разметку в обычный текст. Гипертекст содержит ссылки, которыми веб-страницы связываются друг с другом, делая Всемирную паутину тем, чем она является сегодня. Создавая и публикуя веб-страницы в сети интернет, вы становитесь активным участником Всемирной паутины. HTML поддерживает как изображения, так и другой медиа-контент. С помощью HTML каждый может создать статический, а также динамический сайт. HTML является языком, описывающим структуру и семантику содержимого веб-документа. Контент веб-страницы размечен с помощью тегов, представляющих HTML-элементы. Эти элементы формируют строительные блоки для любого веб-сайта.

CSS (Cascading Style Sheets) — это язык иерархических правил (таблиц стилей), используемый для представления внешнего вида документа, написанного на HTML или XML (включая различные языки XML, такие как SVG и XHTML). CSS описывает, каким образом элемент должен отображаться на экране, на бумаге, голосом или с использованием других медиа средств.

CSS является одним из основных языков свободной веб-разработки, который стандартизован спецификацией W3C. Стандарт CSS делится на уровни: CSS1 в настоящее время устарел, CSS2.1 — рекомендован для применения, а CSS3, разбитый на более мелкие модули, развивается на пути стандартизации.

Язык программирования JavaScript. JavaScript — это легкий, интерпретируемый, объектно-ориентированный язык с функциями первого класса. Наиболее широкое применение находит как язык сценариев веб-страниц, но также используется и в других программных продуктах, например, node.js или Apache CouchDB. JavaScript прототипно-ориентированный, мультипарадигмальный язык с динамической типизацией который поддерживает объектно-ориентированный, императивный и функциональный стили программирования.

Стандартом языка JavaScript является ECMAScript. Все современные браузеры полностью поддерживают ECMAScript 5.1. Старые версии браузеров поддерживают по крайней мере - ECMAScript 3. 17 июня 2015 года состоялся выпуск шестой версии

ECMAScript. Эта версия официально называется ECMAScript 2015, которую чаще всего называют ECMAScript 6 или просто ES6.

Основными конкурентными преимуществами языка программирования JavaScript, по праву являются:

- максимально понятен для пользователя;
- поддерживается наиболее популярными браузерами «по умолчанию»;
- очень высокая скорость работы JavaScript;
- скрипты (программы, написанные на языке JavaScript) подключаются к HTML коду веб-страницы напрямую и при загрузке сразу же выполняются;
- программы можно запускать не только в браузере, но и на сервере.

Язык программирования PHP. PHP – это широко используемый язык сценариев общего назначения с открытым исходным кодом. Говоря проще, PHP это язык программирования, специально разработанный для написания веб-приложений (сценариев), исполняющихся на веб-сервере. Синтаксис языка берет начало из C, Java и Perl. PHP достаточно прост для изучения.

Преимуществом PHP является предоставление веб-разработчикам возможности быстрого создания динамически генерируемых веб-страниц. Подробнее о преимуществах PHP можно узнать здесь. Важным преимуществом языка PHP перед такими языками, как языков Perl и C заключается в возможности создания HTML документов с внедренными командами PHP. Подробнее об этой возможности смотрите здесь. Значительным отличием PHP от какого-либо кода, выполняющегося на стороне клиента, например, JavaScript, является то, что PHP-скрипты выполняются на стороне сервера. Вы даже можете сконфигурировать свой сервер таким образом, чтобы HTML-файлы обрабатывались процессором PHP, так что клиенты даже не смогут узнать, получают ли они обычный HTML-файл или результат выполнения скрипта. PHP позволяет создавать качественные веб-приложения за очень короткие сроки, получая продукты, легко модифицируемые и поддерживаемые в будущем. Язык PHP постоянно совершенствуется, и ему наверняка обеспечено долгое доминирование в области языков веб-программирования, по крайней мере, в ближайшее время.

СУБД MySQL. MySQL является наиболее приспособленной для применения в среде веб СУБД (системой управления базами данных). Не секрет, что для исполнения приложений клиента на большинстве хостинг-площадок провайдеры предоставляют небольшое количество ресурсов (как вычислительных, так и дисковых). Поэтому для данного применения необходима высокоэффективная СУБД, обладающая при этом высокой надежностью.

Symfony — свободный фреймворк, написанный на PHP, который использует паттерн Model-View-Controller.

Symfony предлагает быструю разработку и управление веб-приложениями, позволяет легко решать рутинные задачи веб-программиста. Работает только с PHP 5 и

выше. Имеет поддержку множества баз данных (MySQL, PostgreSQL, SQLite или любая другая PDO-совместимая СУБД). Информация о реляционной базе данных в проекте должна быть связана с объектной моделью. Это можно сделать при помощи ORM инструмента. Symfony поставляется с двумя из них: Propel и Doctrine.

Symfony бесплатен и публикуется под лицензией MIT

По всем этим причинам MySQL стала незыблемым стандартом в области СУБД для веб, а теперь в ней развиваются возможности для использования ее в любых критичных бизнес-приложениях, то есть конкурирует на равных с такими СУБД таких производителей, как Oracle, IBM, Microsoft и Sybase.

Основные преимущества MySQL:

- многопоточность, поддержка нескольких одновременных запросов;
- оптимизация связей с присоединением многих данных за один проход;
- записи фиксированной и переменной длины;
- ODBC драйвер;
- гибкая система привилегий и паролей;
- гибкая поддержка форматов чисел, строк переменной длины и меток времени;
- интерфейс с языками C и Perl, PHP;
- быстрая работа, масштабируемость;
- совместимость с ANSI SQL;
- бесплатна в большинстве случаев;
- хорошая поддержка со стороны провайдеров услуг хостинга;
- быстрая поддержка транзакций через механизм InnoDB.

3.2 Программная реализация

В данном проекте мной были реализованы следующие компоненты.

DefaultController.php – основной контроллер приложения. Который содержит в себе весь функционал контроля над действиями, выборку из базы данных, контроль ajax функций, логику всех сущностей. Все функции контроллера отображены на рисунке 3.1

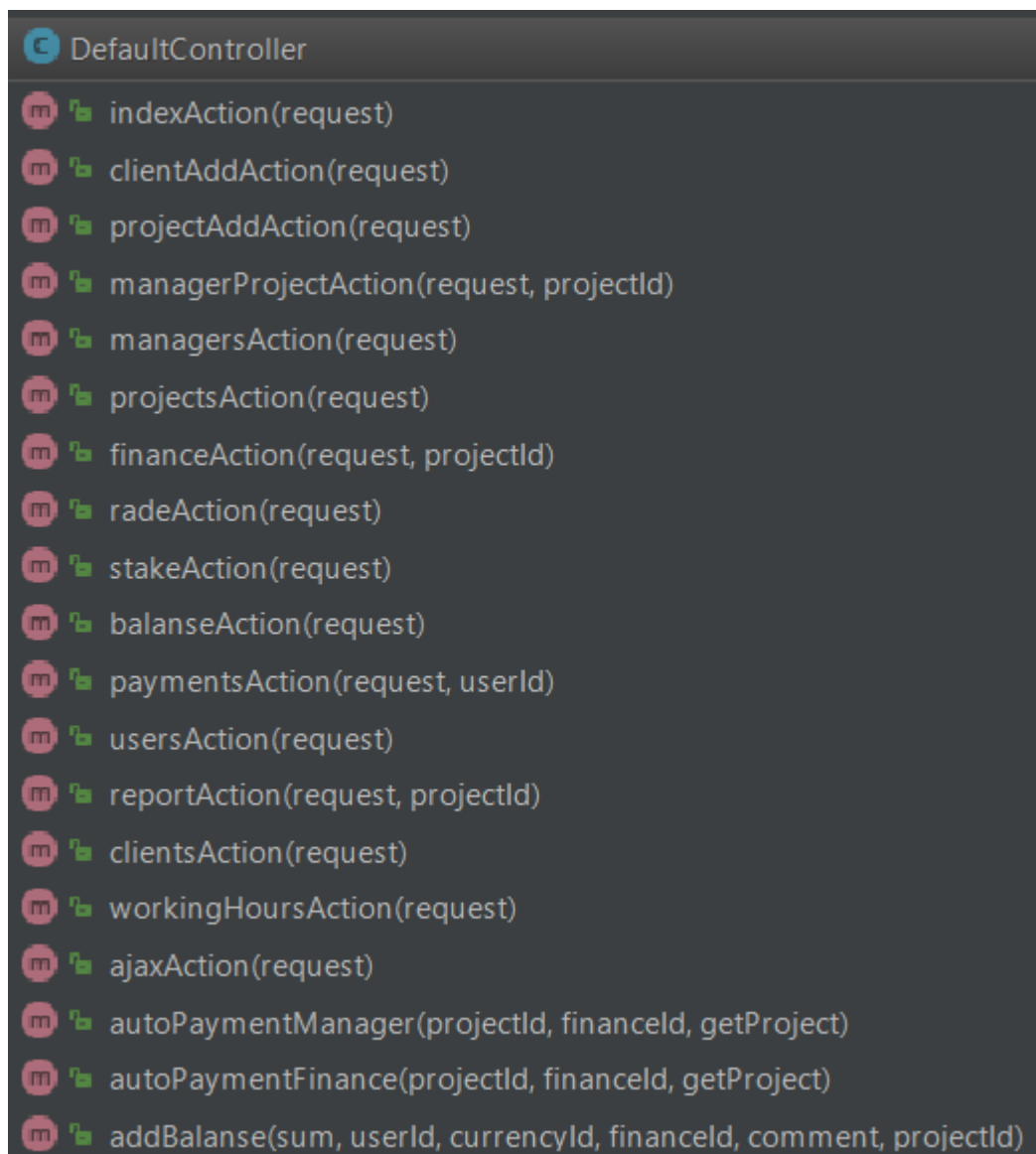


Рисунок 3.1 – Функции DefaultController.php

Из диаграммы видно, что многие методы являются контроллерами для сущностей из нашей системы, к примеру «ProjectAddAction», если обратить внимание на параметры, которые передаются в методы, то есть кроме, стандартного параметра по запросу, еще и другие. Например, чтобы обновить «balanse», мы должны явно передать в метод все параметры, которые нужны. Такой подход позволяет расширить обычный функционал контроллера, для предыдущих расширений возможностей системы. Стандартный метод, обязательно, возвращает путь к модели откуда шел запрос, а также может, что мы и наблюдаем чаще всего, возвращать массив параметров «ключ-значени».

Balanse.php – класс для управления балансом. Рисунок 3.2

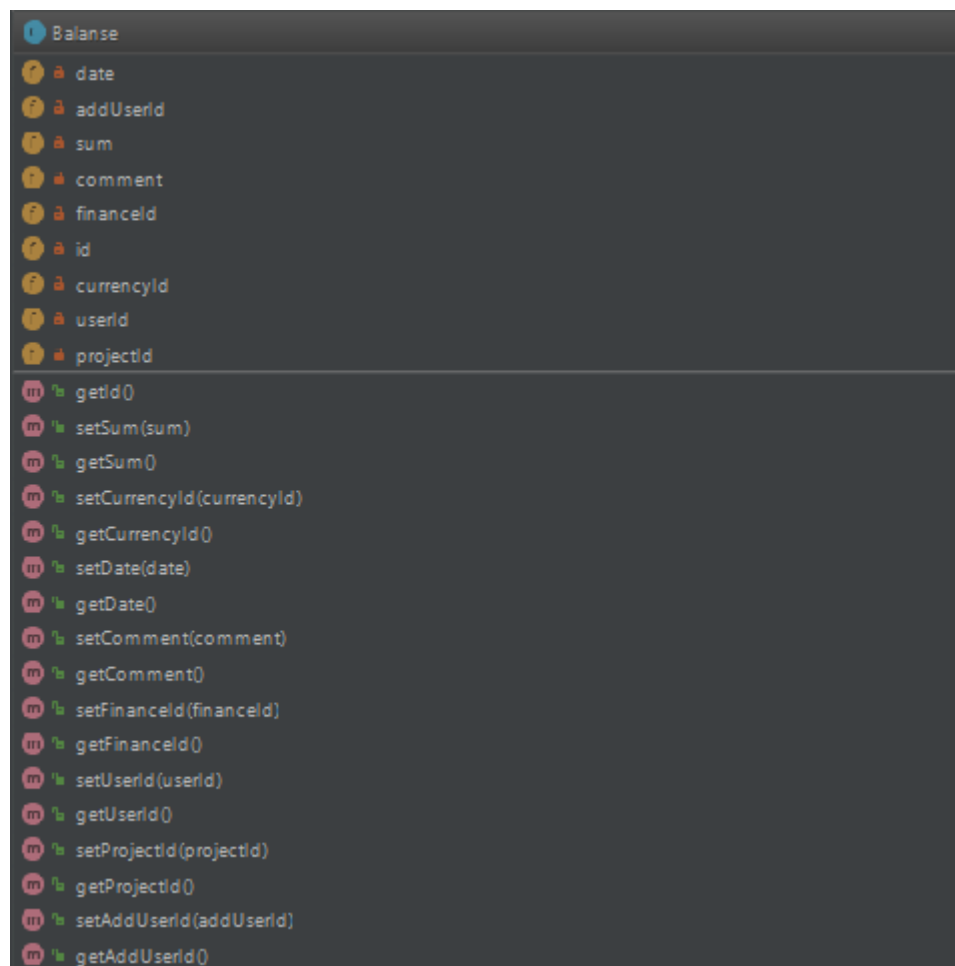


Рисунок 3.2 – Класс Balanse

Clients.php – Класс управления клиентами. Рисунок 3.3.

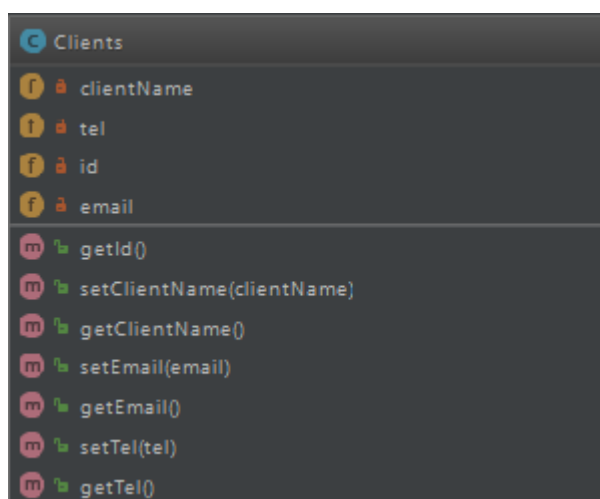


Рисунок 3.3 – Класс Clients

Currency.php – Класс управления валютами. Рисунок 3.4.

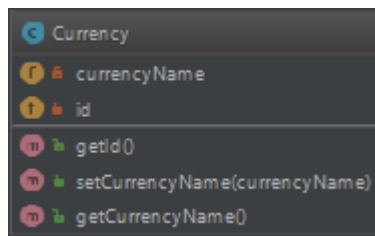


Рисунок 3.4- Класс Currency

Finance.php – Класс управления финансами. Рисунок 3.5.

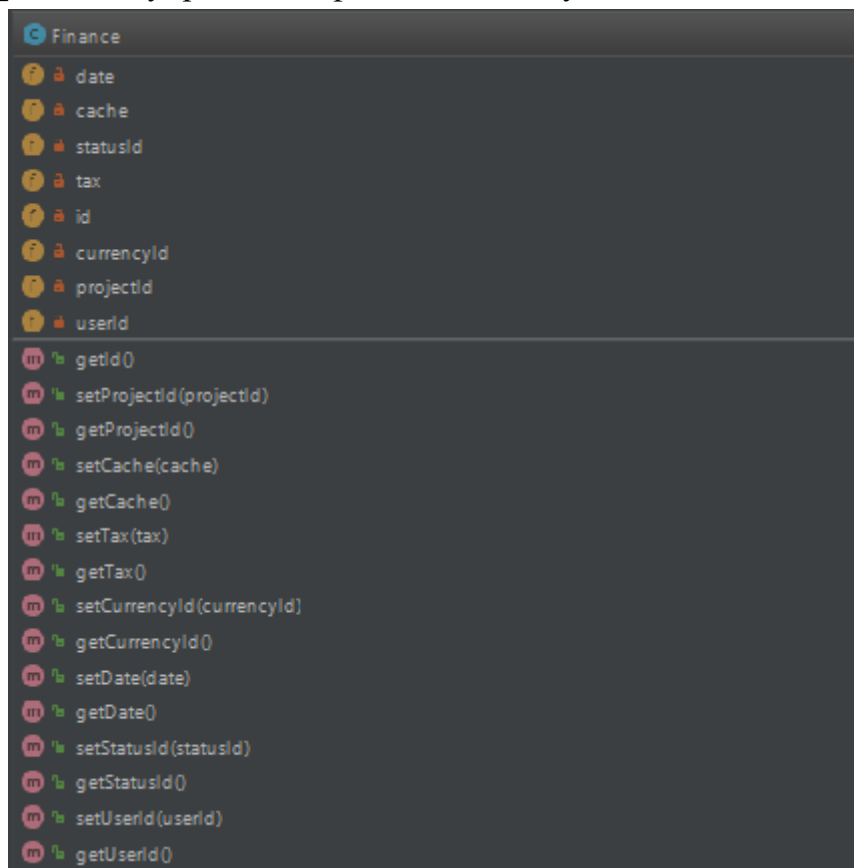


Рисунок 3.5 – Класс Finance

Manager.php – Класс управления менеджерами. Рисунок 3.6.

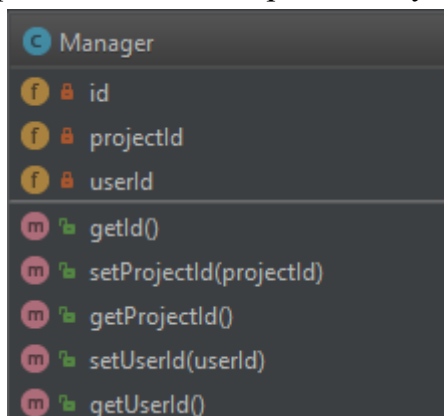


Рисунок 3.6 – Класс Manager

Projects.php – Класс управления сущностями проектов. Рисунок 3.7.

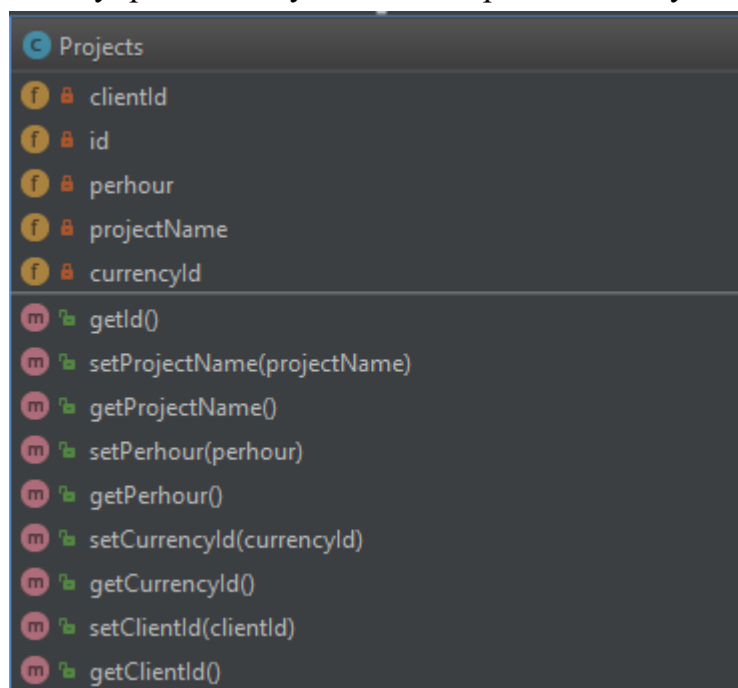


Рисунок 3.7. – Класс Projects

StatusTask.php – Класс состояния задачи. Рисунок 3.8.

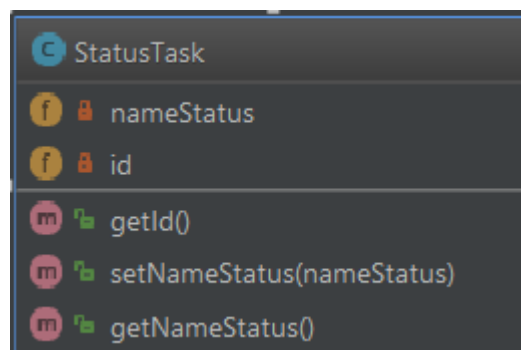


Рисунок 3.8. - Класс StatusTask

Task.php – Класс для управления сущностями задач. Содержит в себе глобальные переменные(поля), которые, собственно, как и в других классах, соответствуют полями соответствующей таблицы в базе данных. Данный класс является самым емким из-за концепции самих задач в нашей системе. Задача в проекте, связывает компоненты системы. Рисунок 3.9.

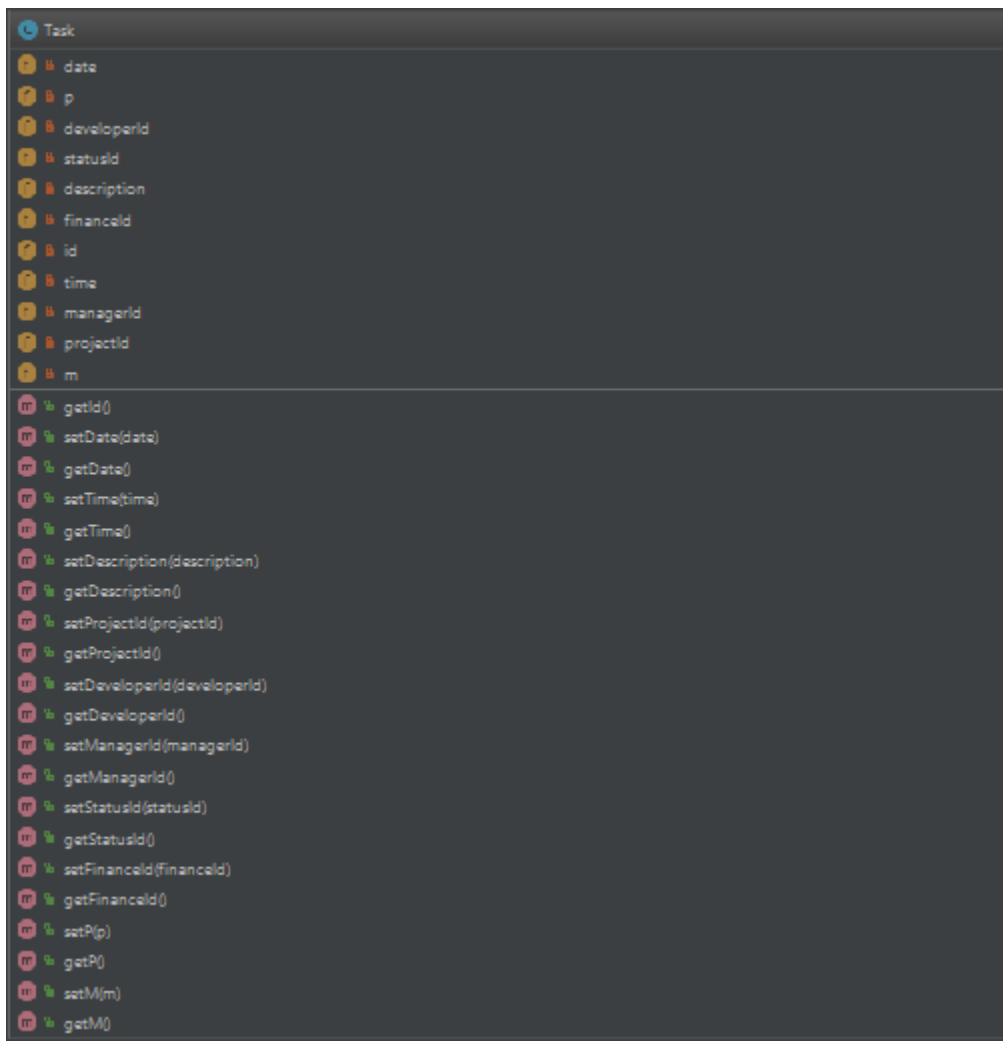


Рисунок 3.9. – Класс Task

User.php – Класс управления пользователями. Рисунок 3.10.

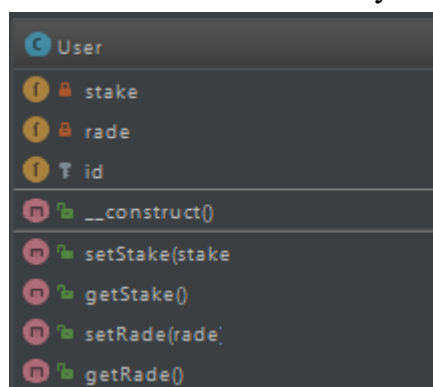


Рисунок 3.10 – Класс User

3.3 Испытание системы

Произведем проверку всех функций ПО и протестируем работоспособность всех функций в целом.

При вводе адреса сайта мы должны попасть на форму логина. Рисунок 3.11

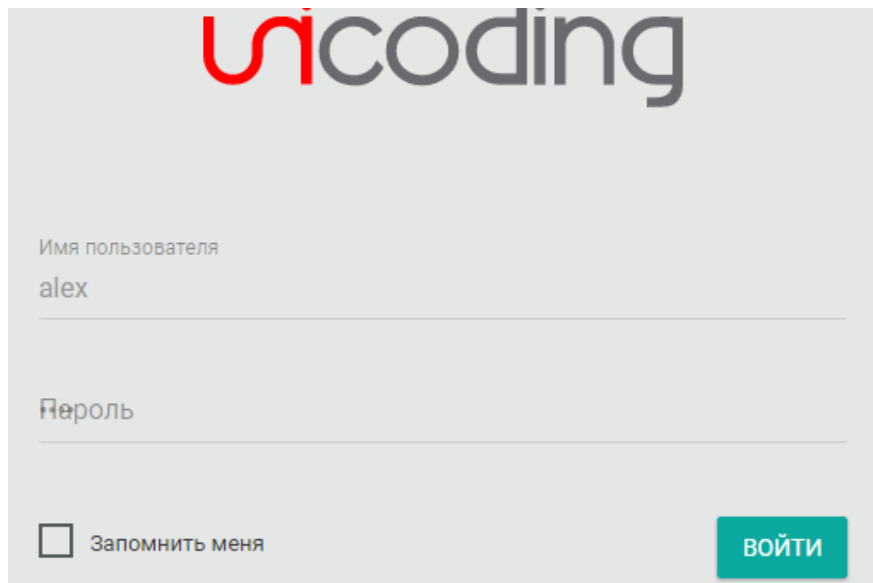
The image shows a login form for the 'unicoding' website. At the top is the 'unicoding' logo in red and grey. Below it are two input fields: 'Имя пользователя' (Username) with the text 'alex' entered, and 'Пароль' (Password). Below the password field is a checkbox labeled 'Запомнить меня' (Remember me). To the right of these fields is a green button labeled 'ВОЙТИ' (Login).

Рисунок 3.11-Вход в систему.

Если мы не зарегистрированы, то должны заполнить форму регистрации. Путь-
../register. Рисунок 3.12

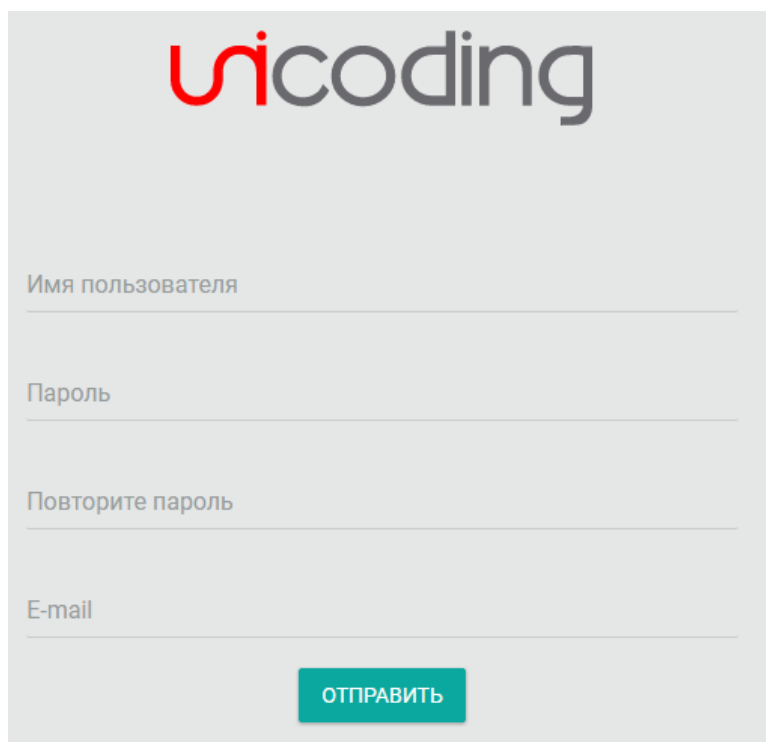
The image shows a registration form for the 'unicoding' website. At the top is the 'unicoding' logo in red and grey. Below it are four input fields: 'Имя пользователя' (Username), 'Пароль' (Password), 'Повторите пароль' (Repeat password), and 'E-mail'. At the bottom of the form is a green button labeled 'ОТПРАВИТЬ' (SEND).

Рисунок 3.12-Регистрация

После входа в систему видим главную страницу. Рисунок 3.13.

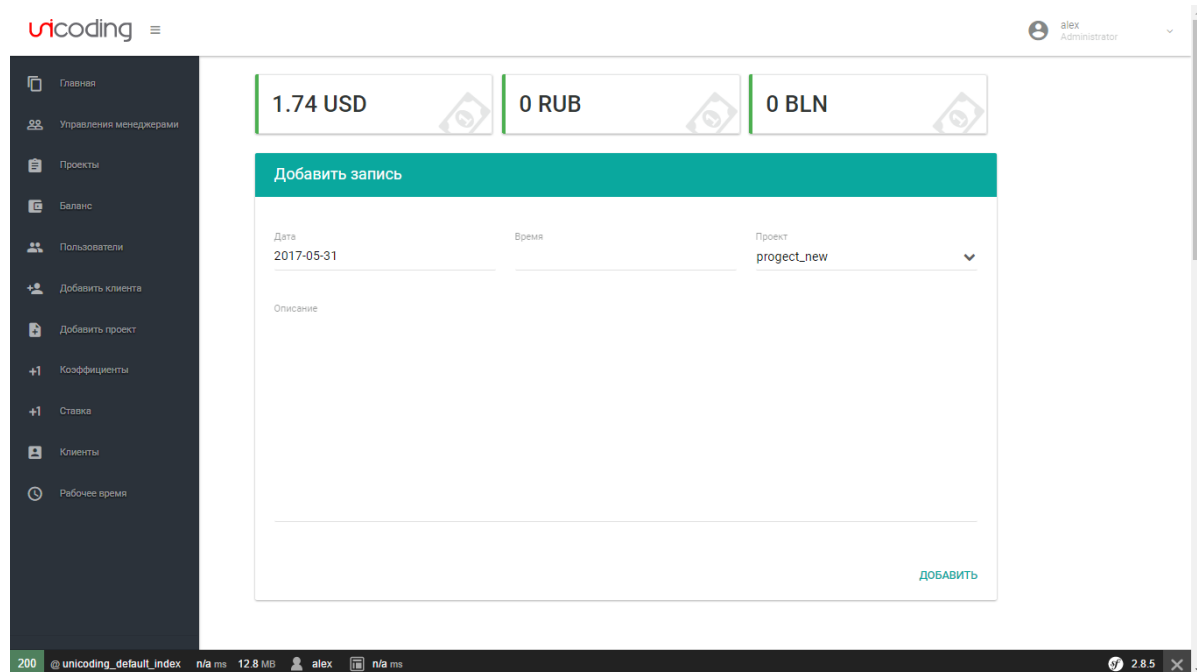


Рисунок 3.13 – Главная страница.

Протестируем все функции и будем сверять с записями в базе данных, если это будет необходимо. Все функции, которые работают у администратора, будут работать и у пользователя, т.к. исходя из архитектуры они наследуются для пользователя.

При нажатии на пункт «Управление менеджерами» видим следующее. Рисунок 3.14

Управление менеджерами

Проект	Ник	Действие
987	albu	
pr5	albu	
1234567890	albu	
gwr98	albu	

Рисунок 3.14 – Управление менеджерами

Попытаемся изменить менеджера для проекта «987». Результат Рисунок 3.15.

Управление менеджерами

Проект	Ник	Действие
987	123	
pr5	albu	
1234567890	albu	
гшг98	albu	

Рисунок 3.15 – Успешное изменение менеджера

Откроем пункт проекты и изменим валюту и стоимость для первого проекта в списке. Рисунок 3.16 и 3.17.

369.00 USD

1,308.00 RUB

123.00 BLN

Проекты

10 ▾ показать строк

Q





Проекты	Стоимость часа	Валюта	
progest_new	1001123	USD	
зе	12211	RUB	
гш	501	USD	
test	60	BLN	

Рисунок 3.16 – Проекты

10 ▼ показать строк

10

▼ показать строк

Проекты	Стоимость часа	Валюта	
project_new	100	RUB	
ze	12211	RUB	
гш	501	USD	
test	60	BLN	

Откроем вкладку «Баланс» и увидим отчетность по балансу фирмы. Рисунок 3.18

10 ▼ показать строк

10

▼ показать строк

Сумма	Должность	Дата	Комментарий	Номер задачи	Имя
0,52	USD	2017-03-29	auto		alex
0,52	USD	2017-03-29	auto		alex
0,70	USD	2017-03-28	auto		alex
50,00	USD	2017-02-23	good		user1
500,00	BLN	2017-02-23	ok		albu
10 000,00	RUB	2017-02-23	great		123

Показано 1 всего 6 из 6 полей

< 1 >

Откроем вкладку пользователи, где мы должны увидеть пользователей системы.

Рисунок 3.19

Список пользователей

Пользователи	
123	
123789	
albu	
alex	
sfd	
user1	

Рисунок 3.19 – Пользователи

Откроем вкладку «Добавить клиента» и заполним форму. Так-же сверим с записями в СУБД. Рисунок 3.20 и 3.21

Добавить клиента

Имя клиента

TestDiplomClient

E-mail

TestDiplomClient@example.com

Телефон

+777777777|

ДОБАВИТЬ

Рисунок 3.20 – Форма добавления клиента

<div><div>↔ T ↔</div><div>▼</div></div>				id	client_name	email	tel
<input type="checkbox"/>				1	cl1	cl1@bla.bla	+123456789
<input type="checkbox"/>				2	123	123@123.123	34569760123
<input type="checkbox"/>				3	cl2	cl2@cl.cl	789654123
<input type="checkbox"/>				4	cl10	cl10	34583475654
<input type="checkbox"/>				5	TestDiplomClient	TestDiplomClient@example.com	+777777777

Рисунок 3.21 – Сверка клиента в СУБД

После добавления клиента, система должна перенаправить нас на страницу добавления проекта. Что происходит автоматически, благодаря технологии ајах, которая сразу отправляет данные в базу. Заполним форму добавления проекта и сверим в СУБД. Рисунок 3.22 и 3.23

Добавить проект

Имя клиента

cl1

Название проекта

ProjectDiplom

Стоимость часа

15

Валюта

USD

Имя менеджера

alex

ДОБАВИТЬ

Рисунок 3.22 – Добавление проекта

Так как форма добавления проекта может быть вызвана, как прямо из меню так и после совершения действия «добавления клиента», так как клиенту, системой предусмотрено, обязательное наличие проекта, тестирование может происходить в двух этапах. Но именно наличие нового проекта в базе данных явно нам докажет что он теперь присутствует в системе. Теперь работа над проектом может быть произведена. Также в форме, из выпадающего списка мы можем выбрать валюта, идентификатор которой отображается в базе данных, производится ввод стоимости часа. Также в таблице присваивается идентификационный номер клиента, который связан с таблицей клиентов, имя которого и берется оттуда. Администратор после совершения операции добавления нового проекта в базу может изменять значения полей из смешных меню, которые работают с теми же полями в базе.

				id	project_name	perhour	currency_id	client_id
<input type="checkbox"/>				1	pr1	2112	0	1
<input type="checkbox"/>				2	pr2	600	0	1
<input type="checkbox"/>				3	987	1000	0	1
<input type="checkbox"/>				4	pr5	300	0	1
<input type="checkbox"/>				5	progetct_new	100	2	1
<input type="checkbox"/>				6	зе	12211	2	1
<input type="checkbox"/>				7	гш	501	1	4
<input type="checkbox"/>				8	test	60	3	3
<input type="checkbox"/>				9	test	700	2	3
<input type="checkbox"/>				10	1234567890	141	1	1
<input type="checkbox"/>				11	гшг98	252	2	2
<input type="checkbox"/>				12	123456789	15	3	4
<input type="checkbox"/>				13	456	56	1	1
<input type="checkbox"/>				14	123132вп	43	1	1
<input type="checkbox"/>				15	45	45	1	1
<input type="checkbox"/>				16	12	12	2	1
<input type="checkbox"/>				17	73	73	1	1
<input type="checkbox"/>				18	21	21	1	3
<input type="checkbox"/>				19	ProjectDiplom	15	1	1

Рисунок 3.23 – Добавление проекта в СУБД

Т.к. теперь мы можем проверить работоспособность добавления задачи к нашему проекту, заполнив задачу. Рисунок 3.25. Предварительно посмотрим, на вкладке «Рабочее время» текущие значения(Рисунок 3.24), после чего их сверим. Рисунок 3.26

Фильтр:

2017

ИЗМЕНИТЬ

10

показать строк

Q

Номер месяца	Имя пользователя	Кол-во часов
Март	alex	84.18
Февраль	albu	30.33
Февраль	123	95.98
Февраль	user1	15.00
Февраль	alex	54.65

Показано 1 всего 5 из 5 полей

<

1

>

Рисунок 3.24 – Начальные данные рабочего времени

1.74 USD

0 RUB

0 BLN

Добавить запись

Дата

2017-05-31

Время

12:00

Проект

ProjectDiplom

▼

Описание

Test task

ДОБАВИТЬ

Рисунок 3.25 – Добавление задачи

Фильтр:

2017

ИЗМЕНИТЬ

10 ▼ показать строк

Q

Номер месяца	Имя пользователя	Кол-во часов
Май	alex	12.00
Март	alex	84.18
Февраль	albu	30.33
Февраль	123	95.98
Февраль	user1	15.00
Февраль	alex	54.65

Рисунок 3.26 – Сверка рабочего времени

Как можем заметить, все функционирует корректно.

На главной странице посмотрим отобразилась ли информация в «Таблице администратора» Рисунок 3.27. И посмотрим список клиентов. Рисунок 3.28 Попробуем изменить данные. Рисунок 3.29.

Таблица Администратора

20	▼	показать строк		Q	Columns
Проект	Стоимость часа	Менеджер	Выполненные	Выставленные	
proect_new	100 RUB	alex	54,98	0,00	
зе	12211 RUB	alex	12,00	0,00	
гш	501 USD	alex	0,00	10,00	
test	60 BLN	alex	22,00	0,00	
test	700 RUB	alex	0,00	5,00	
1234567890	141 USD	albu	22,00	0,00	
123456789	15 BLN	albu	8,00	0,00	
21	21 USD	albu	11,18	0,00	
ProjectDiplom	15 USD	alex	12,00	0,00	

Показано 1 всего 9 из 9 полей

< 1 >

Рисунок 3.27 – Таблица Администратора

Клиенты

Клиент	Email	Телефон	Проекты	
cl1	cl1@bla.bla	+123456789	<ul style="list-style-type: none"> pr1 pr2 987 pr5 proect_new зе 1234567890 456 123132вп 45 12 73 ProjectDiplom 	
cl10	cl10	34583475654	<ul style="list-style-type: none"> гш 123456789 	
cl2	cl2@cl.cl	789654123	<ul style="list-style-type: none"> test test 21 	

Рисунок 3.28 – Клиенты

Клиенты




Клиент	Email	Телефон	Проекты	
cl12	cl12@bla.bla	+79878978	<ul style="list-style-type: none"> • pr1 • pr2 • 987 • pr5 • project_new • ze • 1234567890 • 456 • 123132вп • 45 • 12 • 73 • ProjectDiplom 	
cl10	cl10	34583475654	<ul style="list-style-type: none"> • гш • 123456789 	
cl2	cl2@cl.cl	789654123	<ul style="list-style-type: none"> • test • test • 21 	

Рисунок 3.29 – Изменение данных

Как видим, мы успешно изменили данные клиента 1(Имя, Почту, Телефон)

Теперь попытаемся не правильно заполнить форму входа и регистрации и посмотрим на реакцию системы. Рисунок 3.30 – 3.31. Так как все формы обрабатывает один шаблон, каждая форма в системе будет иметь такой-же уровень защиты.

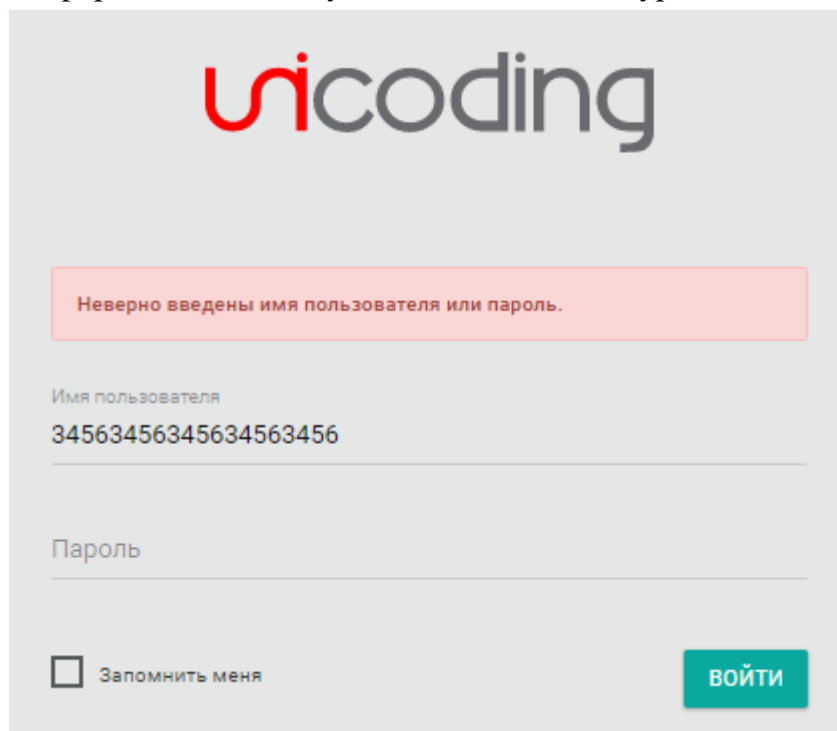


Рисунок 3.30 – Ошибка Входа

The image shows a registration form for 'unicoding'. The form has four input fields: a username field containing 'sdferwerfwewf', a password field with masked characters '*****', a second password field with masked characters '*****', and an email field containing 'fwefewfew'. Below the email field is a 'LOG' button. A red error message box is displayed over the email field, stating: '✖ Адрес электронной почты должен содержать символ "@". В адресе "fwefewfew" отсутствует символ "@".' (The email address must contain the symbol '@'. The address 'fwefewfew' is missing the symbol '@').

Рисунок 3.31 – Ошибка регистрации

4 ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ

4.1 Исходные данные для осуществления расчета

В данном дипломном проекте разрабатывается программное обеспечение, которое представляет собой web-приложение. Оно предназначено для поддержки процессов управления клиентами, проектами и задачами. Будет обеспечивать:

- ведение базы данных;
- комплекс задач по учету и анализу данных и проектах и ресурсах;
- комплекс задач по организации и планированию проектов;
- комплекс задач по управлению проектами;
- генерацию отчетов, формирование выходных документов, оперативной отчетности;
- комплекс задач, обеспечивающих управление доступом, сохранность, восстанавливаемость информации, авторизацию пользователей, управление профилями пользователей.

В наше время существуют подобные решения, но их функционал настолько широк, что является избыточным и трудозатратным для небольших компаний и студий не существует систем обладающих только основным функционалом и с минимальными требованиями к сервер, что делает актуальным разработку такой системы. Заказчиком является УО «Брестский Государственный Технический Университет». Система разрабатывается исключительно для нужд БрГТУ, и не будет свободно распространяться на рынке.

Расчеты будут произведены в следующем порядке:

- 1) Расчет объема функций программного модуля.
- 2) Расчет полной себестоимости.
- 3) Расчет отпускной цены и чистой прибыли.

4.2 Расчет объема функций программного обеспечения

Наименование проекта – «разработка программно-информационного обеспечения для поддержки процессов управления проектами».

Среда разработки ПО – JetBrains PhpStorm IDE (PHP). ПО функционального назначения.

Определение общего объема ПО.

Общий объем ПО (V_o) определяется исходя из количества и объема функций, реализуемых программой:

$$V_o = \sum_{i=1}^n V_i, \quad (4.1)$$

где V_o – общий объем ПС;

V_i – объем функций ПС;

n – общее число функций.

Расчет общего объема ПО (количества строк исходного кода (LOC)) предполагает определение объема по каждой функции. В том случае, когда на стадии технико-экономического обоснования проекта невозможно рассчитать точный объем функций, данный объем может быть получен на основании ориентировочной (прогнозной) оценки имеющихся фактических данных по аналогичным проектам, выполненным ранее, или путем применения нормативов по каталогу функций.

Определение уточненного объема ПО.

На основании информации о функциях разрабатываемого ПО по каталогу функций определяется общий объем ПО. В зависимости от организационных и технологических условий, в которых разрабатывается ПО, корректируется объем на основе экспертных оценок.

Уточненный объем ПО (V_y) определяется по формуле:

$$V_y = \sum_{i=1}^n V_{yi}, \quad (4.2)$$

где V_{yi} – уточненный объем отдельной функции в строках исходного кода (LOC).

В таблице 4.1 приведен перечень и объем функций ПО.

Таблица 4.1 – Перечень и объем функций программного обеспечения

Номер функции	Содержание функций	Объем функции, LOC	
		по каталогу (V_i)	уточненный (V_{yi})
1	2	3	4
101	Организация ввода информации	130	60
102	Контроль, предварительная обработка и ввод информации	490	210
202	Формирование баз данных	1980	600
203	Обработка наборов и записей базы данных	2370	1800
206	Манипулирование данными	7860	3040
506	Обработка ошибочных и сбойных ситуаций	1540	100

Продолжение таблицы 4.1

1	2	3	4
507	Обеспечение интерфейса между компонентами	1680	1400
601	Проведение тестовых испытаний прикладных программ в интерактивном режиме	3780	980
707	Графический вывод результатов	420	410
Итого:		20250	8600

Учитывая информацию, указанную в таблице 4.1, о функциях разрабатываемого программного обеспечения, уточненный объем ПО (V_y) составил 8600 строк исходного кода (LOC) вместо предполагаемого количества строк 20250. Что говорит о значительной разнице объема строк между значением по каталогу и получившимся. Уточненное значение в 2 раза меньше значения по каталогу.

4.3 Расчет полной себестоимости программного обеспечения

Стоимостная оценка программного обеспечения у разработчика предполагает составление сметы затрат, которая включает следующие статьи расходов:

- заработную плату исполнителей (основную – ЗПо и дополнительную – ЗПд);
- отчисления на социальные нужды ($P_{соц}$);
- материалы и комплектующие изделия (P_m);
- спецоборудование (P_c);
- машинное время (P_{mv});
- расходы на научные командировки ($P_{нк}$);
- прочие прямые расходы ($P_{пр}$);
- накладные расходы ($P_{нр}$);
- затраты на освоение и сопровождение программного средства (P_o и $P_{со}$).

Полная себестоимость (C_p) разработки программного обеспечения (ПО) рассчитывается как сумма расходов по всем статьям с учетом рыночной стоимости аналогичных продуктов.

Основной статьей расходов на создание ПО является заработная плата (основная и дополнительная) разработчиков (исполнителей) ($ЗПо + ЗПд$), необходимых для решения специальных задач в команде.

Расчёт заработной платы разработчиков ПО начинается с определения:

- продолжительности времени разработки Фрв, которое устанавливается студентом экспертным путём с учётом сложности, новизны ПО и фактически затраченного времени. В данном дипломном проекте инженер-программист 2-категории работал на ПО 45 дней(Фрв), а руководитель проекта 5 дней(Фрв).
- количества разработчиков ПО. В данном дипломном проекте два разработчика (инженер-программист 2-категории), который работал весь период разработки ПО и руководитель проекта, который работал 5 дней.

Заработная плата разработчиков определяется как сумма основной и дополнительной заработной платы всех исполнителей.

Основная заработная плата каждого исполнителя определяется по формуле:

$$ЗП_O = T_{CT1P} \cdot T_K / 22 \cdot \Phi_{PB} \cdot K_{ПР}, \quad (4.3)$$

где T_{CT1P} – месячная тарифная ставка 1-го разряда ($T_{M1} = 31$ руб.);

T_K – тарифный коэффициент, соответствующий установленному тарифному разряду;

22 – среднее количество рабочих дней в месяце;

Φ_{PB} – фонд рабочего времени исполнителя (продолжительность разработки ПО, дни);

$K_{ПР}$ – коэффициент премий ($K_{ПР} = 1,3$).

Тарифный коэффициент Инженера-программиста 2 категории согласно 11 разряду $T_K = 2,65$.

Тарифный коэффициент Руководителя проекта согласно 14 разряду $T_K = 3,25$.

Основная заработная плата инженера-программиста ($K_{ПР} = 1,3$):

$$ЗП_O = 31 \cdot 2,65 / 22 \cdot 45 \cdot 1,3 = 218,44 \text{ руб.}$$

Основная заработная плата руководителя проекта ($K_{ПР} = 1,3$):

$$ЗП_O = 31 \cdot 3,25 / 22 \cdot 5 \cdot 1,3 = 29,77 \text{ руб.}$$

Дополнительная заработная плата каждого исполнителя ($H_{Доп.ЗП}$) – 20 %. Рассчитывается от основной заработной платы по формуле:

$$ЗП_Д = ЗП_O \cdot (H_{Доп.ЗП} / 100) \quad (4.4)$$

Дополнительная заработная плата инженера-программиста:

$$ЗП_Д = 218,44 \cdot 20 / 100 = 43,69 \text{ руб.}$$

Дополнительная заработная плата руководителя проекта:

$$ЗП_Д = 29,77 \cdot 20 / 100 = 5,95 \text{ руб.}$$

Результаты вычислений внесём в таблицу 4.2.

Таблица 4.2 – Расчет заработной платы

Категории работников	Разряд	Тарифный коэффициент (К _т)	Фонд рабочего времени (Ф _{рв}) дн.	Коэффициент премирования (К _{пр})	Норматив дополнительной заработной платы (Н _{доп.зп.}) %	Зараб. плата, руб.		
						Основная	Дополнительная	Всего
1	2	3	4	5	6	7	8	9
Инженер-программист	11	2,65	45	1,3	20	218,44	43,69	262,13
Руководитель проекта	14	3,25	5	1,3	20	29,77	5,95	35,75
ИТОГО						248,21	49,64	297,88

Таким образом, как видно из таблицы, заработная плата инженера-программиста 262,13 (руб.), а руководителя проекта 35,75(руб.).

Отчисления на социальные нужды (Р_{соц}) определяются в соответствии с действующими законодательными актами по нормативу (34% - отчисления в ФСЗН + 0,6% отчисления по обязательному страхованию):

$$P_{\text{соц}} = (ЗП_{\text{о}} + ЗП_{\text{д}}) \cdot \frac{34,6}{100}, \quad (4.5)$$

$$P_{\text{соц}} = (262,13 + 35,75) \cdot 34,6 / 100 = 103,07 \text{руб.}$$

Расходы по статье «Спецоборудование» (Рс) включает затраты на приобретение технических и программных средств специального назначения, необходимых для разработки ПО, включая расходы на проектирование, изготовление, отладку и др. В данном дипломном проекте для разработки ПО приобретение какого-либо спецоборудования не предусматривалось. Так как спецоборудование не было приобретено, данная статья не рассчитывается.

По статье «Материалы и комплектующие изделия» (Рм) отражаются расходы на бумагу, картридж и красящие ленты для лазерного принтера Canon LBP600B,

необходимые для разработки ПО. Норма расхода материалов в суммарном выражении определяются в расчете на 100 строк исходного кода. Сумма затрат на расходные материалы определяется по формуле():

$$P_M = 3\Pi_o \frac{H_{M3}}{100}, \quad (4.6)$$

$$P_M = 262,13 \cdot 3 / 100 = 7,86 \text{ руб.}$$

Расходы по статье «Машинное время» (Р_{МВ}) включают оплату машинного времени, необходимого для разработки и отладки ПО. Они определяются в машино-часах по нормативам на 100 строк исходного кода машинного времени.

$$P_{MB} = C_{Mi} \frac{V_o}{100} H_{MB}, \quad (4.7)$$

где C_{Mi} – цена одного машино-часа, тыс. руб. (1 руб.);

V_o – уточненный общий объём функций строк исходного кода (ЛОС);

H_{MB} – норматив расхода машинного времени на отладку 100 строк исходного кода, машино-часов. Принимается в размере 0,7.

$$P_{MB} = 1 \cdot 8600/100 \cdot 0,7 = 60,2 \text{ руб.}$$

Расходы по статье «Прочие прямые затраты» (Р_{ПР}) включают в себя затраты на приобретение и подготовку специальной научно-технической информации и специальной литературы и определяются по формуле:

$$P_{PR} = 3\Pi_o \frac{H_{ПЗ}}{100}, \quad (4.8)$$

где $H_{ПЗ}$ – норматив прочих затрат.

Так как никакая специальная научно-техническая информация и специальная литература не приобреталась, то данная статья не рассчитывается.

Затраты по статье «Накладные расходы» (Р_{НР}) связаны с содержанием вспомогательных хозяйств, а также с расходами на общехозяйственные нужды. Определяются по нормативу в процентах к основной заработной плате:

$$P_{NR} = 3\Pi_o \frac{H_{НР}}{100}, \quad (4.9)$$

где $H_{НР}$ – норматив накладных расходов, примем $H_{НР} = 100\%$.

$$P_{NR} = 248,21 \cdot 100 / 100 = 248,21 \text{ руб.}$$

Сумма вышеперечисленных расходов по статьям на ПО служит исходной базой для расчета затрат на освоение и сопровождение ПО:

$$\text{Сумма затрат} = 3\Pi_o + 3\Pi_d + P_{соц} + P_M + P_c + P_{мв} + P_{нк} + P_{пр} + P_{чр} \quad (4.10)$$

Сумма затрат = 248,21 + 49,64 + 103,07 + 7,86 + 0 + 60,2 + 0 + 0 + 248,21 = 717,19 руб.

Организация-разработчик участвует в освоении ПО и несёт соответствующие затраты, на которые составляется смета, оплачиваемая заказчиком по договору. Затраты на освоение ПО (P_o) определяются по формуле:

$$P_o = \frac{\text{Сумма затрат} \cdot H_o}{100\%}, \quad (4.11)$$

где H_o – норматив расходов на освоение, %.

В нашем случае нет затрат на освоение ПО.

Организация-разработчик осуществляет сопровождение ПО и несёт расходы, которые оплачиваются заказчиком в соответствии с договором и сметой на сопровождение. Затраты на сопровождение ПО (P_{co}) вычисляются по формуле:

$$P_{co} = \frac{\text{Сумма затрат} \cdot H_{co}}{100\%}, \quad (4.12)$$

где H_{co} – норматив расходов на сопровождение, %. Примем $H_{co} = 5\%$.

$P_{co} = 717,19 \cdot 5 / 100 = 35,86$ руб.

Полная себестоимость (C_{Π}) разработки программного продукта рассчитывается как сумма расходов по всем статьям. Определяется по формуле:

$$C_{\Pi} = \text{Сумма затрат} + P_o + P_{co} \quad (4.13)$$

$C_{\Pi} = 717,19 + 0 + 35,86 = 753,05$ руб.

Результаты вычислений внесём в таблицу 4.3.

Таблица 4.3 – Расчет себестоимости ПО

№ пп	Наименование статей затрат	Норматив	Сумма затрат, руб.
1	2	3	4
1	Зараб. плата, всего	-	297,88
2	В т.ч. основная	-	248,21
3	Дополнительная	20	49,64
4	Отчисления на социальные нужды	34,6	103,07
5	Спецоборудование	Не применялось	-

Продолжение таблицы 4.3

1	2	3	4
6	Материалы	3	7,86
7	Машинное время	0,7	60,02
8	Научные командировки	Не планировались	-
9	Прочие прямые затраты	-	-
10	Накладные расходы	100	248,21
11	Сумма затрат	-	717,19
12	Затраты на освоение	-	-
13	Затраты на сопровождение	5	35,86
14	Полная себестоимость	-	753,05

Таким образом, как видно из таблицы, полная себестоимость ПО равна 753,05 (руб.).

4.4 Расчет отпускной цены и прибыли программного обеспечения

Для определения отпускной цены ПО необходимо рассчитать плановую прибыль (Π), которая рассчитывается по формуле:

$$\Pi = \frac{C_{\Pi} \cdot R}{100}, \quad (4.14)$$

где C_{Π} – полная себестоимость ПО, руб.;

R – уровень рентабельности ПО, %. (Примем $R = 15\%$).

$\Pi = 717,19 \cdot 15 / 100 = 107,58$ руб.

После расчета прибыли от реализации определяется прогнозируемая цена ПО (Π) без налогов:

$$\Pi_{\Pi} = C_{\Pi} + \Pi \quad (4.15)$$

$\Pi_{\Pi} = 717,19 + 107,58 = 824,77$ руб.

Отпускная цена (цена реализации) ПО включает налог на добавленную стоимость и рассчитывается по формуле:

$$\Pi_o = \Pi_{\Pi} + \Pi_{\Pi} \cdot \text{НДС} \quad (4.16)$$

где C_{Π} – полная себестоимость программного продукта, руб.;

НДС - налог на добавленную стоимость, рассчитывается по формуле:

$$НДС = Ц_n \frac{НДС}{100} \quad (4.17)$$

где C_n — прогнозируемая цена, руб;

$НДС$ — налог на добавленную стоимость, в настоящее время составляет 20%.

$$НДС = 896,49 \cdot 20 / 100 = 179,30 \text{ руб.}$$

$$C_0 = 717,19 + 179,30 + 179,30 = 1075,79 \text{ руб.}$$

Прибыль от реализации ПО за вычетом налога на прибыль ($Пч$) является чистой прибылью, остается организации разработчику и представляет собой экономический эффект от создания нового программного продукта:

$$П_q = П \cdot \left(1 - \frac{Н_п}{100}\right), \quad (4.18)$$

где $Н_п$ — ставка налога на прибыль (в настоящее время $Н_п = 18\%$).

$$Пч = 179,30 \cdot (1 - 18 / 100) = 147,03 \text{ руб.}$$

Все расчеты цены и прибыли по ПО сведены в таблицу 4.4.

Таблица 4.4 – Отпускная цена ПО и чистая прибыль

№ п/п	Наименование	Норматив, %	Сумма, руб.
1	Полная себестоимость	-	753.05
2	Прибыль	15	179.30
3	Цена без НДС	-	896,49
4	НДС	20	179.30
5	Отпускная цена	-	1075,79
6	Чистая прибыль	18	147,03

Итак, определены основные экономические показатели:

- полная себестоимость – 753,05руб;
- отпускная цена – 1075,79руб;
- чистая прибыль – 147,03 руб.

Как видим из приведенных расчетов, данное ПО имеет выгоду и его разрабатывать целесообразно. Оно приносит разработчику чистую прибыль в размере 19,52%. Можно сделать вывод о быстрой окупаемости продукта и его конкурентоспособности за счет современного пользовательского интерфейса и использования новейших технологий, т.е. разработка данного продукта приведёт к положительному экономическому эффекту.

5 ЭНЕРГО– И РЕСУРСОСБЕРЕЖЕНИЕ

5.1 Ресурсосбережение

Ресурсосбережение - это совокупность мер по бережливому и эффективному использованию фактов производства. Обеспечивается посредством использования ресурсосберегающих и энергосберегающих технологий, снижения фондоемкости и материалоемкости продукции, повышения производительности труда, сокращения затрат живого и овеществленного труда, повышения качества продукции, рационального применения труда менеджеров и маркетологов, использования выгод международного разделения труда и др. Способствует росту эффективности экономики, повышению ее конкурентоспособности.

В Республике Беларусь действует межгосударственный стандарт разработанный Межгосударственным Техническим комитетом по стандартизации МТК 111. Настоящий стандарт является основополагающим и устанавливает цель, задачи, объекты, основные принципы, термины и классификацию групп требований рационального использования и экономного расходования материальных ресурсов на всех стадиях жизненного цикла веществ, материалов, изделий, продукции при проведении работ и оказании услуг юридическим и физическим лицам. Целью стандартизации в области ресурсосбережения является создание организационно-методической и нормативной основы, необходимой и достаточной для проведения государственной технической политики, направленной на снижение ресурсоемкости получаемого дохода без ухудшения условий экономического развития страны при безусловном обеспечении высоких потребительских свойств продукции.

Требования ресурсосбережения подразделяют на три группы:

- требования ресурсосодержания, определяющие совершенство процессов, продукции, работ и услуг, например, по составу и количеству использованных материалов, массе, габаритам, объему изделия;
- требования ресурсоемкости (по технологичности), определяющие возможность достижения оптимальных затрат ресурсов при изготовлении, ремонте и утилизации продукции, а также выполнении различных работ и оказании услуг с учетом требований экологической безопасности;
- требования ресурсоэкономичности изделия, определяющие возможность достижения оптимальных затрат ресурсов при эксплуатации, ремонте и утилизации продукции, а также при выполнении работ и оказании услуг.

Указанные группы требований взаимосвязаны при:

- разработке продукции, планировании работ и услуг (устанавливают проектные требования ресурсосодержания и ресурсоэкономичности, рекомендации по ресурсоемкости);
- изготовлении продукции, выполнении работ и оказании услуг (устанавливают уточненные (контрольные) требования ресурсоемкости (по технологичности));
- эксплуатации продукции, выполнении работ и оказании услуг (устанавливают уточненные (контрольные) требования ресурсоэкономичности и ресурсоемкости));
- утилизации продукции (устанавливают требования ресурсоемкости и ресурсоэкономичности).

5.2 Энергосбережение

Энергосбережение (экономия энергии) — реализация правовых, организационных, научных, производственных, технических и экономических мер, направленных на эффективное (рациональное) использование (и экономное расходование) топливно-энергетических ресурсов и на вовлечение в хозяйственный оборот возобновляемых источников энергии. Энергосбережение — важная задача по сохранению природных ресурсов.

В целях укрепления экономической безопасности государства 14 июня 2007 года Президентом Республики Беларусь подписана Директива № 3 «Экономия и бережливость - главные факторы экономической безопасности государства».

Государственное регулирование в области энергосбережения и повышения энергетической эффективности осуществляется путем установления:

- требований к обороту отдельных товаров, функциональное назначение которых предполагает использование энергетических ресурсов;
- запретов или ограничений производства и оборота в Республике Беларусь товаров, имеющих низкую энергетическую эффективность, при условии наличия в обороте или введения в оборот аналогичных по цели использования товаров, имеющих высокую энергетическую эффективность, в количестве, удовлетворяющем спрос потребителей;
- обязанности по учету используемых энергетических ресурсов;
- требований энергетической эффективности зданий, строений, сооружений;
- обязанности проведения обязательного энергетического обследования;
- требований к энергетическому паспорту;

- обязанности проведения мероприятий по энергосбережению и повышению энергетической эффективности в отношении общего имущества собственников помещений в многоквартирном доме;
- требований энергетической эффективности товаров, работ, услуг, размещение заказов на которые осуществляется для государственных или муниципальных нужд;
- требований к региональным, муниципальным программам в области энергосбережения и повышения энергетической эффективности;
- требований к программам в области энергосбережения и повышения энергетической эффективности организаций с участием государства или городского образования и организаций, осуществляющих регулируемые виды деятельности;
- основ функционирования государственной информационной системы в области энергосбережения и повышения энергетической эффективности;
- обязанности распространения информации в области энергосбережения и повышения энергетической эффективности;
- обязанности реализации информационных программ и образовательных программ в области энергосбережения и повышения энергетической эффективности.

ЗАКЛЮЧЕНИЕ

В данном дипломном проекте была создана программно-информационное обеспечение для поддержки процессов управления проектами.

В результате разработки системы, мы обеспечили следующий функционал приложения.

Основные функции системы:

- ведение баланса для каждого проекта, фиксирование их;
- организация и планирование проектов;
- контроль и регистрация финансов;
- управление менеджерами;
- добавление задач и составление отчетов по задачам;
- разделение пользователей по правам, доступа к функционалу системы;
- все данные хранятся в базе данных.

Система реализована на базе самых популярных и современных технологий.

Система обеспечивает санкционированный доступ к базе данных, интегрируется в веб-сервер apache, доступ к базе данных осуществляется с помощью СУБД phpmyadmin.

Приведены результаты испытаний системы, выполнен расчет экономического эффекта.

Приложение конкурентноспособно на рынке, а также имеет простой и удобный интерфейс пользователя, обеспечивает необходимый функционал, ускоряет работу и отзывчивость к пользователю, достоверность и полноту данных, повышает эффективность принимаемых решений.

СПИСОК СОКРАЩЕНИЙ

IDE - Integrated Development Environment – Интегрированная среда разработки

БД – база данных

ГОСТ – государственный стандарт

ДП – дипломное проектирование

ООП – объектно-ориентированное

ОС – операционная система

ПЗ – пояснительная записка

ПО – программное обеспечение

СУБД- система управлением базами данных

					ДП.АС36.120006 – 07 81 00	Лист
Изм	Лист	№ докум	Подп.	Дата		68

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 19.105 – 78 ЕСПД. Общие требования к программным документам.
2. СТ БПИ 01–98. Оформление материалов курсовых и дипломных проектов (работ), отчетов по практике. Общие требования и правила оформления. - Брестский политехнический институт, 1998.
3. Маклафлин, Г. PHP и MySQL. Исчерпывающее руководство / Г. Маклафлин – СПб.: Питер, 2014
4. Php.net – официальный сайт языка. Режим доступа: <https://php.net>, 31.04.2017
5. Stack Overflow – сайт вопросов и ответов для программистов. Режим доступа: <https://stackoverflow.com>, 5.05.2017
6. Habrahabr – русское сообщество программистов. Режим доступа: <https://habrahabr.ru/> 19.05.2017
7. Symfony – свободный фреймворк. Режим доступа: <http://symfony.com>, 3.03.2017
8. Openserver – локальный веб-сервер. Режим доступа: <https://ospanel.io>, 10.04.2017
9. Флэнаган, Д. JavaScript. Подробное руководство / Д. Флэнаган – СПб.: Символ, 2012

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
КАФЕДРА «ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ»

**РАЗРАБОТКА ПРОГРАММНО – ИНФОРМАЦИОННОГО ОБЕСПЕЧЕНИЯ
ДЛЯ ПОДДЕРЖКИ ПРОЦЕССОВ УПРАВЛЕНИЯ ПРОЕКТАМИ**

ТЕКСТ ПРОГРАММЫ
(на оптическом носителе CD-R)
ДП.АС36.120006 - 07 12 00

Листов 4

Объем 20,3мб

Руководитель

Г. Л. Муравьев

Выполнил

А.С. Бурштын

Консультанты:

по ЕСПД

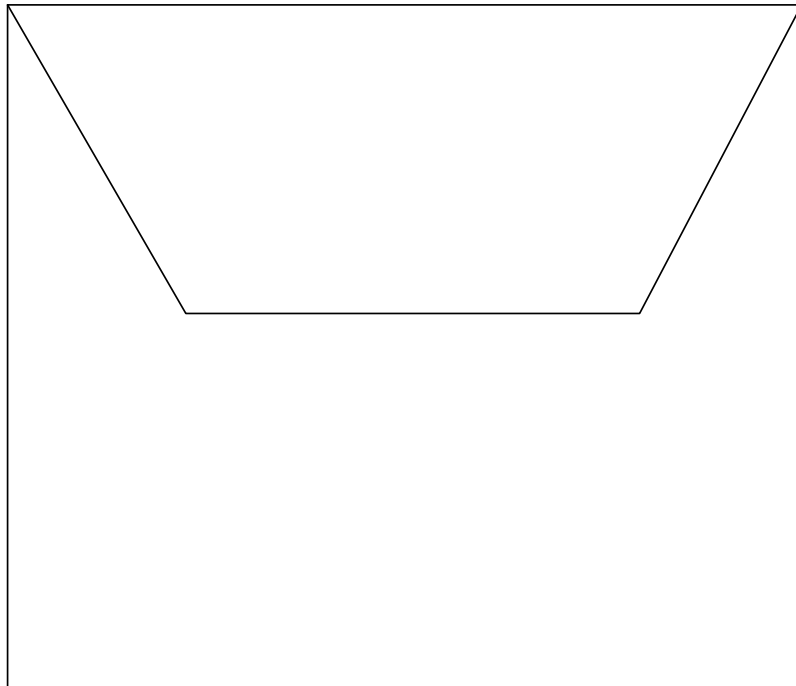
А.Л. Михняев

АННОТАЦИЯ

В дипломном проекте разработано программно-информационное обеспечение для поддержки процессов управления проектами. Система представляет собой web-приложение, которое работает на устройствах с наличием браузера и доступа в интернет. Приложение предназначено для управления проектами в небольших It-компаниях.

СОДЕРЖАНИЕ

1. App- содержит файлы фреймворка, а также параметры конфигурации сервера.
2. Bin – внутренний каталог фреймворка.
3. Vendor – каталог фреймворка.
4. Web - каталог фреймворка.
5. Entity – каталог, содержащий все пользовательские классы, используемые в приложении.
6. Src – основной каталог функционала приложения.
7. Crm.sql – дамп базы данных



УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
КАФЕДРА «ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ»

**РАЗРАБОТКА ПРОГРАММНО – ИНФОРМАЦИОННОГО ОБЕСПЕЧЕНИЯ
ДЛЯ ПОДДЕРЖКИ ПРОЦЕССОВ УПРАВЛЕНИЯ ПРОЕКТАМИ**

ИНСТРУКЦИЯ ПО УСТАНОВКЕ И ПРОВЕРКЕ СИСТЕМЫ

ДП.АС36.120006 - 07 33 00

Листов 6

Руководитель

Г. Л. Муравьев

Выполнил

А.С. Бурштын

Консультанты:

по ЕСПД

А.Л. Михняев

АННОТАЦИЯ

В дипломном проекте разработано программно-информационное обеспечение для поддержки процессов управления проектами. Система представляет собой web-приложение, которое работает на устройствах с наличием браузера и доступа в интернет. Приложение предназначено для управления проектами в небольших It-компаниях.

Документ «Инструкция по установке и проверке системы» состоит из двух разделов.

В разделе «Установка системы» описано каким образом устанавливается данная система на устройства.

В разделе «Проверка системы» описана проверка системы на работоспособность.

СОДЕРЖАНИЕ

1 УСТАНОВКА СИСТЕМЫ.....	4
2 ПРОВЕРКА СИСТЕМЫ.....	6

1 УСТАНОВКА СИСТЕМЫ

1. Загрузить содержимое архива на сервер.



crm

2. Создать базу данных mysql и импортировать sql дампы из исходников.

Сервер: 127.0.0.1:3306 » База данных: 1

Структура SQL Поиск Запрос по шаблону Экспорт Импорт Операции Привил

Импортируемый файл:

Файл может быть сжат в архив (gzip, bzip2, zip) или находиться без сжатия.
Имя сжатого файла должно заканчиваться в виде **[формат].[сжатие]**. Пример: **.sql.zip**

☒ Обзор вашего компьютера: D:\OpenServer\crm.sql [Обзор...](#) (Максимальный размер: 100МБ)
Вы также можете просто перетащить файл на любой странице.

☐ Выберите из каталога загрузки сервера d:/openserver/userdata/temp/: *Файлы для загрузки отсутствуют!*

Кодировка файла: utf-8

Частичный импорт:

☒ Разрешить скрипту разбивать процесс импорта при приближении временного лимита. *(Может быть использовано при этом вероятны проблемы с транзакциями.)*
Пропустить указанное число запросов (для SQL), начиная со следующего: 0

Прочие параметры:

☒ Включить проверку внешних ключей

Формат:

SQL

Параметры формата:

Режим совместимости SQL: NONE

☒ Не использовать атрибут AUTO_INCREMENT для нулевых значений

Консоль

3. Ввести в конфигурационный файл app/config/parameters.yml данные базы

```
# This file is auto-generated during the composer install
parameters:
    database_host: 127.0.0.1
    database_port: null
    database_name: crm
    database_user: test
    database_password: 1234
    mailer_transport: smtp
    mailer_host: 127.0.0.1
    mailer_user: null
    mailer_password: null
    secret: 7d8212cac2c8532cbf98490b638c22d608f5f5fd
```

4. В таблице «fos_user» вставить вручную запись главного администратора и в поле «roles» Вписать «a:1:{i:0;s:10:"ROLE_ADMIN"}»

roles (DC2Type=array)
a:0:{}
a:1: {i:0;s:10:"ROLE_ADMIN";}

5. Перейти из консоли сервера в каталог сайта и ввести «php app/console cache:clear --env=dev», для перезагрузки приложения. Теперь система готова к использованию.

2 ПРОВЕРКА СИСТЕМЫ

Проверка системы заключается в проверки корректности установки и связи с базой данных. И успешным показателем входа будет успешный логин в систему.

Перейдем на по адресу системы из браузера.

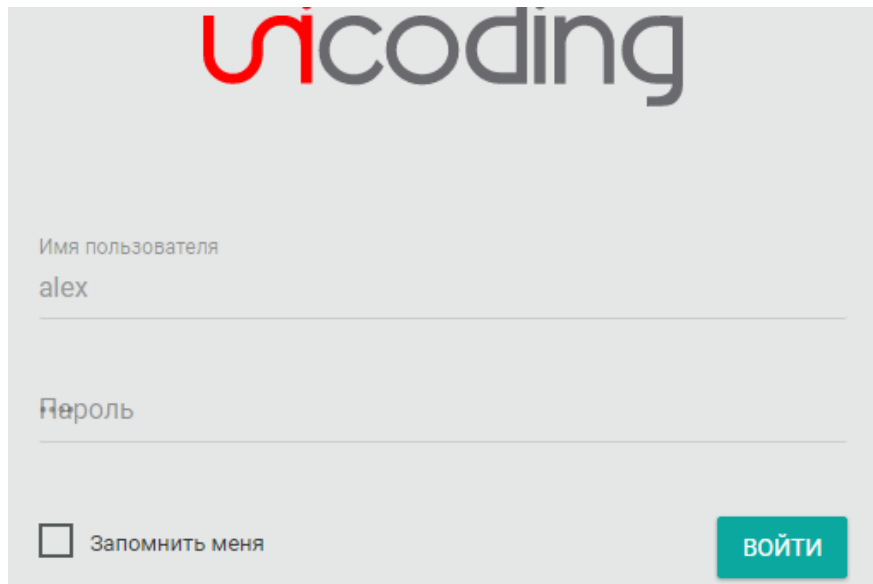
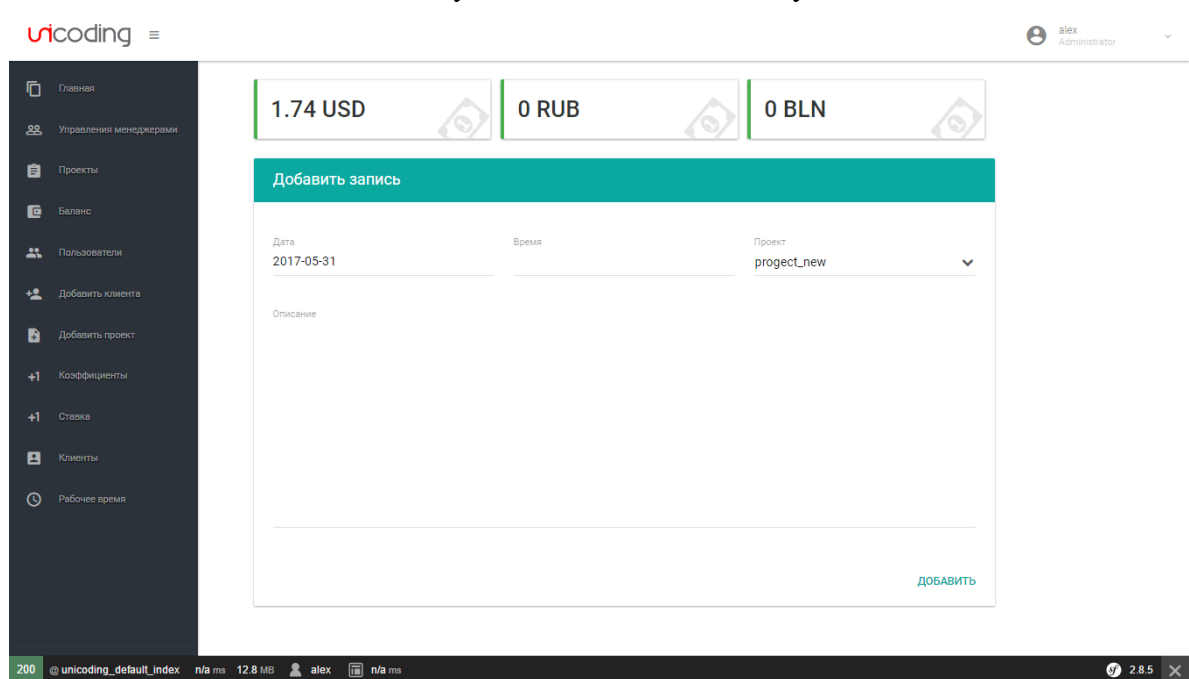


Рисунок 2.1 – Вход в систему



Рисунк 2.2. – Успешный вход

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
КАФЕДРА «ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ»

**РАЗРАБОТКА ПРОГРАММНО – ИНФОРМАЦИОННОГО ОБЕСПЕЧЕНИЯ
ВЕБ – ПРИЛОЖЕНИЯ «ТРАНСПОРТНОЕ СРЕДСТВО»**

ОПИСАНИЕ ПРИМЕНЕНИЯ СИСТЕМЫ

ДП.АС36.120006 - 07 31 00

Листов 6

Руководитель

Г. Л. Муравьев

Выполнил

А.С. Бурштын

Консультанты:

по ЕСПД

А.Л. Михняев

АННОТАЦИЯ

Данный документ представляет собой описание применения программно-информационного обеспечения для поддержки процессов управления проектами

Документ «Описание применения системы» состоит из трех разделов.

В разделе «Назначение системы» указано наименование, область применения системы.

В разделе «Условия применения» указаны условия для нормального функционирования данной системы.

В разделе «Входные и выходные данные» приведены сведения о входных данных, которые вводит пользователь, и выходных, которые получает пользователь при работе с системой.

СОДЕРЖАНИЕ

1 НАЗНАЧЕНИЕ СИСТЕМЫ	4
2 УСЛОВИЯ ПРИМЕНЕНИЯ	5
3 ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ	6

1 НАЗНАЧЕНИЕ СИСТЕМЫ

Разработанное программно-информационное обеспечение для поддержки процессов управления проектами предназначено для сокращения затрат на учет и обработку данных по управлению проектами, повышения оперативности доступа, достоверности и полноты данных, повышения эффективности принимаемых решений.

На сегодняшний день управление проектами неотъемлемая часть работы веб студий различного масштаба. Многие считают такие системы удобными, потому, что они позволяют экономить время на распределение ресурсов, контроль финансов и отслеживанием активности.

Система обеспечивает решение следующих задач:

- ведение базы данных;
- поддержка комплекса задач по учету и анализу данных о проектах и ресурсах;
- поддержка комплекса задач по организации и планированию проектов;
- поддержка комплекса задач по управлению проектами;
- поддержка комплекса задач по, обеспечивающих управление доступом;
- генерацию отчетов.

2 УСЛОВИЯ ПРИМЕНЕНИЯ

Для полноценного функционирования системы необходимо выполнение следующих требований:

- наличие устройств с доступом к серверу, где базируется приложение;
- на устройстве должен быть установлен современный браузер;

Требования к квалификации пользователя выставляет предприятие, работающее с системой.

4 ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ

Входными данными системы является следующая информация:

- данные о задаче;
- данные пользователя системы;
- информация о клиенте;
- информация о проекте;
- информация о коэффициентах;
- начальные параметры;

Выходными данными системы является следующая информация:

- таблицы менеджеров, закрепленных за проектами;
- список проектов, со стоимостью часа и валютой;
- баланс проекта, с валютой, суммой, датой комментарием и номером задачи;
- список пользователей системы;
- коэффициенты менеджеров;
- ставки менеджеров;
- список клиентов с полной информацией о них;
- рабочее время по месяцам и количеством часов.