

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование кафедры)

09.03.03 Прикладная информатика
(код и наименование направления подготовки / специальности)

Бизнес-информатика
(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Разработка информационной системы учёта учебного времени в образовательном учреждении»

Обучающийся

Н.А. Гребнев
(Инициалы Фамилия)

(личная подпись)

Руководитель

Н.Н. Казаченок

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2023

Аннотация

Тема бакалаврской работы – «Разработка информационной системы учёта учебного времени в образовательном учреждении».

Объект исследования – процесс учета учебного времени в образовательном учреждении с использованием комплекса программно-аппаратных средств.

Предметом исследования является автоматизация процесса учета учебного времени в образовательном учреждении.

Целью данной выпускной квалификационной работы является разработка информационной системы учета учебного времени в образовательном учреждении.

В работе рассмотрены проблемы, связанные с составлением учебного расписания в техникуме.

В первой главе произведен анализ деятельности предприятия и выявлена необходимость внедрения информационной системы в деятельность предприятия.

Во второй главе описывается процесс разработки логической модели информационной системы. Определяется выбор технологии моделирования и проектирования базы данных, которая используется в системе. Выполняется концептуальное и логическое моделирование.

Третья глава включает в себя физическое проектирование информационной системы и содержит подробное описание процесса разработки системы. В результате работы была создана функциональная информационная система, которая успешно прошла тестирование и может быть использована для решения задач в учебном учреждении.

В заключении описываются результаты выполнения выпускной квалификационной работы.

Объём работы - 60 страниц, на которых представлены 22 рисунка, 11 таблиц. При написании работы использовалось 22 источника.

Оглавление

Введение.....	5
Глава 1 Анализ учреждения и обоснование необходимости внедрения информационной системы учёта учебного времени	7
1.1 Техничко-экономическая характеристика предприятия и описание его деятельности.....	7
1.2 Концептуальное моделирование процессов техникума	11
1.2.1 Выбор технологии концептуального моделирования	11
1.2.2 Разработка и анализ модели «Как есть»	14
1.3 Обоснование необходимости реинжиниринга процессов.....	16
1.4 Разработка модели «Как должно быть»	17
1.5 Анализ существующих разработок.....	19
1.6 Постановка задачи на разработку информационной системы учета учебного времени.....	21
Глава 2 Логическое проектирование информационной системы учета учебного времени	24
2.1 Выбор технологии логического моделирования	24
2.2 Разработка логической модели информационной системы.....	25
2.3 Разработка спецификаций интерфейса.....	27
Глава 3 Физическое проектирование информационной системы.....	30
3.1 Выбор архитектуры информационной системы учета учебного времени	30
3.2 Выбор базы данных для информационной системы учета учебного времени	34
3.3 Разработка программного обеспечения информационной системы... ..	36
3.3.1 Выбор инструментов реализации	36
3.3.2 Реализация информационной системы.....	41
3.4 Тестирование разработанной системы.....	49

3.5 Описание функциональности информационной системы учета учебного времени.....	50
3.6 Расчет экономической эффективности внедрения системы	51
Заключение	57
Список используемой литературы и используемых источников.....	59

Введение

В современном мире развитие информационных технологий оказывает существенное влияние на различные сферы деятельности, в том числе и на образование. Одним из важных аспектов управления образовательным процессом является составление расписания занятий, а также учет учебного времени, что позволяет эффективно планировать учебный процесс и контролировать выполнение учебного плана. В связи с этим возникает необходимость внедрения информационной системы, которая позволит автоматизировать процесс учета учебного времени в образовательном учреждении.

Объектом исследования в данной работе является процесс учета учебного времени в образовательном учреждении с использованием комплекса программно-аппаратных средств.

Предметом исследования является автоматизация процесса учета учебного времени в образовательном учреждении.

Целью данной выпускной квалификационной работы является разработка информационной системы учета учебного времени в образовательном учреждении.

Для достижения поставленной цели были поставлены следующие задачи:

- определение требований к информационной системе;
- анализ существующих методов учета учебного времени;
- выбор инструментов, методологий, а также технологий программной реализации системы;
- разработка концептуальной модели системы;
- проектирование и разработка структуры информационной системы;
- реализация и тестирование системы, а также оценка ее эффективности

Данная работа состоит из трех разделов:

- аналитический раздел, который включает в себя анализ деятельности образовательного учреждения и определение требований к разрабатываемой информационной системе учета учебного времени;
- раздел логического проектирования, где произведен выбор технологии логического моделирования информационной системы, а также определены принципы разработки интерфейса информационной системы;
- раздел физического проектирования, который подробно описывает процесс разработки системы, её тестирования, а также обосновывает экономическую эффективность проекта.

Результатом данной работы является разработанная информационная система, которая позволит облегчить учет учебного времени в образовательном учреждении, повысить эффективность планирования учебного процесса и контроля за его выполнением. В дальнейшем данная система может быть доработана и использована в других образовательных учреждениях.

Глава 1 Анализ учреждения и обоснование необходимости внедрения информационной системы учёта учебного времени

1.1 Техничко-экономическая характеристика предприятия и описание его деятельности

В работе рассмотрена деятельность государственного автономного профессионального образовательного учреждения Свердловской области «Техникум индустрии питания и услуг «Кулинар» – ГАПОУ СО «ТИПУ «Кулинар».

Техникум «Кулинар» – это учебное заведение, расположенное в городе Екатеринбург, которое специализируется на подготовке студентов в области кулинарного и кондитерского искусства, а также обслуживания в общественном питании.

Основное направление деятельности: предоставление образовательных услуг.

На сегодняшний день техникум осуществляет образовательную деятельность в 3 филиалах.

В техникуме «Кулинар» реализуются различные программы обучения, включающие в себя курсы по направлениям:

- повар-кондитер;
- пекарь;
- поварское и кондитерское дело;
- организация обслуживания в общественном питании;
- технология продукции общественного питания.

Система обучения построена таким образом, чтобы студенты могли получить необходимые знания и практические навыки для работы в профессиональной кулинарии и обслуживании клиентов в общественном питании.

Техникум «Кулинар» имеет современное оборудование и классы, обеспеченные необходимой техникой для обучения поваров и кондитеров. Преподаватели и мастера производственного обучения являются опытными специалистами в области кулинарии и обслуживания в общественном питании, которые могут поделиться своим опытом и знаниями со студентами.

Студенты техникума «Кулинар» имеют возможность получить практический опыт работы в ресторанах и кафе, что помогает им научиться работать в реальных условиях и готовить разнообразные блюда, отвечающие требованиям клиентов.

Техникум «Кулинар» имеет высокую репутацию и является одним из лучших учебных заведений в Екатеринбурге для подготовки профессиональных поваров, кондитеров и официантов.

Целью техникума «Кулинар» является подготовка высококвалифицированных, творчески мыслящих специалистов в области кулинарии и обслуживания в общественном питании, на основе оптимального сочетания фундаментального образования и практико-ориентированной профильной подготовки с учетом требований времени.

Задачи техникума «Кулинар» включают в себя:

- расширение перечня направлений подготовки специалистов среднего звена с учетом интересов личности, потребностей рынка труда, перспектив развития экономики и социальной сферы;
- повышение качества подготовки и обеспечение уровня профессиональной компетенции, конкурентоспособности, профессиональной мобильности специалистов в соответствии с целевой установкой, требованиями рынка труда и образовательных стандартов;
- формирование качеств личности студента на основе принципов гуманизации, интеллектуализации, информатизации среднего профессионального образования, а также усиления общенаучной и общепрофессиональной подготовки;

- обеспечение качественного роста педагогических кадров и укрепление материально-технической базы, как основы подготовки высококвалифицированных кадров;
- совершенствование организации образовательного процесса;
- информатизация содержания образования;
- обеспечение преемственности содержания среднего профессионального образования с другими уровнями образования; продолжение интеграции с вузом.

На рисунке 1 изображена организационная структура техникума «Кулинар» и распределение ответственности между сотрудниками.

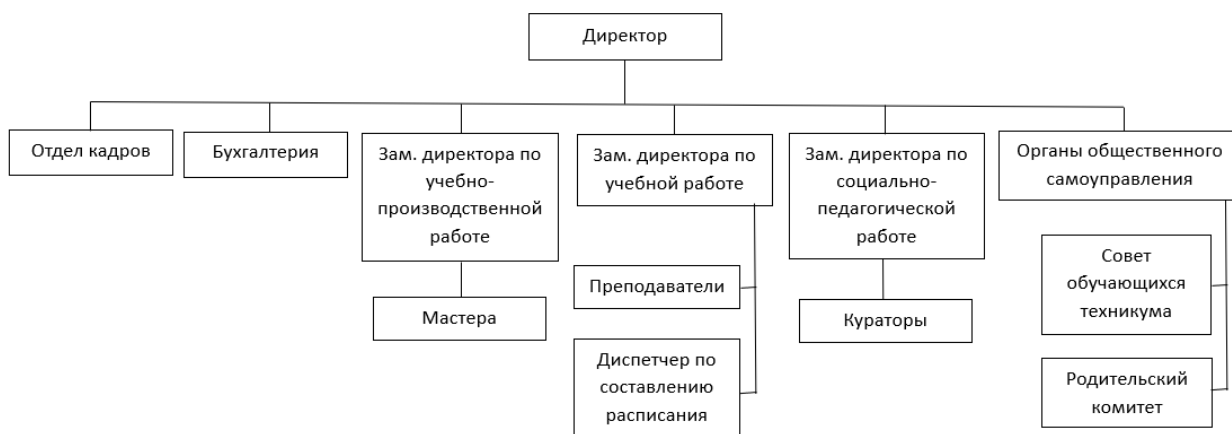


Рисунок 1 – Организационная структура предприятия

Организационная структура предприятия состоит из:

- директор техникума «Кулинар» - отвечает за общее управление и координацию работы техникума. Он принимает решения по основным вопросам управления, контролирует исполнение решений и отчитывается перед учредителем о результатах работы техникума;
- заместитель директора по учебной работе отвечает за организацию учебного процесса, разработку учебных планов, контроль за

- качеством образовательных услуг, сотрудничество с работодателями и другими учебными заведениями;
- заместитель директора по социально-педагогической работе отвечает за организацию воспитательной работы среди студентов, проведение культурно-массовых мероприятий, поддержание дисциплины в техникуме и взаимодействие с кураторами и родителями;
 - заместитель директора по учебно-производственной работе – отвечает за организацию процесса обучения, в том числе планирование и координацию учебных программ, контроль за качеством образовательной деятельности, поддержание связей с работодателями и организацией практики студентов, разработку и внедрение новых технологий обучения и оценки результатов обучения;
 - преподаватели отвечают за проведение учебных занятий, контроль знаний студентов, разработку методических материалов и привлечение к учебному процессу опыта из своей профессиональной деятельности;
 - мастера производственного обучения - проводят практические и лабораторные занятия, тренинги, участвуют в организации практик и производственных стажировок студентов;
 - технический персонал отвечает за обеспечение работы технических средств, содержание помещений, приготовление питания в столовой техникума, организацию обслуживания студентов, уборку помещений, поддержание порядка на территории техникума и другие технические задачи;
 - диспетчер по составлению расписания в техникуме отвечает за планирование и организацию расписания занятий и других событий в учебном заведении. В его обязанности входит составление расписания занятий, координация преподавателей и студентов, а

также корректировка расписания и обновление информации при каких-либо изменениях.

Таким образом, каждый сотрудник техникума «Кулинар» имеет свои обязанности, ответственность за выполнение которых несет он сам и непосредственный руководитель соответствующего отдела. Работа всех специалистов направлена на достижение основной цели предприятия - обеспечение качественного и эффективного образовательного процесса, а также на инновационное развитие учебного заведения.

1.2 Концептуальное моделирование процессов техникума

1.2.1 Выбор технологии концептуального моделирования

В методологии моделирования выделяют различные подходы к построению и отображению моделей бизнес-процессов или виды моделирования, основными среди которых считаются структурное, объектно-ориентированное и интегрированное [11].

Структурное моделирование – это область системного анализа и вид моделирования, который используется как средство исследования систем и может служить для их разработки. Оно изучает структуру системы с точки зрения состава элементов и подсистем и отношений между ними (структура), а также с точки зрения свойств системы, которые позволяют достигать заданной цели (функции). Основным видом структурного моделирования является функциональное моделирование.

Функциональное моделирование – это вид моделирования, который подразумевает описание процессов в виде взаимосвязанных, четко структурированных функций. Таким образом, главный элемент – это функция (операция), а бизнес-процесс представляется в виде последовательности функций, преобразующих входы процесса в выходы с использованием определенных ресурсов. Функциональное моделирование в свою очередь также делится на несколько видов, которые представлены в таблице 1.

Таблица 1 – Виды функционального моделирования

Вид моделирования	Метод	Нотация
Функциональное моделирование	SADT	IDEF0
	DFD	Гайна - Сарсона Gane / Sarson Йордана - Де Марко Yourdon / DeMarko
	WFD	IDEF3 (PFDD)
	ARIS	eEPC
		BPMN

Объектно-ориентированное моделирование подразумевает описание процессов, как набора взаимодействующих объектов без детализации выполняемых операций, но с описанием условий и событий. Объект – это какой-либо предмет, который преобразуется при выполнении процессов. В основе – объектная модель, которая базируется на таких принципах, как инкапсуляция, абстрагирование, полиморфизм, наследование, параллелизм, устойчивость и т.д. При этом статическую структуру модели описывают объекты, а поведение модели – сообщения, которыми эти объекты обмениваются [2, с. 16]. Виды объектно-ориентированного моделирования представлены в таблице 2.

Таблица 2 – Виды объектно-ориентированного моделирования

Вид моделирования	Метод	Язык графического описания
Объектно-ориентированное моделирование	метод Гради Буча	UML
	метод Джеймса Румбаха (OMT)	
	метод Айвара Джекобсона (OOSE)	

Интегрированное моделирование объединяет различные виды моделей – структурного анализа, объектно-ориентированные, имитационные и другие.

Таким образом, это совокупность нескольких различных моделей, каждая из которых описывает отдельные перспективы его структуры, а все

вместе они образуют полное и комплексное представление о моделируемом объекте. Виды интегрированного моделирования представлены в таблице 3.

Таблица 3 – Виды интегрированного моделирования

Вид моделирования	Методология	Метод
Интегрированное моделирование	ARIS	eEPC

Для построения модели «Как есть» в данной работе было принято решение использовать технологию функционального моделирования. В таблице 4 представлено сравнение нотаций функционального моделирования.

Таблица 4 – Сравнение видов функционального моделирования

Нотация	Преимущества	Недостатки
IDEF0	позволяет описать систему как совокупность функций и данных, удобна для анализа процессов и моделирования бизнес-процессов	слишком сложна для простых систем, не подходит для моделирования деталей системы, ограниченная поддержка инструментов
DFD	проста в использовании, удобна для описания потоков данных в системе	неудобна для моделирования бизнес-процессов, не подходит для описания сложных логических операций
IDEF3	хорошо подходит для моделирования бизнес-процессов, удобна для описания последовательности действий в системе	не подходит для моделирования систем в целом, неудобна для описания сложных логических операций, ограниченная поддержка инструментов
eEPC	широко используется в проектировании информационных систем, удобна для описания архитектуры системы	слишком сложна для моделирования простых систем, ограниченная поддержка инструментов
BPMN	широко используется для моделирования бизнес-процессов, удобна для описания действий и потоков работ	не подходит для моделирования систем в целом, ограниченная поддержка инструментов, неудобна для моделирования деталей системы

Рассмотрев преимущества и недостатки каждой из нотаций, было определено, что для концептуального моделирования процессов предприятия наиболее подходящим языком моделирования является IDEF0.

1.2.2 Разработка и анализ модели «Как есть»

На рисунке 2 представлена контекстная диаграмма «Как есть» бизнес-процесса «Обучение студентов». На входе информационные потоки включают прием абитуриентов, лекционные и методические материалы. На выходе информационные потоки включают:

- дипломы о средне-специальном образовании;
- рейтинг студентов;
- приказ об отчислении студентов;
- приказ о переводе студентов на следующий курс.

Регулируется деятельность техникума законодательством РФ, учебным планом, уставом техникума, государственным заданием и нормами СанПиН.

На рисунке 3 представлена декомпозиция «Как есть» бизнес-процесса «Обучение студентов».

Так же на рисунке 4 представлена диаграмма декомпозиции «Как есть» бизнес-процесса «Составление расписания».

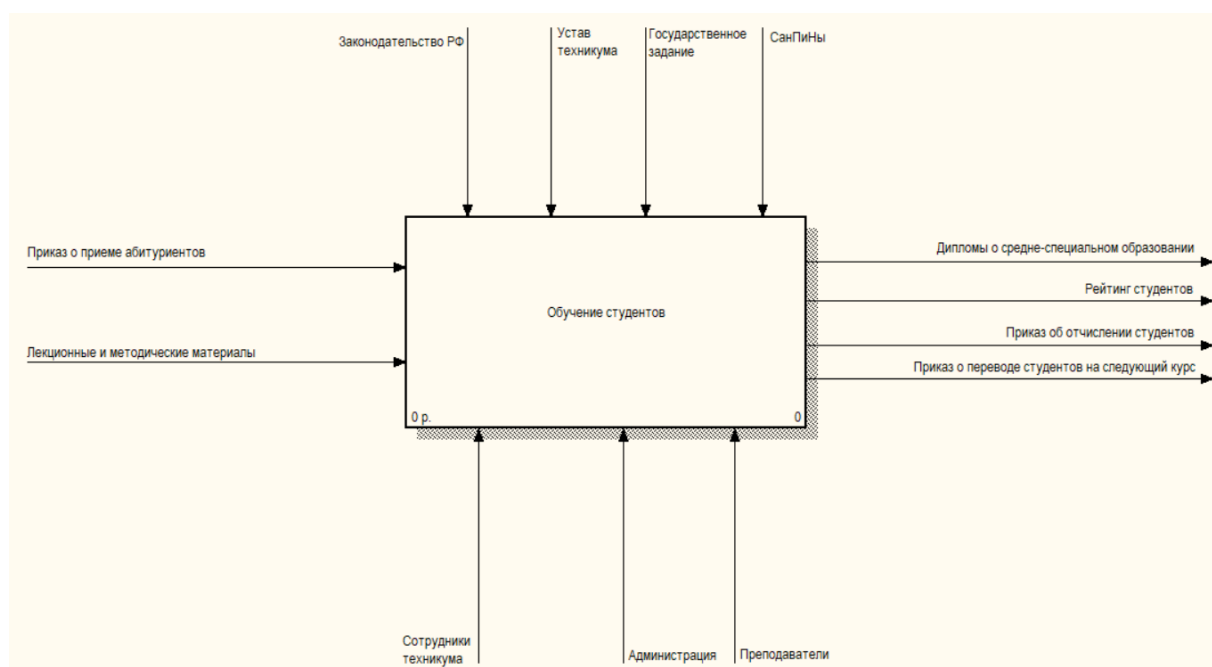


Рисунок 2 – IDEF0 контекстная диаграмма «Как есть» для основного бизнес-процесса «Обучение студентов»

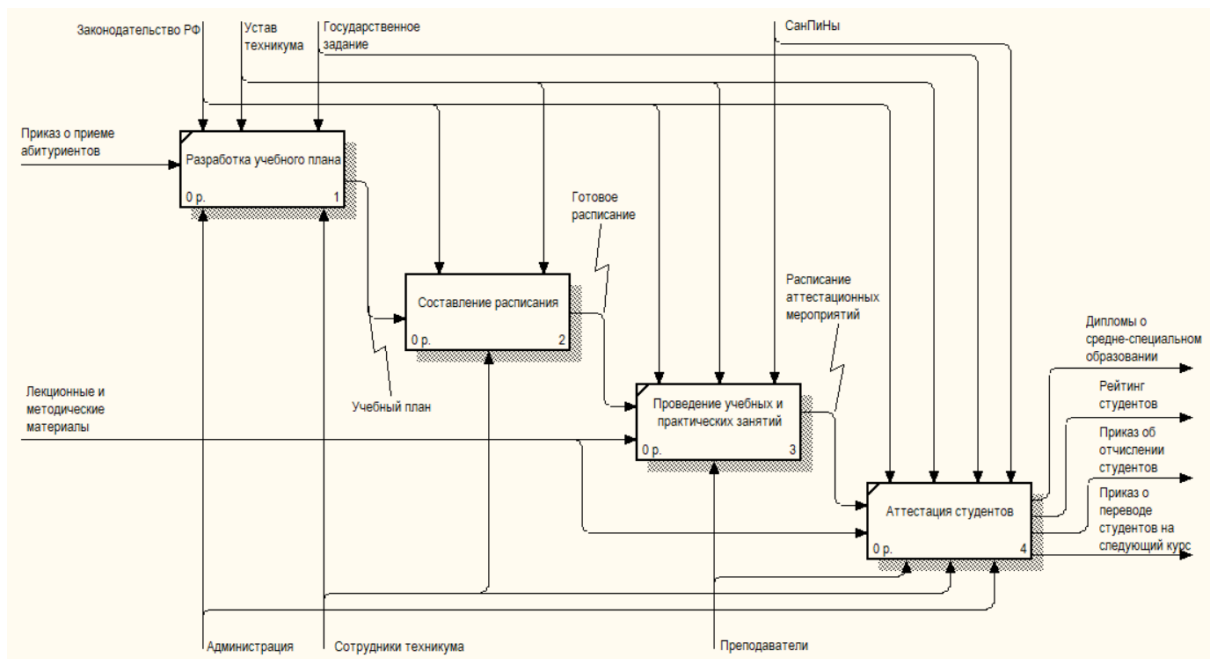


Рисунок 3 – IDEF0 диаграмма «Как есть» декомпозиции бизнес-процесса «Обучение студентов»

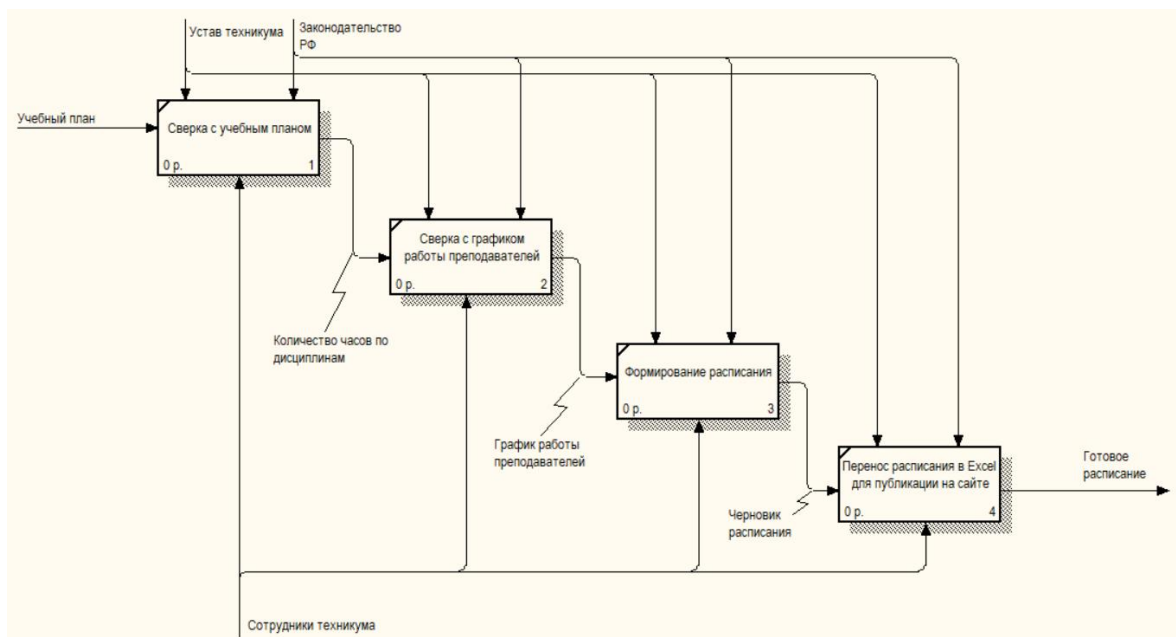


Рисунок 4 – IDEF0 диаграмма «Как есть» декомпозиции бизнес-процесса «Составление расписания»

Анализ процессов показал, что в настоящее время техникум испытывает сложности с составлением расписания. Расписание составляется на бумаге без использования компьютера, отсутствует база данных, к

которой можно обратиться для просмотра проведенных часов по предметам для составления нового расписания.

Основная проблема – это отсутствие информационной системы, в которой бы хранилась вся информация о количестве учебных часов, которые необходимо провести, информация о проведенных занятиях и информация о занятости преподавателей и аудиторий.

1.3 Обоснование необходимости реинжиниринга процессов

Составление расписания в образовательном учреждении – это сложный процесс, который требует учета многих факторов, таких как занятость преподавателей, доступность аудиторий, расписание занятий, предпочтения учеников и учителей и многое другое. Ниже представлены некоторые лучшие практики, которые помогут оптимизировать процесс составления расписания в образовательном учреждении:

- автоматизация процесса составления расписания посредством программного обеспечения может значительно упростить процесс, сократить время работы и уменьшить возможность ошибок;
- учет предпочтений учителей и учеников поможет улучшить качество обучения и увеличить мотивацию участников образовательного процесса;
- использование аналитических инструментов для определения оптимального расписания позволит определить оптимальное расписание, учитывая ограничения, такие как: доступность аудиторий, количество учителей и учеников и так далее;
- регулярное обновление расписания позволит быстро реагировать на изменения в расписании, такие как: изменения количества учеников, изменения в доступности аудиторий;

- прозрачность расписания для всех участников образовательного процесса, чтобы они могли легко получить информацию о расписании и не пропустить занятия.

Выбор оптимального решения по оптимизации/реинжинирингу процесса составления расписания в образовательном учреждении зависит от конкретных потребностей и ограничений учебного заведения. Например, если учебное заведение имеет большое количество учеников и учителей, то использование программного обеспечения для автоматизации процесса составления расписания может быть оптимальным решением. Если же учебное заведение имеет ограниченное количество аудиторий, то использование аналитических инструментов для определения оптимального расписания.

Внедрение информационной системы решит следующие проблемы:

- ИС будет производить учёт количества проведенных часов по предметам, а также остаток часов, которые необходимо провести в соответствии с учебным планом;
- ИС будет проводить проверку на возможные ошибки при составлении расписания, такие как занятость преподавателя или аудитории.

Таким образом, в целях оптимизации процесса составления расписания необходимо создать информационную систему, которая упростит работу сотрудников техникума и уменьшит вероятность ошибок в процессе распределения и учета учебного времени.

1.4 Разработка модели «Как должно быть»

На рисунках 5, 6 и 7 изображена функциональная модель «Как должно быть», красным цветом выделены изменения модели «Как есть».

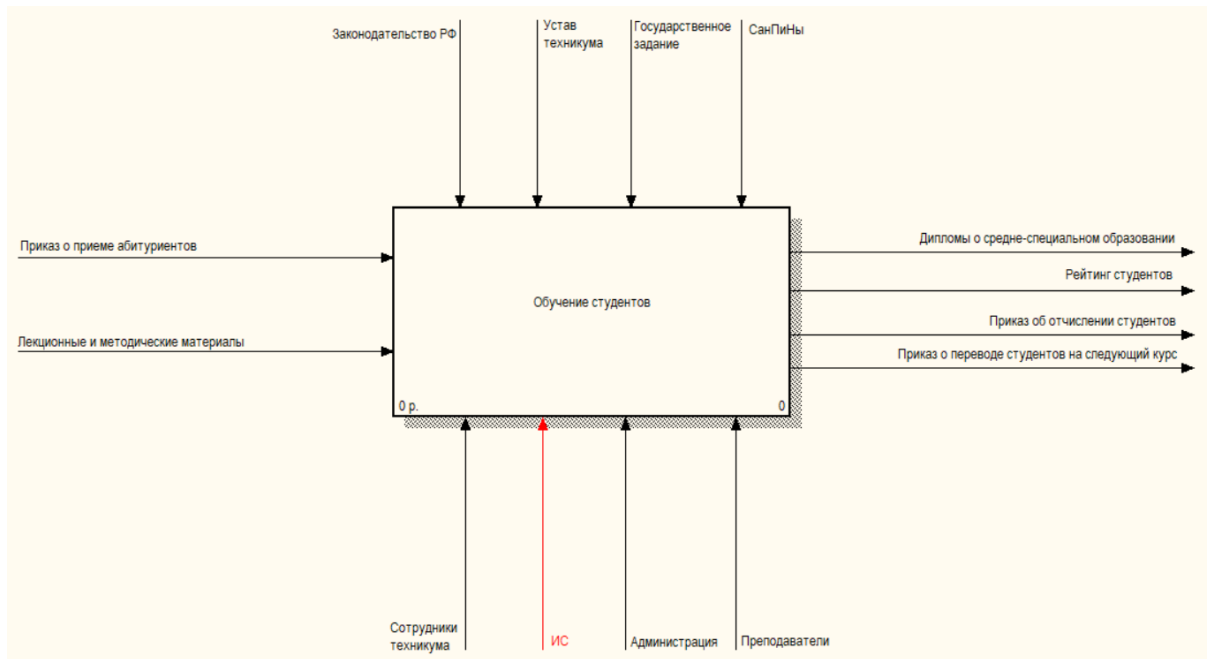


Рисунок 5 – IDEF0 контекстная диаграмма «Как должно быть» для основного бизнес-процесса «Обучение студентов»

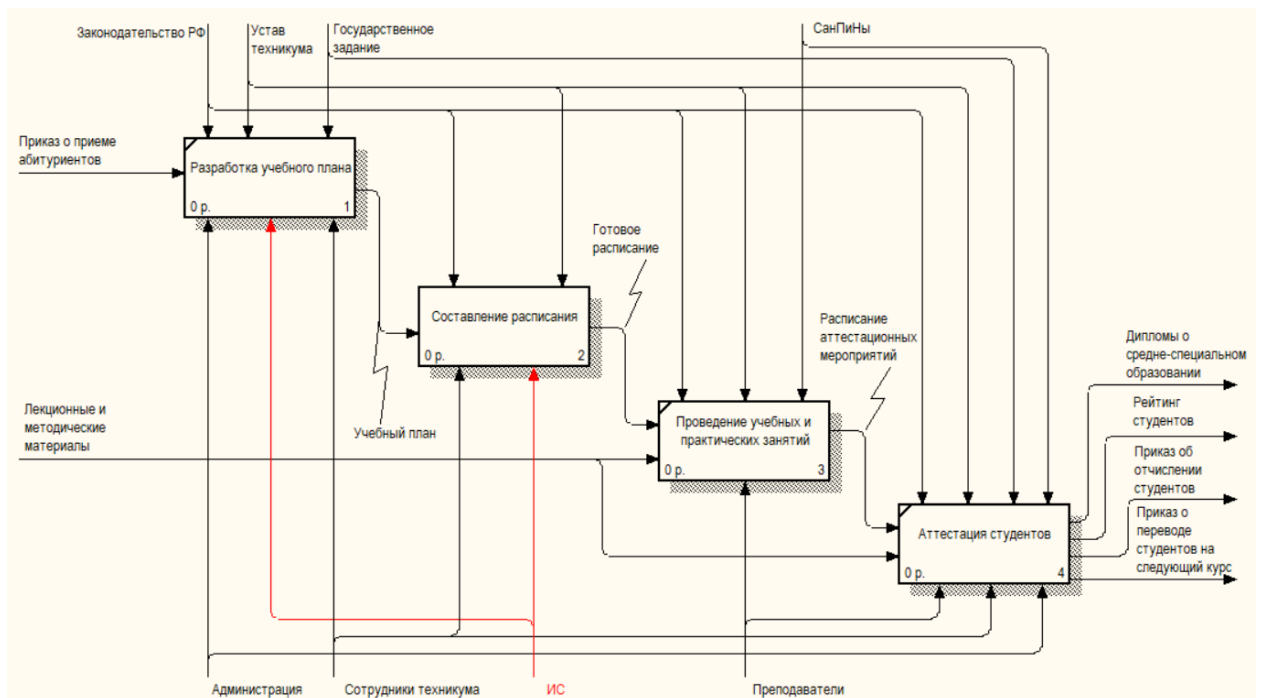


Рисунок 6 – IDEF0 диаграмма «Как должно быть» декомпозиции бизнес-процесса «Обучение студентов»

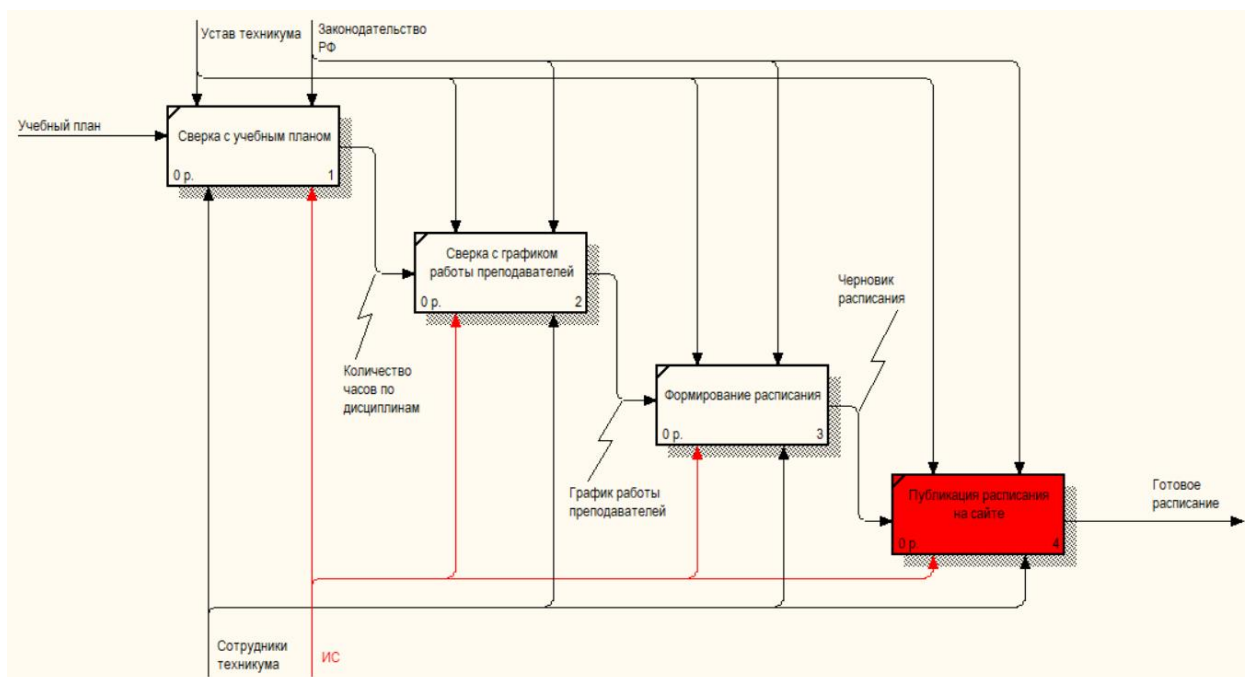


Рисунок 7 – IDEF0 диаграмма «Как должно быть» декомпозиции бизнес-процесса «Составление расписания»

Внедрение ИС в бизнес-процесс «Составление расписания» упростит и ускорит процесс составления расписания и избавит от необходимости ручного переноса и учета количества учебных часов. Такие изменения позволят перераспределить нагрузку сотрудников для решения других задач, что в конечном счете повысит эффективность предприятия в целом.

1.5 Анализ существующих разработок

Рассмотрим имеющиеся разработки, которые могут помочь сотрудникам предприятия в составлении расписания. В качестве критериев анализа аналогов разрабатываемой системы выберем стоимость и следующие функциональные возможности:

- автоматическое составление расписания;
- ручное составление расписания;
- внесение изменений в расписание;
- публикация расписания на веб сайте;

- составление отчетов по количеству проведенных часов;
- составление расписания для нескольких учебных корпусов;
- использование точных дат, а не условных дней.

Таблица 5 – Сравнительная характеристика программ учета учебного времени

Программный продукт	Стоимость	Автоматическое составление расписания	Ручное составление расписания	Внесение изменений в расписание	Учет учебных часов	Публикация расписания	Составление расписания для нескольких площадок	Точные даты при составлении расписания
БИТ.Расписание для колледжей	50 000 р	+	-	+	+	+	+	-
1С: А.с.р.	29 000р	+	-	+	+	-	+	-
Галактика РУЗ	267 000 р	+	+	+	+	+	+	+
Составитель расписания	300р	+	-	-	+	-	-	-
aSc TimeTables	€ 149	+	-	+	+	+	+	+
Экспресс-расписание Колледж	7990р	+	-	+	+	+	-	+
«АВТОРасписание»	48 000р	+	-	+	+	+	+	-
«Ректор-Колледж»	14 700 р	+	-	+	+	-	-	+
«НИКА-Колледж»	12 700 р	+	+	+	+	+	-	-
ФЕТ	бесплатная	+	-	-	+	+	+	-
Свободное Расписание	бесплатная	+	+	+	+	-	-	-
TMIS - Система Управления Расписанием	бесплатная	+	-	-	-	-	-	+

Были рассмотрены следующие программные продукты:

- «БИТ.Расписание для колледжей»;

- «1С: Автоматизированное составление расписания»;
- «Галактика РУЗ (расписания учебных занятий)»;
- «Составитель расписания»;
- «FET»;
- «aSc TimeTables»;
- «Экспресс-расписание Колледж»;
- «АВТОРасписание»;
- «Ректор-Колледж»;
- «НИКА-Колледж»;
- «Свободное Расписание»;
- «ТМIS - Система Управления Расписанием».

Обзор различных ИС показал, что основными недостатками рассмотренных программных продуктов являются высокая стоимость программ и отсутствие необходимого функционала.

Исходя из этого было принято решение сформулировать функциональные требования для системы учета учебного времени, а затем на основе этих данных реализовать программу, которая удовлетворяла бы всем требованиям предприятия.

1.6 Постановка задачи на разработку информационной системы учета учебного времени

В рамках ВКР необходимо реализовать ИС, которая помогала бы в составлении расписания учебных занятий и публикации расписания на веб сайте. Система должна иметь отдельный доступ и различный функционал для студентов и администратора расписания:

- для студентов – просмотр текущего расписания по группам;

- для администратора расписания – составление и изменение расписания, составление отчетов по количеству часов проведенных предметов.

Функциональные требования к системе для студентов:

- просмотр текущего расписания с фильтрацией по курсу и группе;
- просмотр расписания за выбранный период времени с фильтрацией по курсу и группе.

Функциональные требования системы для администратора расписания:

- составление расписания;
- изменение расписания;
- просмотр расписания с фильтрацией по группам и преподавателям;
- проверка на пересечения занятий готового расписания;
- публикация расписания на веб сайте;
- формирование отчетов по количеству проведенных часов по предметам.

Требования к информационной системе:

- интуитивно понятный интерфейс, который позволит пользователям быстро и легко составлять расписание;
- система должна быть доступна из любого места и на любом устройстве с доступом в интернет, чтобы пользователи могли редактировать и просматривать расписание независимо от локации;
- возможность генерировать различные отчеты и статистику, такие как сводные таблицы расписания, списки занятий, нагрузка на преподавателей и использование аудиторий;
- возможность внесения изменений в расписание в соответствии с изменениями в учебных планах, составе преподавателей, группах студентов или доступности аудиторий.

Бизнес-цели:

- сокращение времени на разработку и публикацию расписания;
- снижение вероятности ошибок при составлении расписания и уменьшение времени на их исправление;
- улучшение коммуникации между преподавателями, студентами и администрацией в отношении расписания;
- увеличение удобства использования для студентов и преподавателей.

Также одним из требований, предъявляемых к разрабатываемой системе, является создание и настройка удобного пользовательского интерфейса, обеспечивающего легкое восприятие и обработку информации, а также минимизирующего пользовательские операции.

Выводы по главе 1

В первой главе был проведен анализ деятельности предприятия, а также определена технологии концептуального моделирования в пользу IDEF0.

На основе полученной информации была создана функциональная модель «Как есть» и «Как должно быть», которая наглядно показала необходимость внедрения ИС в деятельность предприятия.

Помимо этого, был произведен анализ уже существующих систем учета учебного времени и выявлены главные недостатки существующих ИС. Высокая стоимость и ограниченный функционал определил необходимость разработки собственного программного обеспечения.

Анализ существующих программ помог определить основные требования к готовой системе, среди которых: удобный, понятный интерфейс и возможность доступа к системе с любых устройств.

Следующим шагом необходимым для достижения цели данной работы является логическое проектирование программного обеспечения.

Глава 2 Логическое проектирование информационной системы учета учебного времени

2.1 Выбор технологии логического моделирования

Для лучшего понимания разрабатываемой информационной системы необходимо разработать логическую модель, на которой будут отображены все происходящие процессы.

В предыдущей главе были рассмотрены основные методы моделирования. Для описания логической модели лучше всего подойдет объектно-ориентированное моделирование в нотации UML.

Унифицированный язык моделирования (UML) играет важную роль в разработке программного обеспечения, а также в системах, не связанных с ИТ, поскольку он дает возможность визуально показать поведение и структуру системы или процесса. UML помогает продемонстрировать возможные ошибки в структурах приложений, поведении системы и других бизнес-процессах [14].

Чтобы показать, какая задача решается системой в процессе ее работы, используется диаграмма вариантов использования. Она не предназначена для отображения деталей проекта и не позволяет описать внутреннее устройство системы. Диаграммы вариантов использования включают в себя элементы, такие как актеры и варианты использования.

Используя диаграмму вариантов использования, можно подробно описать поведение информационной системы. Прецеденты позволяют пользователям и разработчикам общаться на общем языке. Каждый элемент на диаграмме прецедентов может содержать множество операций и составных частей. Описание прецедентов помогает конечным пользователям понимать, как взаимодействовать с элементами системы. Представление поведения элемента с высокой степенью детализации обеспечивается благодаря использованию диаграммы вариантов использования [6, с. 45].

Для описания логической модели данных будет использована методология IDEF1X. IDEF1X – это язык моделирования данных, используемый для проектирования баз данных и описания бизнес-процессов [21, с. 21]. Он был разработан в рамках программы и используется в качестве стандарта моделирования.

В качестве инструмента для реализации диаграммы вариантов использования была выбрана программа MS Visio, логическая модель разработана в программе Toad Data Modeller.

2.2 Разработка логической модели информационной системы

Чтобы определить взаимодействие между разрабатываемой информационной системой и пользователями необходимо разработать диаграмму вариантов использования, которая представлена на рисунке 8.



Рисунок 8 – Диаграмма вариантов использования

На диаграмме представлены следующие роли:

Администратор расписания – сотрудник, ответственный за составление расписания, внесение изменений, учет учебного времени и публикацию расписания на сайте.

Студент – пользователь, который может просматривать расписание занятий.

На рисунке 9 представлена диаграмма классов разрабатываемой системы.

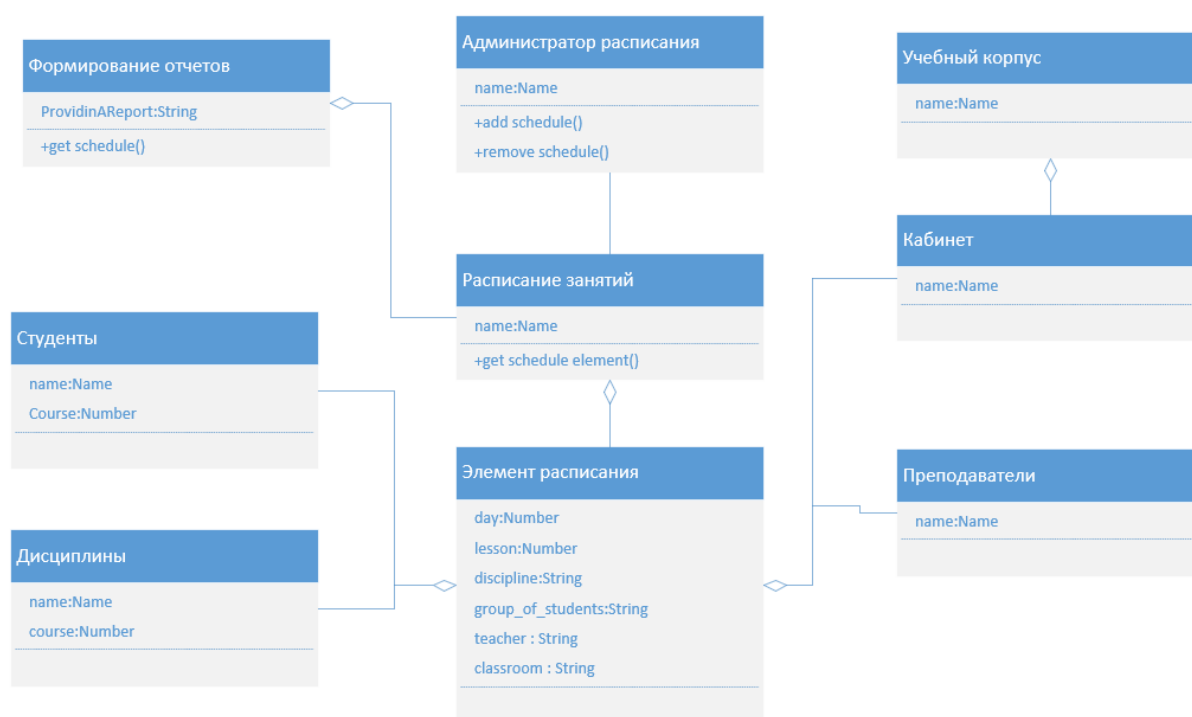


Рисунок 9 – Диаграмма классов

На рисунке 10 изображена логическая модель данных, представляющая основные таблицы и структуру базы данных.

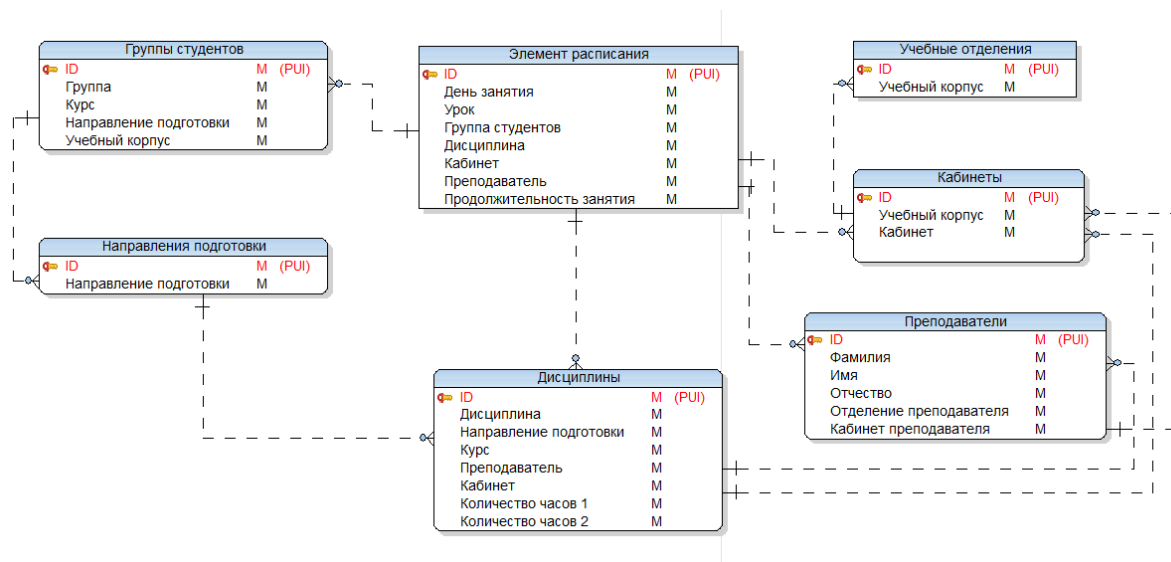


Рисунок 10 – Логическая модель данных

Разработанная логическая модель данных включает в себя 7 таблиц: группы студентов, направления подготовки, учебные занятия, дисциплины, отделения, аудитории, преподаватели.

2.3 Разработка спецификаций интерфейса

Разработка спецификаций интерфейса будет производиться с использованием стандарта ГОСТ Р ИСО 9241-210-2012 «Эргономика взаимодействия человек-система. Часть 210. Человеко-ориентированное проектирование интерактивных систем» [5].

Изучив документ, были определены основные принципы разработки пользовательского интерфейса:

- проектирование на основе точного определения пользователей, задач и среды;
- вовлечение пользователей в разработку;
- улучшение проекта за счет его оценки пользователями;
- итеративное совершенствование проекта;
- учет опыта пользователя.

Создание пользовательского интерфейса включает в себя следующие этапы:

- исследование, анализ и определение общих требований к пользовательскому интерфейсу;
- определение сценариев использования и пользовательской модели интерфейса;
- разработка прототипа пользовательского интерфейса;
- реализация пользовательского интерфейса;
- тестирование и оценка качества пользовательского интерфейса [18].

На первых двух этапах следует продумать структуру приложения, определить основные требования к пользовательскому интерфейсу. На этапе создания прототипа показываются основные аспекты функционирования ИС и базовые подходы к пользовательскому интерфейсу. Созданный прототип позволяет обсудить базовые решения с пользователем до этапа разработки. На этапе тестирования, к которому рекомендуется подключать пользователей, можно выявить не только ошибки программирования, но и оценить, насколько пользовательский интерфейс отвечает потребностям и ожиданиям пользователей. Итерационный подход позволяет устранять выявленные недостатки на последующих итерациях разработки [15, с. 7].

Пользовательский интерфейс - объект, обеспечивающий высокую информативность выводимой на экран информации, организующий удобство ее вывода и обработки пользователем автоматизированной системы, именно поэтому важно, чтобы сотрудникам было легко освоить работу в программе, гибкость же программного кода в случае необходимости позволит удовлетворить растущие требования к системе [10, с. 312].

Совместно с сотрудниками техникума был разработан макет пользовательского интерфейса информационной системы, который представлен на рисунке 11.

Расписание "Кулинар"

Вход для преподавателей

Учебная неделя: 17.04.2023 - 22.04.2023 Курс: 2 Группа: 201 повар-кондитер

Время	Понедельник 17.04.23			Вторник 18.04.23			Среда 19.04.23			Четверг 20.04.23			Пятница 21.04.23			Суббота 22.04.23		
	Кабинет	Преподаватель	Я	Кабинет	Преподаватель	М	Кабинет	Преподаватель	М	Кабинет	Преподаватель	М	Кабинет	Преподаватель	М	Кабинет	Преподаватель	М
8.30 - 9.15 9.25 - 10.10	биология Пузанова Л. Н.	209	Я							ИКТ Шаманаева И. С.	306	М	экономические и правовые основы Шаманаева И. С.	406	М	МДК 03.02 Жилина Е. В.	ДО	М
10.20 - 11.05 11.35 - 12.20	биология Пузанова Л. Н.	209	Я	история Решетникова О. Б.	410	М				ИКТ Шаманаева И. С.	306	М	экономические и правовые основы Шаманаева И. С.	406	М	МДК 03.02 Жилина Е. В.	ДО	М
12.40 - 13.25 13.35 - 14.20	биология Пузанова Л. Н.	209	Я	МДК 03.02 Жилина Е. В.	407	М	история Решетникова О. Б.	410	М	история Решетникова О. Б.	ДО	М	история Решетникова О. Б.	410	М			
14.30 - 15.15 15.25 - 16.10				МДК 03.02 Жилина Е. В.	407	М	физическая культура Чобанян Э. Л.	СП	М	история Решетникова О. Б.	ДО	М						
16.20 - 17.05 17.15 - 18.00							физическая культура Чобанян Э. Л.	СП	М									
18.10 - 18.55 19.05 - 19.50																		

Расписание "Кулинар" ©2023

Рисунок 11 – Макет пользовательского интерфейса

Разработка внешнего вида ИС является одним из важнейших этапов - ведь именно современный и понятный интерфейс обеспечивает легкое восприятие и обработку информации, а также минимизирует пользовательские операции.

Выводы по главе 2

Во второй главе был произведен выбор технологии логического моделирования информационной системы. Было принято решения для моделирования процессов использовать нотацию UML, а для моделирования логической модели данных использовать методологию IDEF1X.

После выбор технологий было произведено моделирование разрабатываемой информационной системы. Разработана диаграмма вариантов использования и логическая модель данных.

Были определены принципы разработки интерфейса информационной системы и совместно с сотрудниками техника был разработан макет пользовательского интерфейса.

Глава 3 Физическое проектирование информационной системы

3.1 Выбор архитектуры информационной системы учета учебного времени

За время разработки программных продуктов были разработаны различные системные подходы к проектированию программных систем. Это позволяет управлять сложностью программных систем и облегчает их сопровождение и модификацию в будущем, является ключевым фактором для создания надежных, масштабируемых и расширяемых программных систем.

В таблице 6 представлены основные виды архитектур [3, с. 10-18].

Таблица 6 – Виды архитектур

Вид архитектуры	Характеристика
Клиент-серверная архитектура	в этой архитектуре информационная система разделяется на клиентскую и серверную части. Клиентская часть отвечает за интерфейс пользователя и отправку запросов на сервер, а серверная часть отвечает за обработку запросов и предоставление необходимой информации.
Распределенная архитектура	в этой архитектуре информационная система состоит из нескольких независимых компонентов, которые могут располагаться на разных серверах и обмениваться информацией между собой.
Микросервисная архитектура	эта архитектура основана на использовании множества небольших сервисов, каждый из которых выполняет определенную функцию. Микросервисы могут быть разработаны на разных языках программирования и могут быть развернуты на разных серверах.
Централизованная архитектура	в этой архитектуре все компоненты системы находятся на одном сервере и взаимодействуют между собой через один канал связи.

Продолжение таблицы 6

Вид архитектуры	Характеристика
Одностраничное приложение (SPA - Single-Page Application)	это архитектура, где веб-приложение загружает только одну HTML-страницу, а все остальные компоненты загружаются асинхронно при необходимости, используя JavaScript. В SPA часто используется фронтенд-фреймворк, такой как Angular, React или Vue, для упрощения разработки и управления состоянием приложения.
Модель представление контроллер (MVC - Model-View-Controller)	это архитектура, которая разделяет приложение на три компонента: модель, представление и контроллер. Модель представляет данные, представление отображает эти данные пользователю, а контроллер обрабатывает запросы пользователя и управляет обновлением модели и представления. MVC часто используется в бэкенд-фреймворках, таких как Ruby on Rails, Laravel или Django.

Исходя из поставленных задач и требований к техническим возможностям системы сделан вывод о том, что веб-ориентированные архитектуры такие как SPA и MVC – будут наиболее подходящими для проектирования ИС.

Проведем сравнение архитектур SPA и MVC в таблице 7.

Таблица 7 – Сравнение архитектур SPA и MVC

Характеристика	SPA	MVC
Загрузка страниц	одна HTML-страница загружается, затем компоненты загружаются асинхронно при необходимости	каждая страница загружается отдельно
Логика клиента	клиентская логика управляется на стороне клиента при помощи JavaScript	клиентская логика управляется на стороне сервера, а клиент получает только отображение данных
Оптимизация	меньше запросов к серверу, но больше JavaScript на стороне клиента, что может замедлить работу браузера	больше запросов к серверу, но меньше JavaScript на стороне клиента

Продолжение таблицы 7

Характеристика	SPA	MVC
Скорость переходов	быстрые переходы между компонентами без перезагрузки страницы	необходимость перезагрузки страницы при переходе на другую страницу
Разделение ответственности	ответственность разделена между клиентом и сервером, приложение может иметь только фронтенд-часть	ответственность разделена между клиентом и сервером, сервер может иметь более сложную бизнес-логику
Кэширование	может быть сложно кэшировать динамический контент	для каждой страницы можно кэшировать результаты

Сравнительный анализ убедил нас остановиться на архитектуре MVC.

MVC (Model-View-Controller) — это шаблон проектирования, который разделяет приложение на три основных компонента: Модель, Представление и Контроллер [9].

MVC позволяет разделить логику приложения на три отдельных компонента, что упрощает разработку, тестирование и сопровождение приложения. Кроме того, этот шаблон позволяет легко вносить изменения в один из компонентов без влияния на остальные, что упрощает масштабирование приложения [1].

- модель (Model) представляет данные и бизнес-логику приложения. Она отвечает за хранение данных, их обработку и предоставление их для отображения в Представлении;
- представление (View) представляет пользовательский интерфейс. Она отображает данные, которые были получены из Модели, и отвечает за предоставление интерфейса пользователя;
- контроллер (Controller) отвечает за обработку пользовательского ввода и управление Моделью и Представлением. Он получает запросы от Представления, обрабатывает их, взаимодействует с

Моделью для получения данных и обновляет Представление на основе этих данных.

На рисунке 12 визуальна представлена структура MVC.

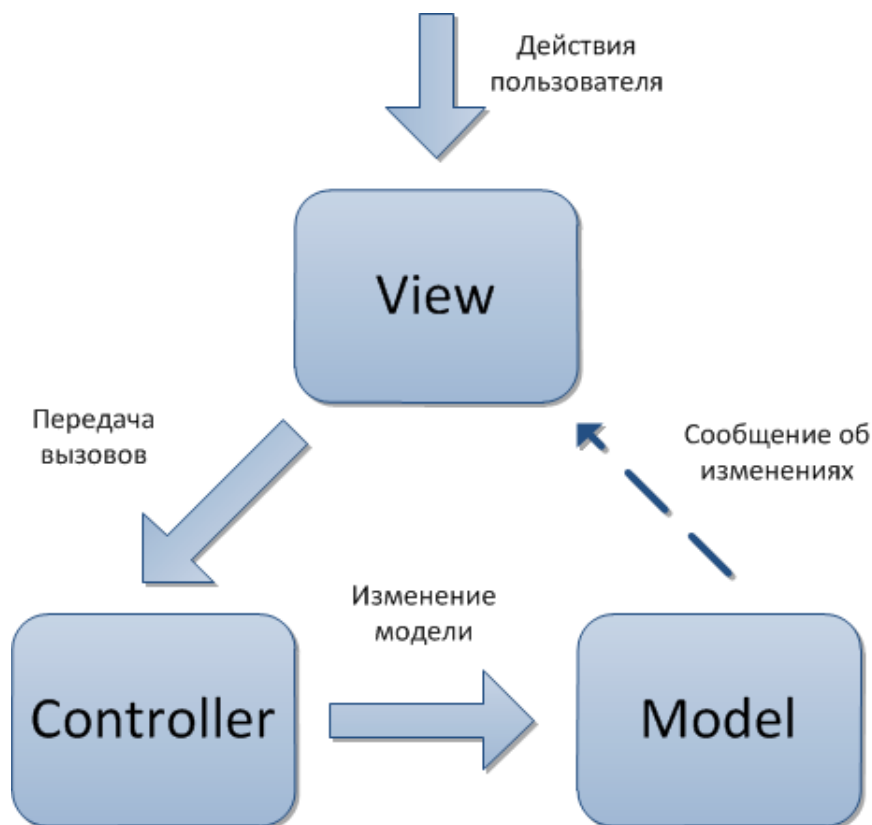


Рисунок 12 – Структура MVC

Самое очевидное преимущество, которое мы получаем от использования концепции MVC — это чёткое разделение логики представления (интерфейса пользователя) и логики приложения.

Поддержка различных типов пользователей, которые используют различные типы устройств является общей проблемой наших дней. Предоставляемый интерфейс должен различаться, если запрос приходит с персонального компьютера или с мобильного телефона. Модель возвращает одинаковые данные, единственное различие заключается в том, что контроллер выбирает различные виды для вывода данных.

Помимо изолирования видов от логики приложения, концепция MVC существенно уменьшает сложность больших приложений. Код получается гораздо более структурированным, и, тем самым, облегчается поддержка, тестирование и повторное использование решений [9].

Концепция MVC широко используется в веб-разработке для создания приложений благодаря универсальности, гибкости и простоте применения.

3.2 Выбор базы данных для информационной системы учета учебного времени

Для разработки информационной системы необходимо выбрать базу данных, в соответствии с разработанной ранее логической моделью она должна быть реляционной.

Реляционные базы хранят данные в таблицах, которые могут быть связаны друг с другом при помощи ключей. Каждая строка таблицы представляет отдельную запись, а каждый столбец содержит отдельное поле данных. Схема базы данных в реляционных СУБД определяет типы данных для каждого столбца таблицы, и она строго контролирует структуру и целостность данных [20, с. 180].

Нереляционные базы данных, используют более свободный подход к хранению данных. Данные могут быть организованы в виде документов, ключ-значений, графов или колонок. В отличие от реляционных баз данных, нереляционные базы не требуют строгой схемы данных, что позволяет более гибко изменять структуру данных в процессе работы приложения.

При выборе СУБД для нашей ИС важно, чтобы она имела либо свободную лицензию, либо бесплатную для некоммерческого использования.

На 2023 год самыми популярными реляционными базами данных являются:

- MySQL
- PostgreSQL

- Microsoft SQL Server
- Oracle

MySQL - это бесплатная и открытая реляционная СУБД, которая широко используется веб-приложениями и дружелюбна к разработчикам. MySQL поддерживает множество операционных систем, включая Linux, Windows и Mac OS X, и имеет множество драйверов, что делает ее простой в использовании в любом языке программирования [4, с. 13].

PostgreSQL - это мощная реляционная СУБД с открытым исходным кодом, которая также может быть бесплатно использована. PostgreSQL предлагает множество функций, таких как хранимые процедуры, триггеры, географические запросы, что делает ее подходящей для использования в крупных проектах.

Microsoft SQL Server - это коммерческая реляционная СУБД, разработанная Microsoft. Она предлагает богатый набор функций и инструментов для администрирования баз данных. SQL Server интегрируется хорошо с другими продуктами Microsoft, такими как Excel, SharePoint и др.

Oracle - это коммерческая реляционная СУБД, разработанная Oracle Corporation. Она является одной из самых популярных и масштабируемых реляционных СУБД. Oracle поддерживает множество функций, таких как кластеризация, резервное копирование, восстановление и синхронизацию данных. Она также предлагает богатый набор инструментов и функций для администрирования баз данных.

Проведем сравнение СУБД в таблице 8.

Таблица 8 – Сравнение СУБД

Характеристика	MySQL	PostgreSQL	Microsoft SQL Server	Oracle
Лицензия	open Source	open Source	коммерческая	коммерческая
Модель данных	табличная (связь между таблицами основана на ключах)	табличная (связь между таблицами основана на ключах)	табличная (связь между таблицами основана на ключах)	табличная (связь между таблицами основана на ключах)
Язык запросов	SQL (Structured Query Language)	SQL (Structured Query Language)	SQL (Structured Query Language)	SQL (Structured Query Language)
Производительность	очень высокая	высокая	высокая	высокая
Масштабирование	горизонтальное и вертикальное	горизонтальное и вертикальное	горизонтальное и вертикальное	горизонтальное и вертикальное
Транзакции	поддерживаются и обеспечивают целостность данных	поддерживаются и обеспечивают целостность данных	поддерживаются и обеспечивают целостность данных	поддерживаются и обеспечивают целостность данных
Резервное копирование	поддерживается	поддерживается	поддерживается	поддерживается
Поддержка хранимых процедур и триггеров	поддерживается	поддерживается	поддерживается	поддерживается
Сложность настройки и администрирования	относительно простая	относительно сложная	средняя	средняя

MySQL – оказалась наиболее подходящей СУБД по большинству характеристик.

3.3 Разработка программного обеспечения информационной системы

3.3.1 Выбор инструментов реализации

В первую очередь необходимо определиться с платформой, на основе которой будет разрабатываться приложение.

Фреймворки, использующие паттерн MVC (Model-View-Controller) для разработки веб-приложений, помогают упростить и ускорить процесс разработки.

Преимущества фреймворков, использующих паттерн MVC, включают:

- Разделение ответственности: MVC позволяет разделить логику приложения на отдельные компоненты, что упрощает понимание и сопровождение кода;
- Легкая поддержка и расширение: благодаря разделению на компоненты, изменение или добавление новых функций становится более простым, что облегчает сопровождение проекта;
- Улучшенная параллельная разработка: разделение на компоненты позволяет нескольким разработчикам работать над различными частями приложения параллельно.

Ниже приведены некоторые популярные фреймворки, которые поддерживают паттерн MVC:

- Java: Spring MVC, Struts, Play Framework;
- Python: Django, Flask, Pyramid;
- PHP: Laravel, CodeIgniter, Yii;
- JavaScript: AngularJS, React, Backbone.js;
- NET: ASP.NET MVC, NancyFx, DotNetty;
- Go: Revel, Beego, Buffalo.

В таблице 9 произведено сравнение всех фреймворков, каждый из которых предоставляет разработчикам инструменты для упрощения разработки приложений, используя паттерн MVC. Они помогают сократить время разработки и повышают качество конечного продукта.

Таблица 9 – Сравнение платформ

Язык программирования	Появление	Направление	Сложность изучения	Популярность	Комьюнити	Применение
Java	1995	общего назначения, кроссплатформенный	средняя	высокая	большое	веб-приложения, мобильные приложения, игры
Python	1991	Общего назначения, научные вычисления	Низкая	Очень высокая	Очень большое	Научные и математические вычисления, веб-разработка, анализ данных
PHP	1995	Веб-разработка	Низкая	Высокая	Большое	Веб-разработка, создание сайтов, CMS
JavaScript	1995	Веб-разработка	Средняя	Очень высокая	Очень большое	Веб-разработка, создание веб-приложений, игры
.NET	2002	Общего назначения, кроссплатформенный	Средняя	Высокая	Большое	Веб-разработка, мобильные приложения, научные вычисления
Go	2009	Многопоточное программирование, обработка больших данных	Средняя	Высокая	Большое	Бэкенд-разработка, обработка больших данных, микросервисная архитектура

В связи с высокой популярностью и низкой сложностью изучения языка выбор в пользу программирования системы на языке Python.

Разработка ИС будет произведена с использованием очень популярного фреймворка, написанного на языке Python – Django.

Django - это высокоуровневый веб-фреймворк для разработки веб-приложений на языке Python. Он содержит множество инструментов и

библиотек для быстрой и эффективной разработки веб-приложений. Django включает в себя множество функций, таких как Объектно-реляционное отображение, автоматическую админ-панель, систему маршрутизации URL, шаблонизацию, систему аутентификации и авторизации, и многое другое [7, с. 18].

Django также использует паттерн проектирования Model-View-Controller (MVC), который помогает разделить логику приложения на три отдельных компонента: модели, представления и контроллеры. Это делает код более читаемым и поддерживаемым.

Преимущества Django:

- открытый исходный код;
- универсальность, любой веб-проект можно сделать на Django;
- высокая безопасность из коробки;
- поддерживает MVT;
- подходит для высоконагруженных проектов;
- очень популярный;
- бесплатный.

Благодаря своей мощной функциональности и гибкости, Django является одним из наиболее популярных фреймворков для веб-разработки на Python. Он используется такими крупными компаниями, как Instagram, Pinterest и Mozilla.

Помимо фреймворка в разработке мы будем использовать интегрированную среду разработки (IDE).

Существует множество IDE для языка программирования Python. Некоторые из самых популярных IDE для Python:

- PyCharm – это одна из самых популярных IDE для Python, созданная компанией JetBrains. PyCharm предоставляет широкий набор инструментов для быстрой разработки, тестирования и отладки Python-приложений;

- Visual Studio Code – это бесплатная, легковесная и кроссплатформенная IDE для разработки Python, разработанная компанией Microsoft. VS Code имеет множество расширений и интеграций, которые позволяют настроить IDE для вашего стиля разработки;
- Spyder – это научная IDE для Python, предназначенная для анализа данных, научных вычислений и машинного обучения. Она включает в себя интегрированные средства для работы с библиотеками NumPy, SciPy, Pandas и другими;
- Jupyter Notebook – это интерактивная IDE для Python, предназначенная для научных вычислений и анализа данных. Jupyter Notebook позволяет объединять код, текст, графики и другие элементы в единый документ.

PyCharm – это интегрированная среда разработки (IDE) для языка программирования Python, созданная компанией JetBrains. Она предоставляет широкий набор инструментов для быстрой разработки, тестирования и отладки Python-приложений [13, с. 39].

Некоторые из основных возможностей PyCharm:

- редактор кода: PyCharm предоставляет мощный редактор кода с подсветкой синтаксиса, автодополнением, функцией перехода к определению и другими возможностями, которые позволяют ускорить процесс разработки.
- отладчик: PyCharm имеет встроенный отладчик, который позволяет легко отслеживать ошибки в коде и находить их местоположение. Отладчик также поддерживает удаленное отладочное соединение.
- инструменты тестирования: PyCharm включает в себя инструменты для создания и запуска тестов, включая поддержку библиотеки unittest, pytest и других.
- интеграция с системами контроля версий: PyCharm поддерживает интеграцию с системами контроля версий, такими как Git, SVN,

Mercurial, Perforce и другими. Это позволяет легко отслеживать изменения в коде и управлять проектами.

- анализ кода: PyCharm предоставляет множество инструментов для анализа кода, включая статический анализатор кода, поиск неиспользуемых импортов, анализ производительности и другие.
- интеграция с виртуальными окружениями: PyCharm поддерживает создание и управление виртуальными окружениями Python, что позволяет изолировать зависимости проекта и упрощает управление версиями пакетов.

PyCharm доступен в двух версиях: Community Edition (бесплатная версия) и Professional Edition (платная версия с более широким набором функций). В случае нашего проекта функционала Community Edition нам будет достаточно.

3.3.2 Реализация информационной системы

Перед началом разработки необходимо разобраться как устроен Django. Рассмотрим алгоритм обработки запросов в Django изображенный на рисунке 13.

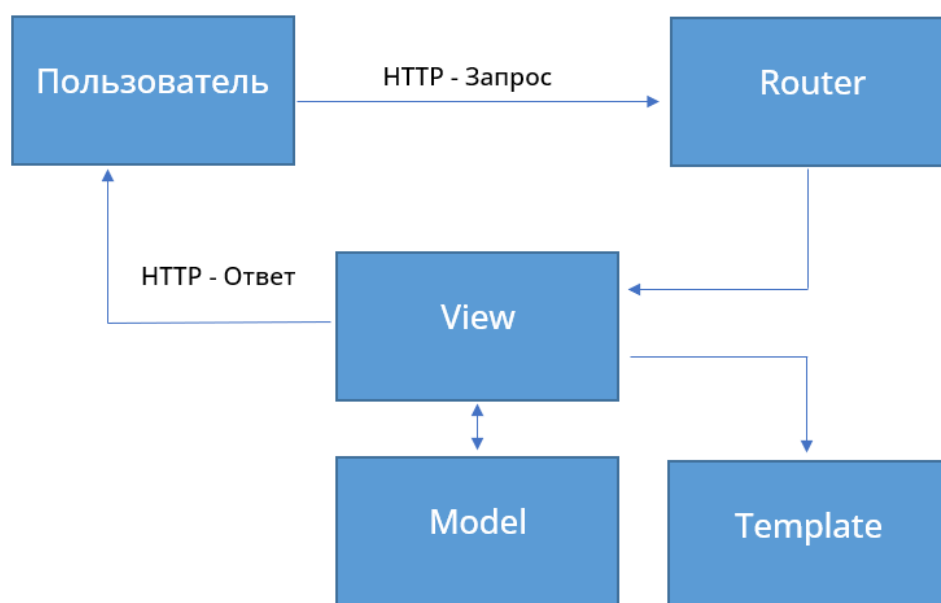


Рисунок 13 – Алгоритм обработки запросов в Django

На сервер поступает запрос от пользователя, который попадает на роутер Django. В рамках роутера Django определяет какая функция должна быть вызвана. Доступные функции для вывода страниц находятся во View, который используется как контроллер в шаблоне MVT. Далее View начинает работу с моделью, которая отвечает за взаимодействие с базой данных, вызывает объекты и функции, отбирает данные. После, когда данные отобраны происходит вывод на Template. Template в виде html страницы со специальными программными вставками в виде переменных шаблона, а также различные управляющие конструкции. Готовый шаблон возвращается на контроллер которые преобразуется в HTML страницу и отправляется в виде ответа на запрос пользователю [12, с. 23-25].

Разработка приложений в Django происходит в несколько этапов:

- создание проекта;
- настройка проекта;
- создание моделей данных;
- создание админ-панели;
- добавление тестовых данных;
- создание шаблонов для вывода страниц сайта;
- добавление необходимой функциональности.

Для создания проекта Django в IDE PyCharm необходимо создать проект с чистым окружением, добавить интерпретатор Django и запустить команду установки Django. Во время разработки рекомендуется создавать для разного функционала разные приложения [22, с. 22]. В нашем случае мы создадим два приложения, одно приложение будет предназначаться для администратора расписания, а второе для студентов. На рисунке 14 изображена структура файлов созданного проекта Django.

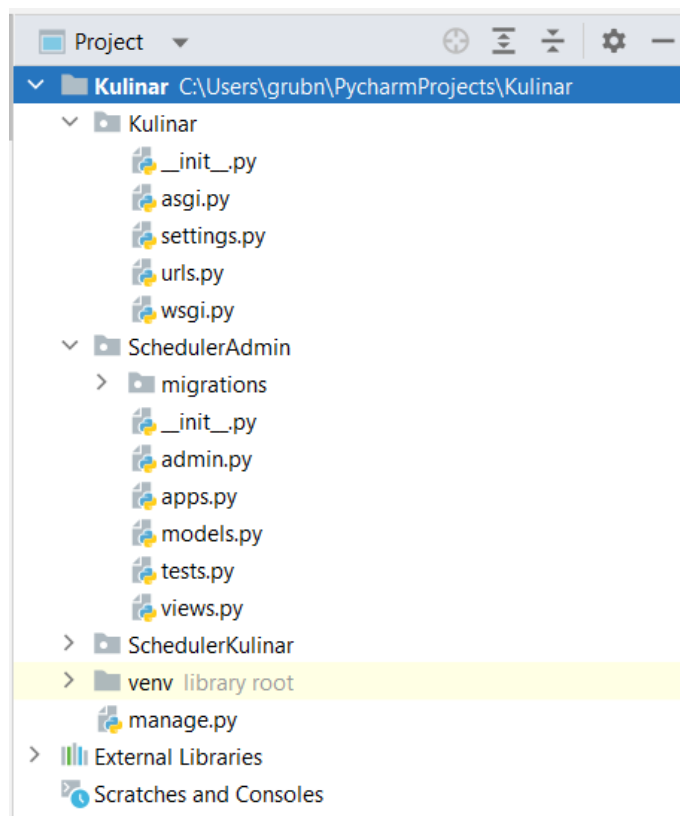


Рисунок 14 – Структура созданного проекта

Настройка проекта производится в файле `settings.py` который находится в корневой папке проекта. Для начала разработки нас интересуют такие параметры как `DEBUG`, `ALLOWED_HOSTS`, `ROOT_URLCONF`, `DATABASES`. Параметр `DEBUG` включает режим отладки приложения, у параметра `ALLOWED_HOSTS` указываются разрешенные адреса хостов, у параметра `ROOT_URLCONF` указывается путь к роутеру, который будет перенаправлять запросы, в `DATABASES` указываются данные для подключения к базе данных [17].

Модель – это объект в Python, который служит для управления данными. Одной модели соответствует одна таблица. В модели описываются все поля создаваемой таблицы. У каждого поля указывается тип и вспомогательные функции такие как проверка данных и форматирование данных [16, с. 10-13]. На основании описанной модели инструментами

Django будут созданы таблицы в базе данных. На рисунке 15 представлен фрагмент кода, описывающий модель с данными по преподавателям.

```
class Teachers(models.Model):
    surname = models.CharField(max_length=50, verbose_name='Фамилия')
    name = models.CharField(max_length=50, verbose_name='Имя')
    patronymic = models.CharField(max_length=50, verbose_name='Отчество')
    teacher_area = models.ForeignKey('Departaments', on_delete=models.SET_NULL, blank=True, null=True, verbose_name='Отделение')
    teacher_classroom = models.ForeignKey('Classrooms', on_delete=models.SET_NULL, blank=True, null=True, verbose_name='Кабинет')
    email = models.EmailField(null=True, blank=True, max_length=50, verbose_name='email')

    class Meta:
        ordering = ['surname']
        verbose_name = 'Преподавателя'
        verbose_name_plural = 'Преподаватели'

    def __str__(self):
        return '{0} {1} {2}'.format(self.surname, self.name, self.patronymic)
```

Рисунок 15 – Модель таблицы Преподаватели

На этапе создания админ-панели первым делом создается пользователь с правами администратора, это делается через консоль проекта встроенными средствами Django [22]. Для управления базой данных с помощью админ панели необходимо зарегистрировать созданные модели в файле admin.py.

На рисунке 16 представлен внешний вид окна авторизации админ-панели Django.

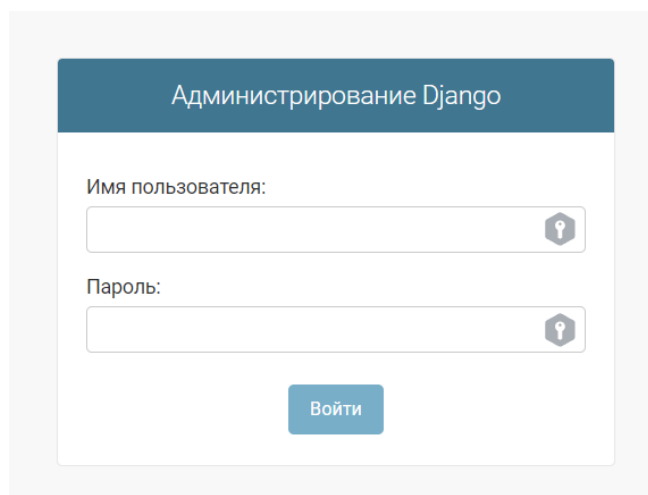


Рисунок 16 – Окно авторизации админ-панели Django

В админ-панели есть функционал необходимый для ввода и изменения данных по учебным занятиям, дисциплинам, группам, преподавателям и так далее. На рисунке 17 представлен внешний вид панели администратора системы.

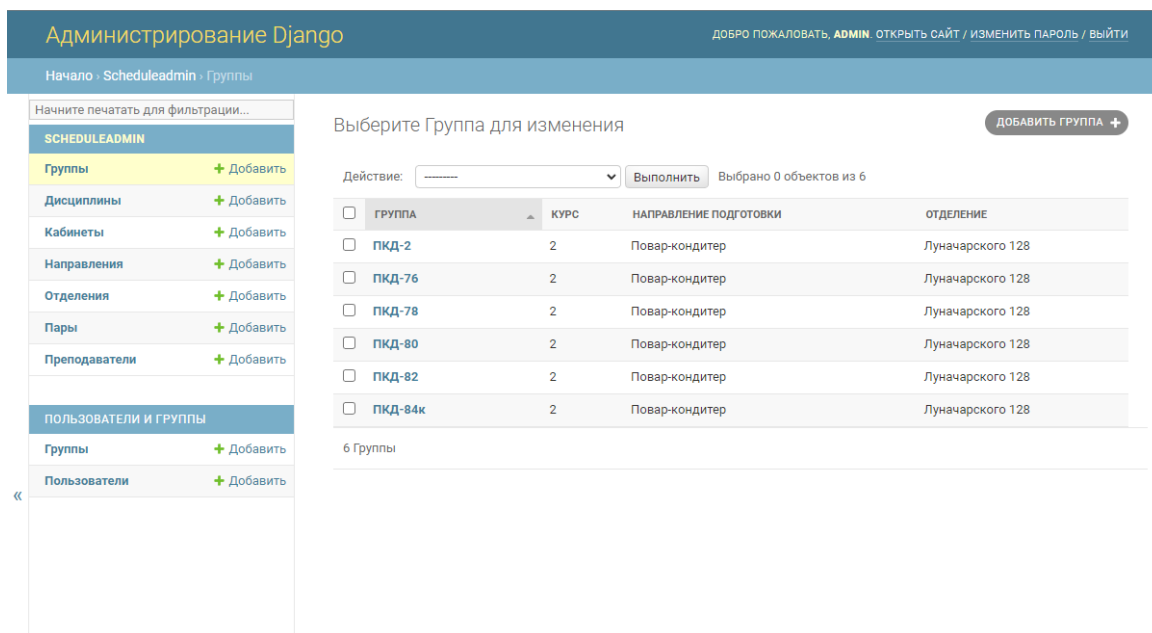


Рисунок 17 – Админ-панель Django

Для работы системы нам необходимо создать справочник, в котором будут храниться такие данные как:

- список групп студентов, где указываются номер группы, курс, направление подготовки и основной учебный корпус;
- дисциплины, где указывается название дисциплины, направление подготовки студентов, курс на котором студенты будут изучать данную дисциплину, основной преподаватель если предмет ведет только один преподаватель, основной кабинет если изучение дисциплины проходит только в одном кабинете, количество часов, которые необходимо провести в первом семестре и количество часов, которые необходимо провести во втором семестре;

- кабинеты, где указываются номер кабинета и учебный корпус, в котором он находится;
- направления подготовки содержат список имеющихся направлений подготовки;
- отделения содержат список учебных корпусов;
- преподаватели, где указывается ФИО преподавателей, основной кабинет преподавателя если такой имеется и основное учебное отделение.

Также необходимо разработать шаблоны, в которые будут подставляться данные из базы данных. Нужны основной шаблон, который будет повторяться на всех страницах и уникальные для отображения данных.

В основном шаблоне присутствуют такие элементы как шапка сайта, в которой указывается название сайта, меню сайта и подвал – нижняя часть сайта.

Уникальные шаблоны нужны для отображения запросов пользователя.

Вот список необходимых уникальных шаблонов:

- шаблон отображения расписания для студентов;
- шаблон отображения расписания для администратора расписания;
- шаблон страницы изменения расписания;
- шаблон страницы с отчетами по группам.

Последним этапом разработки информационной системы будет реализация функциональных возможностей.

Основная функциональная возможность разрабатываемой ИС это вывод информации из базы данных по запросам пользователей. Так на рисунке 18 представлен внешний вид интерфейса готового расписания с фильтрацией по преподавателю, на рисунке 19 с фильтрацией по группе.

По группам

По преподавателям

Расписание по преподавателям

Преподаватель: Период:

Понедельник 13.03.2023		Вторник 14.03.2023		Среда 15.03.2023		Четверг 16.03.2023		Пятница 17.03.2023		Суббота 18.03.2023				
№	Группа / Дисциплина	Кабинет / Площадка	№	Группа / Дисциплина	Кабинет / Площадка	№	Группа / Дисциплина	Кабинет / Площадка	№	Группа / Дисциплина	Кабинет / Площадка			
1	T-56 Русский язык и культура речи	303 Лун 128	1	ПКД-58 Русский язык и культура речи	303 Лун 128	1	ПКД-84к Литература	303 Лун 128	1	ПКД-76 Литература	303 Лун 128	1	T-54 Русский язык и культура речи	дистант
2	T-56 Русский язык и культура речи	303 Лун 128	2	ПКД-58 Русский язык и культура речи	303 Лун 128	2	ПКД-84к Литература	303 Лун 128	2	ПКД-76 Литература	303 Лун 128	2	T-64 Русский язык и культура речи	дистант
3	T-54 Русский язык и культура речи	303 Лун 128	3	ПКД-76 Литература	303 Лун 128	3			3	ПКД-62 Русский язык и культура речи	303 Лун 128	3	ПКД-84к Русский язык и культура речи	
4			4	ПКД-76 Литература	303 Лун 128	4			4	ПКД-62 Русский язык и культура речи	303 Лун 128	4	ПКД-84к Русский язык и культура речи	
5			5	ПКД-60 Русский язык и культура речи	303 Лун 128	5			5			5		
6			6	ПКД-60 Русский язык и культура речи	303 Лун 128	6			6			6		

Расписание Кулинар 2023

Рисунок 18 – Интерфейс сформированного расписания с фильтрацией по преподавателю

По группам

По преподавателям

Расписание

Курс: Группа: Период:

Понедельник 13.03.2023		Вторник 14.03.2023		Среда 15.03.2023		Четверг 16.03.2023		Пятница 17.03.2023		Суббота 18.03.2023				
№	Дисциплина / Преподаватель	Кабинет / Площадка	№	Дисциплина / Преподаватель	Кабинет / Площадка	№	Дисциплина / Преподаватель	Кабинет / Площадка	№	Дисциплина / Преподаватель	Кабинет / Площадка			
1	МДК 07.01 Лабораторная работа Золотарева И.В.	ЛБ Лун 128	1	Математика Вавилова Е.В.	312 Лун 128	1	Математика Вавилова Е.В.	312 Лун 128	1	Математика Вавилова Е.В.	312 Лун 128	1		
2	МДК 07.01 Лабораторная работа Золотарева И.В.	ЛБ Лун 128	2	Математика Вавилова Е.В.	312 Лун 128	2	Математика Вавилова Е.В.	312 Лун 128	2	Математика Вавилова Е.В.	312 Лун 128	2		
3	МДК 07.01 Лабораторная работа Золотарева И.В.	ЛБ Лун 128	3	Литература Менщикова И.Ю.	303 Лун 128	3	Физическая культура Чебожян Э.Л.	сп-зал М/С 16	3	Экология Мальчихин О.Н.	306 Лун 128	3	Экология Мальчихин О.Н.	306 Лун 128
4	МДК 07.01 Лабораторная работа Золотарева И.В.	ЛБ Лун 128	4	Литература Менщикова И.Ю.	303 Лун 128	4	Физическая культура Чебожян Э.Л.	сп-зал М/С 16	4	Экология Мальчихин О.Н.	306 Лун 128	4	Экология Мальчихин О.Н.	306 Лун 128
5	МДК 07.01 Лабораторная работа Золотарева И.В.	ЛБ Лун 128	5			5			5			5		
6			6			6			6			6		

Расписание Кулинар 2023

Рисунок 19 – Интерфейс сформированного расписания с фильтрацией по группе

Одно из важных требований к разрабатываемой ИС - возможность генерировать различные отчеты. На рисунке 20 представлен интерфейс администратора расписания при формировании отчёта по группе.

Отчет по группам

Курс: Группа: Период:

№ п/п	Наименование дисциплин	Ф.И.О.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	Текущий месяц	Проведено всего	Учебный план	
1	Русский язык	Менщикова																																0	0	
2	Литература	Менщикова				4														4														8	8	
3	Иностранный язык 1 подгруппа	Скобарина							4												4			4										12	12	
4	Иностранный язык 2 подгруппа	Скобарина																																0	0	
5	Математика	Вавилова						4	4					4		4						4	4										24	24		
6	Физическая культура	Касимцева				4						2																					6	6		
7	Основы безопасности жизнедеятельности	Щербakov																															0	0		
8	Родная литература	Менщикова																															0	0		
9	Экологические основы природопользования	Мальчихин													4	4																	8	8		
10	Микробиология, физиология питания, санитария и гигиена	Вахрушева																															0	0		
11	Организация хранения и контроль запасов и сырья	Якорнова							4		4										4		6										18	18		
12	Техническое оснащение организаций питания	Лысова																															0	0		
13	МДК 07.01 Выполнение работ по профессии повар	Золотарева				4														6				4									14	14		
14	МДК 07.02 Выполнение работ по профессии кондитер	Мамаева											4		6																		10	10		
15	МДК 07.03 Выполнение работ по профессии пекарь	Нигаматанова																															0	0		

Расписание Кулинар 2023

Рисунок 20 – Интерфейс сформированного отчёта по группе

Внешний вид интерфейса раздела редактирования представлен на рисунке 21.

Редактор расписания

Курс: Период:

Загрузить из файла:

ПКД-76		ПКД-78		ПКД-80		ПКД-82		ПКД-84к		ПКД-2	
Понедельник 13.03.2023	ПКД-76	Понедельник 13.03.2023	ПКД-78	Понедельник 13.03.2023	ПКД-80	Понедельник 13.03.2023	ПКД-82	Понедельник 13.03.2023	ПКД-84к	Понедельник 13.03.2023	ПКД-2
№	Дисциплина / Преподаватель	№	Дисциплина / Преподаватель	№	Дисциплина / Преподаватель	№	Дисциплина / Преподаватель	№	Дисциплина / Преподаватель	№	Дисциплина / Преподаватель
1	МДК 07.01 Лабораторная работа Золотарева И.В.	1	Основы безопасности жизнедеятельности Щербakov С.В.	1	Практика	1	Практика	1		1	МДК 04.02 Редкина Е.А.
2	МДК 07.01 Лабораторная работа Золотарева И.В.	2	Основы безопасности жизнедеятельности Щербakov С.В.	2		2		2		2	МДК 04.02 Редкина Е.А.
3	МДК 07.01 Лабораторная работа Золотарева И.В.	3	Физическая культура Касимцева Е.А.	3		3		3		3	МДК 04.02 Редкина Е.А.
4	МДК 07.01 Лабораторная работа Золотарева И.В.	4	Физическая культура Касимцева Е.А.	4		4		4	Орг. хранение и контр. зап. и сырая Якорнова Е.В.	4	Яси 1/4
5	МДК 07.01 Лабораторная работа Золотарева И.В.	5		5		5		5	Орг. хранение и контр. зап. и сырая Якорнова Е.В.	5	Яси 1/4
6		6		6		6		6	Орг. хранение и контр. зап. и сырая Якорнова Е.В.	6	Яси 1/4
Вторник 14.03.2023	ПКД-76	Вторник 14.03.2023	ПКД-78	Вторник 14.03.2023	ПКД-80	Вторник 14.03.2023	ПКД-82	Вторник 14.03.2023	ПКД-84к	Вторник 14.03.2023	ПКД-2
№	Дисциплина / Преподаватель	№	Дисциплина / Преподаватель	№	Дисциплина / Преподаватель	№	Дисциплина / Преподаватель	№	Дисциплина / Преподаватель	№	Дисциплина / Преподаватель
1	Математика Вавилова Е.В.	1	МДК 07.02 Лабораторная работа Мамаева Е.О.	1	Практика	1	Практика	1	Основы безопасности жизнедеятельности Щербakov С.В.	1	МДК 04.02 Редкина Е.А.
2	Математика Вавилова Е.В.	2	МДК 07.02 Лабораторная работа Мамаева Е.О.	2		2		2	Основы безопасности жизнедеятельности Щербakov С.В.	2	МДК 04.02 Редкина Е.А.
3	Литература Менщикова И.Ю.	3	МДК 07.02 Лабораторная работа Мамаева Е.О.	3		3		3	Основы безопасности жизнедеятельности Щербakov С.В.	3	МДК 04.02 Редкина Е.А.
4	Литература Менщикова И.Ю.	4	МДК 07.02 Лабораторная работа Мамаева Е.О.	4		4		4		4	
5		5	МДК 07.02 Лабораторная работа Мамаева Е.О.	5		5		5		5	
6		6	МДК 07.02 Лабораторная работа Мамаева Е.О.	6		6		6		6	

Рисунок 21 – Интерфейс раздела «Редактор расписания»

Таким образом с использованием реализованной ИС учета учебного времени специалист не только обладает доступом к базе данных учреждения, но также может составлять и редактировать расписание учебных занятий через «Редактор расписания».

3.4 Тестирование разработанной системы

Тестирование приложений – это процесс проверки функциональности, надежности и безопасности приложений. Во время разработки приложения за надежность и безопасность отвечает фреймворк Django, он обладает рядом встроенных механизмов, направленных на обеспечение надежности и безопасности. Нам же в свою очередь необходимо провести проверку функциональности. Она проводится с целью выявления ошибок и недочетов в работе приложения и улучшения качества пользовательского опыта [19, с. 1].

На этапе тестирования нам необходимо проверить на работоспособность весь имеющийся функционал приложения, а именно:

- составление расписания;
- изменение расписания;
- составление отчетов;
- отображение расписания для студентов;
- отображение расписания для преподавателей;

Для тестирования веб-приложений используются различные методы и инструменты, включая ручное и автоматизированное тестирование. Мы будем использовать ручное тестирование.

Тестирование составления расписания включает в себя проверку ручного ввода расписания и загрузку расписания из таблицы excel.

Изменение расписания происходит в ручном режиме, на этом этапе необходимо убедиться, что при изменении расписания данные обновляются в базе данных и отображаются изменения в основном расписании.

Проверка составления отчетов происходит путем выбора различных групп и периодов обучения.

Проверка отображения готового расписания схожа с проверкой составления отчетов, необходимо выбрать нужный курс и группу, убедиться, что для каждой группы подставляются свои данные на примере нескольких групп.

По результатам тестирования были выявлены недочеты в отображении данных, которые были устранены.

3.5 Описание функциональности информационной системы учета учебного времени

Финальная функциональность системы полностью соответствует определенным требованиям. Рабочие сценарии разделяются на два основных потока: работа от лица студента и работа от лица администратора расписания.

Студент может производить поиск и просмотр расписания интересующих его групп. Для этого ему нужно зайти на сайт, где расположено расписание и воспользоваться фильтрами для поиска нужной группы.

Администратору расписания для работы с расписанием необходимо авторизоваться в системе. У администратора есть функционал для просмотра расписания, внесения изменений в расписание и составления отчетов, где указаны дни и количество проведенных часов у конкретной группы.

Во вкладке изменение расписания можно изменять готовое расписание и создавать новое. Для изменения расписания необходимо выбрать необходимую группу и учебную неделю. Создание нового расписания может происходить как в ручном режиме, так и импортироваться из файла excel. Отчеты формируются за выбранный месяц для одной группы.

3.6 Расчет экономической эффективности внедрения системы

Расчет экономической эффективности является важным инструментом для оценки финансовой состоятельности и успешности проекта по разработке и внедрению ИС учета учебного времени. Он помогает снизить риски и принять обоснованное решение, основанное на бизнес-целях и ожидаемых выгодах.

Для расчета экономической эффективности разработки и внедрения ИС учета учебного времени в техникуме необходимо:

- провести анализ затрат;
- сравнить показатели экономической эффективности текущего решения и внедряемого;
- сделать вывод на основе сравнения эффективности.

В таблице 10 составлен план разработки ИС.

Таблица 10 – Задачи на разработку ИС

Задача	Затраты, дней	Затраты, часов
Разработка требований	5	40
Логическое моделирование информационной системы	5	40
Физическое моделирование информационной системы	5	40
Разработка пользовательского интерфейса	5	40
Разработка Backend-а	10	80
Разработка Frontend-а	5	40
Тестирование и отладка системы	5	40
Итого:	40	320

В расчете стоимости разработки необходимо учитывать не только трудовые ресурсы, но и материальные затраты. В данном случае разработчик уже обладает персональным компьютером и необходимым программным

обеспечением для выполнения работ. Кроме того, сотрудник работает удаленно и предприятие не несет дополнительных затрат на аренду офиса.

В таблице 11 представлены затраты на заработную плату, которые необходимо учесть при разработке проекта.

Таблица 11 – Расходы на разработку ИС

Роль	Человеко-часов	Медианная зарплата в месяц в рублях	Итого
Python разработчик	80	100 000	50 000
Тестировщик	40	80 000	20 000
Дизайнер	80	60 000	30 000
Аналитик	120	120 000	80 000
Общий итог после налогов		126 000	
Приблизительный итог до налогов		180 000	

Расчет производился на основе профиля сотрудника, количества времени, затрачиваемого на работу, и средней заработной платы, полученной из статистических данных, представленных в сервисе Хабр Карьера за второе полугодие 2022 года [8].

Итоговую сумму до уплаты налогов можно подсчитать приблизительно исходя из НДФЛ в 13% и остальных взносов в 30%. К остальным взносам относятся взносы в Пенсионный фонд (ПФР), Фонд обязательного медицинского страхования (ФОМС), Фонд социального страхования (ФСС), Фонд социального страхования за травматизм (ФСС). Нужно учитывать, что в законодательстве предусмотрены разные процентные ставки по налогу и взносам в зависимости от статуса работника и других факторов. Поэтому ставка в 30% является приблизительной.

Таким образом, можно считать, что стоимость разработки информационной системы составляет 180000 рублей.

Основной целью разработки ИС учета учебного времени для техникума «Кулинар» была автоматизация формирования составления расписания, в результате чего нагрузка на специалиста снизилась. Достижение данной цели возможно посредством сбора данных в единую систему и использования этой системы, освобождая их от бумажной волокиты.

Выгоды от использования ИС:

- уменьшение количества ошибок при составлении расписания;
- уменьшение времени, необходимого на составление расписания;
- возможность быстрого корректирования расписания при изменении графика занятий;
- повышение удобства и доступности расписания для студентов и преподавателей.

На рисунке 22 изображено сравнение время составления расписания занятий до внедрения ИС и после. На диаграмме видно, что продолжительность составления расписания уменьшилась примерно на 40%.

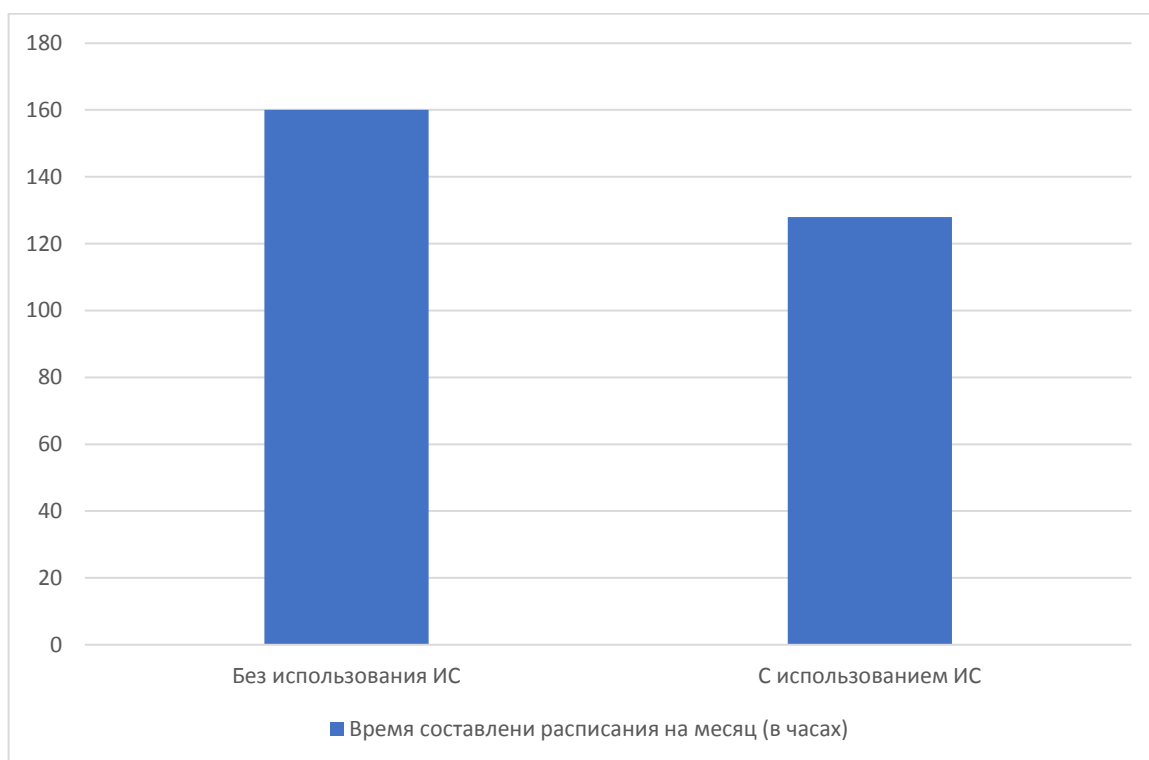


Рисунок 22 – Время составления расписания

Зарботная плата диспетчера по составлению расписания составляет 23 000 рублей в месяц после вычета налогов – это 160 рабочих часов в месяц и 143,75 рублей в час.

К трудовым характеристикам относится ΔT – абсолютное снижение трудовых затрат за месяц (1).

$$\Delta T = T_0 - T_1, \quad (1)$$

где T_0 – трудовые затраты, рассчитанные в часах за месяц при использовании текущего решения;

T_1 – трудовые затраты, рассчитанные в часах за месяц при использовании нового решения.

Получаем: $\Delta T = 160 - 128 = 32$ ч.

Коэффициент относительного снижения трудовых затрат K_T рассчитаем по формуле (2).

$$K_T = \frac{\Delta T}{T_0} \cdot 100, \quad (2)$$

Получаем: $K_T = 32 / 160 * 100\% = 20\%$.

В результате разработки и внедрения ИС учета учебного времени нагрузка на специалиста снизилась на 20 процентов – один дополнительный выходной в неделю.

Индекс, показывающий уменьшение трудовых затрат или повышение производительности труда Y_T , можно получить, используя формулу (3).

$$Y_T = \frac{T_0}{T_1}, \quad (3)$$

Результат расчета: $Y_T = 160 / 128 = 1,25$.

Абсолютное снижение стоимостных затрат ΔC в рублях за год, может быть рассчитано по формуле (4).

$$\Delta C = C_0 - C_1, \quad (4)$$

где C_0 – стоимостные затраты, рассчитанные в рублях за год при использовании текущего решения;

C_1 – стоимостные затраты, рассчитанные в рублях за год при использовании нового решения;

Получаем:

$$\Delta C = 160 * 143,75 - 128 * 143,75 = 23\ 000 - 18\ 400 = 4\ 600 \text{ р.}$$

Здесь 143,75 – это среднее количество рублей, которые получает сотрудник за час работы.

Также необходимо рассчитать срок окупаемости затрат на внедрение ИС учета учебного времени (T_{OK}). Используем формулу (5).

$$T_{OK} = K_{\Pi} / \Delta C, \quad (5)$$

где K_{Π} – 180 000 рублей затраты на создание проекта (проектирование и внедрение) из таблицы 10 – Расходы на разработку ИС.

$$T_{OK} = 180\ 000 / 4\ 600 = 39,13$$

Таким образом срок окупаемости проекта составляет около 39 месяцев.

Не все преимущества автоматизации и улучшения процессов могут быть измерены с помощью экономических показателей. После внедрения средств автоматизации, становится возможным более дешевое и быстрое хранение и поиск информации для сотрудников.

Таким образом, помимо экономических выгод, автоматизация и связанные с ней улучшения процессов могут приносить значительные

преимущества, которые не всегда могут быть оценены в денежном выражении. Это позволяет организации не только снизить затраты, но и улучшить свою конкурентоспособность и общую эффективность.

Выводы по главе 3

В третьей главе были рассмотрены архитектурные подходы к проектированию программных систем. Произведено сравнение и выбор наиболее подходящей архитектуры для разработки системы, исследуемой в работе. Принято решение остановиться на архитектуре MVC.

Были рассмотрены популярны реляционные базы данных, а также проведен сравнительный анализ, по результатам которого было принято решение использовать базу данных MySQL для разработки.

Также был проведен обзор популярных платформ, которые при разработке используют архитектурный подход MVC. Проведено сравнение платформ и фреймворков и принято решение разрабатывать информационную систему на языке программирования Python с использованием фреймворка Django.

По окончании всех подготовительных этапов мы приступили к разработке информационной системы. Был определен порядок этапов разработки информационной системы.

После разработки проведено ручное тестирование функционала информационной системы, во время которого были выявлены и устранены недочеты, связанные с отображением данных.

Помимо всего вышперечисленного, нами был описан функционал разработанной информационной системы, а также произведен расчет экономической эффективности внедрения системы. Определив стоимость разработки системы нам удалось выявить выгоды от использования информационной системы и срок окупаемости разработанной информационной системы учета учебного времени.

Заключение

В рамках выполнения выпускной квалификационной работы было произведено исследование процесса составления расписания занятий в техникуме, а также разработана и реализована информационная система для автоматизации этого процесса.

Такое программное обеспечение позволяет существенно сократить время и усилия, затрачиваемые на составление расписания вручную, а также избежать ошибок и несоответствий в расписании.

В первой главе был произведен анализ деятельности предприятия и определены преимущества концептуального моделирования с помощью IDEF0. Созданная функциональная модель показала необходимость внедрения ИС в деятельность предприятия. Был произведен анализ существующих систем составления учебного расписания, выявлены их недостатки и определены требования к готовой системе.

Во второй главе была выбрана технология моделирования UML для создания диаграммы вариантов использования и диаграммы классов, а также IDEF1X для построения логической модели данных. Также было выполнено концептуальное и логическое моделирование системы, которое помогло визуализировать основные компоненты системы и их взаимосвязь. На основе этих моделей совместно с сотрудниками техникума был разработан макет пользовательского интерфейса.

В третьей главе работы были рассмотрены и проанализированы различные архитектурные подходы к проектированию программных систем, а также популярные реляционные базы данных и платформы, использующие архитектурный подход MVC. После проведенного исследования было принято решение разрабатывать информационную систему на языке программирования Python с использованием фреймворка Django и реляционной базы данных MySQL.

По окончании подготовительных этапов была разработана информационная система. Проведено ручное тестирование и устранены недочеты. Описан функционал и выявлена экономическая эффективность и выгода от использования программного продукта.

Данная информационная система обладает практической значимостью, так как позволяет облегчить учет учебного времени, повысить эффективность планирования учебного процесса и контроля за его выполнением. Более того, данная система может быть использована в других образовательных учреждениях после доработки и адаптации к конкретным потребностям и специфике организации.

Таким образом, проделанная работа была выполнена в полном объеме, все поставленные задачи успешно выполнены, цель достигнута.

Список используемой литературы и используемых источников

1. Архитектура приложений [Электронный ресурс]: URL: https://head.ru/blog/arkhitektura-prilozheniy/#Почему_важна_архитектура_веб-приложений (дата обращения: 18.04.2023).
2. Бабич А.В. Введение UML. Учебное пособие. Национальный Открытый Университет «ИНТУИТ»: Ай Пи Ар Медиа, 2022. 198 с.
3. Беляева И. В. Архитектура информационных систем. УлГТУ, 2019. 192 с.
4. Введение в СУБД MySQL: учебное пособие. Москва: Интернет-Университет Информационных Технологий (ИНТУИТ). Ай Пи Ар Медиа, 2021. 228 с.
5. ГОСТ Р ИСО 9241-210-2012 Эргономика взаимодействия человек-система. Часть 210. Человеко-ориентированное проектирование интерактивных систем. URL: <https://docs.cntd.ru/document/1200097755> (дата обращения: 05.04.2023).
6. Дерябкин В.П., Козлов В.В. Проектирование информационных систем по методологии UML с использованием QT-Технологии программирования: учебное пособие. АСИ СамГТУ, 2017. 156 с.
7. Дронов В.А. Django 3.0. Практика создания веб-сайтов на Python. ВHV-СПб, 2020. 704 с.
8. Зарплаты айтишников во втором полугодии 2022: +7% за счет регионов [Электронный ресурс] URL: <https://habr.com/ru/specials/714346/> (дата обращения: 21.04.2023).
9. Как работает архитектурный паттерн «модель-вид-контроллер» [Электронный ресурс] URL: <https://medium.com/nuances-of-programmingкак-работает-архитектурный-паттерн-модель-вид-контроллер-3d68fad6315e> (дата обращения: 18.04.2023).
10. Кириченко А. В., Хрусталева А. А. HTML5+CSS3. Основы современного WEB-дизайна. Наука и техника, 2019 г. 352 с.

11. Краткий путеводитель по методологиям и нотациям описания и моделирования бизнес-процессов. Часть 1 [Электронный ресурс] URL: <https://infostart.ru/1c/articles/1426878/> (дата обращения: 07.04.2023).
12. Кулаков К.А., Димитров В.М. Архитектура и фреймворки веб-приложений: учебное электронное пособие. Петр ГУ, 2020. 58 с.
13. Постолиит А. В. Python, Django и PyCharm для начинающих. БХВ, 2021 г. 464 с.
14. Простое руководство по UML-диаграммам и моделированию баз данных [Электронный ресурс] URL: <https://www.microsoft.com/ru-ru/microsoft-365/business-insights-ideas/resources/guide-to-uml-diagramming-and-database-modeling> (дата обращения: 05.04.2023).
15. Спицина И.А., Аксёнов К.А. Разработка информационных систем. Пользовательский интерфейс: учебное пособие для СПО. Профобразование УрФУ, 2020. 98 с.
16. Antonio Mele. Django 3 By Example: Build powerful and reliable Python web applications from scratch. Packt Publishing, 2020. 716 p.
17. Django documentation [Электронный ресурс] URL: <https://docs.djangoproject.com/en/4.2/> (дата обращения: 15.04.2023).
18. Figma — простое решение для дизайнера, сложное решение для верстальщика [Электронный ресурс] URL: <https://habr.com/ru/post/463181> (дата обращения: 10.04.2023).
19. Moritz Lenz. Python. Continuous Integration and Delivery. Apress, 2019 г. – 168 с.
20. Ralph Stair, George Reynolds. Principles of Information Systems. Cengage Learning, 2018. 656 p.
21. Sherry L. Mee Bell. Data Modeling and Database Design for Information Systems: A Practical Guide Using Entity-Relationship Model. Wiley, 2011. 408 p.
22. William S. Vincent. Django for Beginners: Build websites with Python and Django. Independently published, 2020. 309 p.