

АННОТАЦИЯ

Выпускная квалификационная работа Цызырова Максима Сергеевича по теме «Разработка электронного журнала регистрации и учета бюллетеней АО «УКБП»».

Пояснительная записка: страниц 59, рисунков 20, таблиц 26.

Графическая часть: 3 листа ф. А1.

БАЗА ДАННЫХ, ВЕБ РАЗРАБОТКА, ЖУРНАЛ, БЮЛЛЕТЕНЬ, ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, Laravel, PHP, MySQL.

Система предназначена для регистрации и учета заданий, которые выдаются на предприятии АО «УКБП».

Основа этой системы лежит в простых концепциях, что позволяет автоматизировать процесс заполнения бюллетеней в журнале регистрации, исключая из производственных бизнес-процессов их бумажное представление, а так же упрощает поиск бюллетеней в дальнейшем.

Рабочие на предприятии ежедневно получают задания от заказчиков, которые необходимо выполнить. При этом данные заказы ещё и необходимо контролировать, например, за статусом их выполнения. В бумажном виде журнала регистрации и учета бюллетеней это было бы сделать проблематичнее, чем в электронном его аналоге.

Внедрение данной системы позволяет систематизировать имеющуюся на предприятии базу данных заказов, автоматизировать процесс внесения данных в технические бюллетени электронного журнала, а также сократить время их поиска.

Система полностью прошла диагностику различными механизмами тестирования. Система работает с отсутствием ошибок.

СОДЕРЖАНИЕ

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ	8
ВВЕДЕНИЕ	9
1. ТЕХНИЧЕСКОЕ ЗАДАНИЕ	10
1.1. Назначение и цели создания системы.....	10
1.1.1 Назначения системы	10
1.1.2. Цели создания	10
1.2. Характеристика объекта автоматизации	10
1.2.1. Общее описание	10
1.2.2. Состав и порядок функционирования	11
1.3. Общие требования к системе.....	17
1.3.1. Требования к структуре и функционированию системы	17
1.3.1.1 Режимы функционирования системы.....	19
1.3.2 Дополнительные требования	20
1.3.2.1. Требования к способам и средствам связи для информационного обмена между компонентами модуля информационной системы	20
1.3.2.2. Требования к способу связи для информационного обмена между компонентами системы, в которую входит модуль информационной системы.....	20
1.3.2.3. Требования к надёжности	21
1.3.2.4. Требования к безопасности	21
1.3.2.5. Требования к эргономике и технической эстетике	22
1.3.2.6. Требования к защите информации от несанкционированного доступа.....	23
1.4. Требования к функциям, выполняемым системой.....	23
1.4.1. Функция «Вход в приложение».....	24
1.4.2. Функция «Поиск бюллетеней»	25
1.5. Требования к видам обеспечения.....	26
1.5.1. Требования к информационному обеспечению	26

1.5.2.	Требования к алгоритмическому обеспечению.....	26
1.5.3.	Требования к программному обеспечению	26
1.5.4.	Требования к математическому обеспечению.....	27
1.5.5.	Требования к лингвистическому обеспечению	27
1.5.6.	Требования к информационной безопасности.....	27
2.	ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ	28
2.1.	Выбор средств управления данными	28
2.2.	Проектирование базы данных.....	28
2.2.1.	Концептуальная схема базы данных.....	29
2.2.2.	Логическая схема базы данных	31
2.2.3.	Физическая схема базы данных.....	32
2.1.	Проектирование файлов данных	34
2.2.	Организация сбора, передачи, обработки и выдачи информации	34
3.	АЛГОРИТМИЧЕСКОЕ ОБЕСПЕЧЕНИЕ СИСТЕМЫ	35
4.	ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ СИСТЕМЫ	37
4.1.	Структура программного обеспечения и функции его компонентовДля работы МИС необходимо следующее ПО:	37
4.2.	Выбор компонентов программного обеспечения	37
4.2.1.	Операционная система	37
4.2.2.	Инструментальное средство разработки и язык программирования	37
4.2.3.	Вспомогательное программное обеспечениеВспомогательное ПО не требуется.....	39
4.3.	Разработка прикладного программного обеспечения	39
4.4.	Особенности реализации, эксплуатации и сопровождения системы ..	42
4.5.	Руководство пользователя.....	43
4.5.1.	Требования к условиям эксплуатации	43
4.5.2.	Инсталляция и настройка.....	43
4.5.3.	Порядок и особенности работы.....	43
4.5.4.	Исключительные ситуации и их обработка	46
5.	ТЕСТИРОВАНИЕ	47

5.1. Модульное и интеграционное тестирование	47
5.1.1. Условия и порядок тестирования.....	47
5.1.2. Исходные данные для контрольных примеров	48
5.2. Функциональное тестирование	49
5.2.1. Условия и порядок тестирования.....	49
5.2.2. Исходные данные для контрольных примеров	49
5.3. Нагрузочное тестирование.....	51
5.3.1. Условия и порядок тестирования.....	51
5.3.2. Исходные данные для контрольных примеров	51
5.4. Тестирование интерфейсов.....	51
5.5. Результаты тестирования	51
6. ЭКОНОМИЧЕСКИЙ РАЗДЕЛ.....	53
6.1. Экономический анализ внедрения системы	53
6.2. Оценка трудозатрат на разработку модуля информационной системы ...	53
6.3. Расчёт трудовых и стоимостных показателей	55
6.4. Расчёт окупаемости МИС	56
6.5. Выводы экономического раздела.....	57
ЗАКЛЮЧЕНИЕ	58
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	59

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ

АО – акционерное общество.

БД – база данных.

ВР – выделенные ресурсы.

ИР – информационные ресурсы.

ИС – информационная система.

ОС – операционная система.

ПО – программное обеспечение.

СЗИ – средства защиты информации.

ТЗ – техническое задание.

ЦАПЖЦИ – центр автоматизации процессов жизненного цикла изделия

ВВЕДЕНИЕ

На сегодняшний день на каждом крупном предприятии имеется острая необходимость в систематизации учета заказов, которые поступают практически на ежедневной основе. Для решения этой проблемы руководители применяют различные способы их учета. Одним из самых распространенных способов является структуризация заказов в виде списка записей в определенный журнал.

Подобный метод работы позволяет вести учет о количестве заказов, об отделе (структурном подразделении) в котором будет выполняться данный заказ и о его руководителе. Также такой подход позволяет заархивировать записи на случай, если к ним придется обратиться в будущем.

Данный процесс осуществлялся вручную и сопровождался заполнением бумажного журнала регистрации и учета бюллетеней, что приносило неудобства для работы специализированных на этом работников, так как, например, при заполнении бумажного журнала они могли сделать опечатку. Так же заполнение бюллетеней занимало достаточно много времени. Именно поэтому было решено разработать данную систему

Электронный журнал регистрации и учета бюллетеней позволяет вносить данные заказа напрямую из базы данных предприятия. Система позволяет автоматизировать процесс регистрации и учета, позволяет редактировать статус выполнения заказа, а так же упрощает поиск заказов, внесенных в журнал в виде бюллетени. Соответственно, благодаря этому сокращается время работы с документной частью заказов.

Целью данной ВКР является разработка системы, в которую вносятся техническая документация заказов для дальнейшего учета.

Следовательно, в результате разработки и внедрения данной системы может быть повышение дохода предприятия и более удобный учет заказов.

1. ТЕХНИЧЕСКОЕ ЗАДАНИЕ

1.1. Назначение и цели создания системы

1.1.1. Назначения системы

Web-приложение электронного журнала регистрации и учёта бюллетеней предназначено для автоматизации, безопасности, и хранения бюллетеней АО «УКБП». Бюллетень – поле, которое хранит в себе следующую информацию: название заказа, техническое задание заказа, статус выполнения заказа, цех производства, за которым закреплено изготовление заказа, начальник цеха.

1.1.2. Цели создания

Электронный журнал регистрации и учета бюллетеней создается с целью автоматизации процесса внесения данных заказа, дальнейшего их учета и поиска, а так же снижения экономических, временных и трудовых затрат.

Для реализации поставленных целей модуль информационной системы должен решать следующие задачи:

- сокращение время затрат на внесение данных;
- обеспечение более легкого поиска данных;
- сохранность данных;
- изменение статуса работы над заказом.

1.2. Характеристика объекта автоматизации

1.2.1. Общее описание

Электронный журнал регистрации и учета бюллетеней – система, предназначенная для автоматизации внесения, контроля и учета данных

технической документации заказов. Так же данной системе учитывается статус работы заказа

Основная цель разработки электронного журнала регистрации и учета бюллетеней - упростить и упорядочить работу отдела документооборота, ускорить работу с документацией, в следствие чего повысить качество работы . С помощью данной системы, предприятие так же сможет снизить количество ошибок, обусловленных человеческим фактором, из-за уменьшения ручной работы.

Электронный журнал регистрации и учета бюллетеней позволяет хранить данные о заказе, что значительно упрощает их поиск. Например, в бумажном варианте такого журнала для того, чтобы найти необходимый бюллетень, содержащий информацию о заказе пришлось бы поднимать архивы.

В общем и целом журнал регистрации и учета бюллетеней является важной системой для предприятия, которая хочет повысить эффективность работы и оптимизировать функцию учета документации о заказе.

1.2.2. Состав и порядок функционирования

Объектом автоматизации системы является процесс внесения данных в журнал регистрации, также выполнения процессов, указанных ниже:

- предоставление информации;
- предоставление доступа к редактированию статуса выполнения бюллетеня;
- обеспечение защиты данных;
- хранение данных о задании.

Функциональный состав системы:

- доступ к базе данных предприятия;
- загрузка файлов;
- сохранение данных.

Система обязательно должна иметь в своем составе следующие

КОМПОНЕНТЫ:

- программный пакет Laravel;
- программный пакет PHP;
- программный пакет MySQL;
- программный пакет Composer.

Бизнес-процессы до внедрения МИС представлены на диаграмме в нотации IDEF0 (Рисунок 1.1 - 1.3). Вербальное описание бизнес-процессов приведено в таблице 1.1.

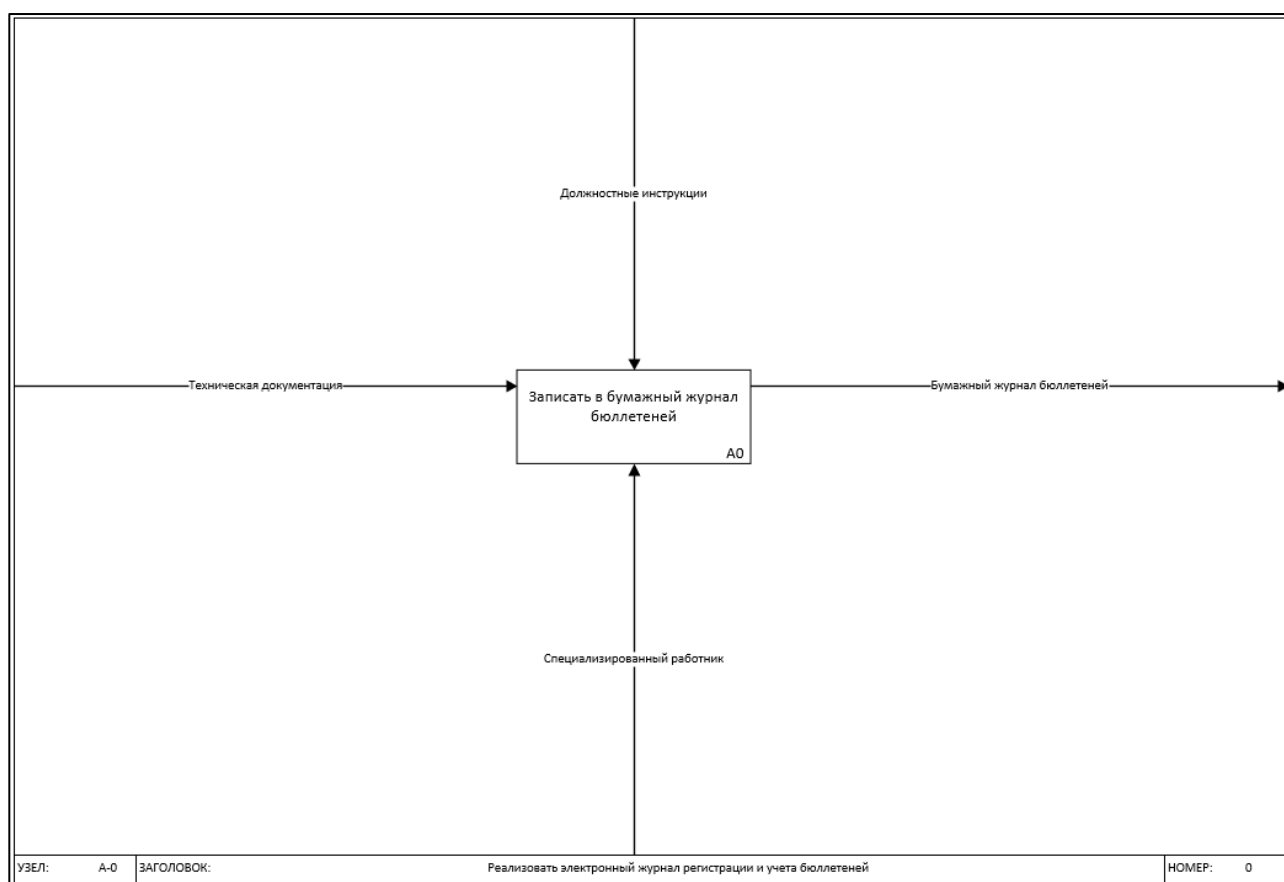


Рис. 1.1. IDEF0 до внедрения программы, узел A-0

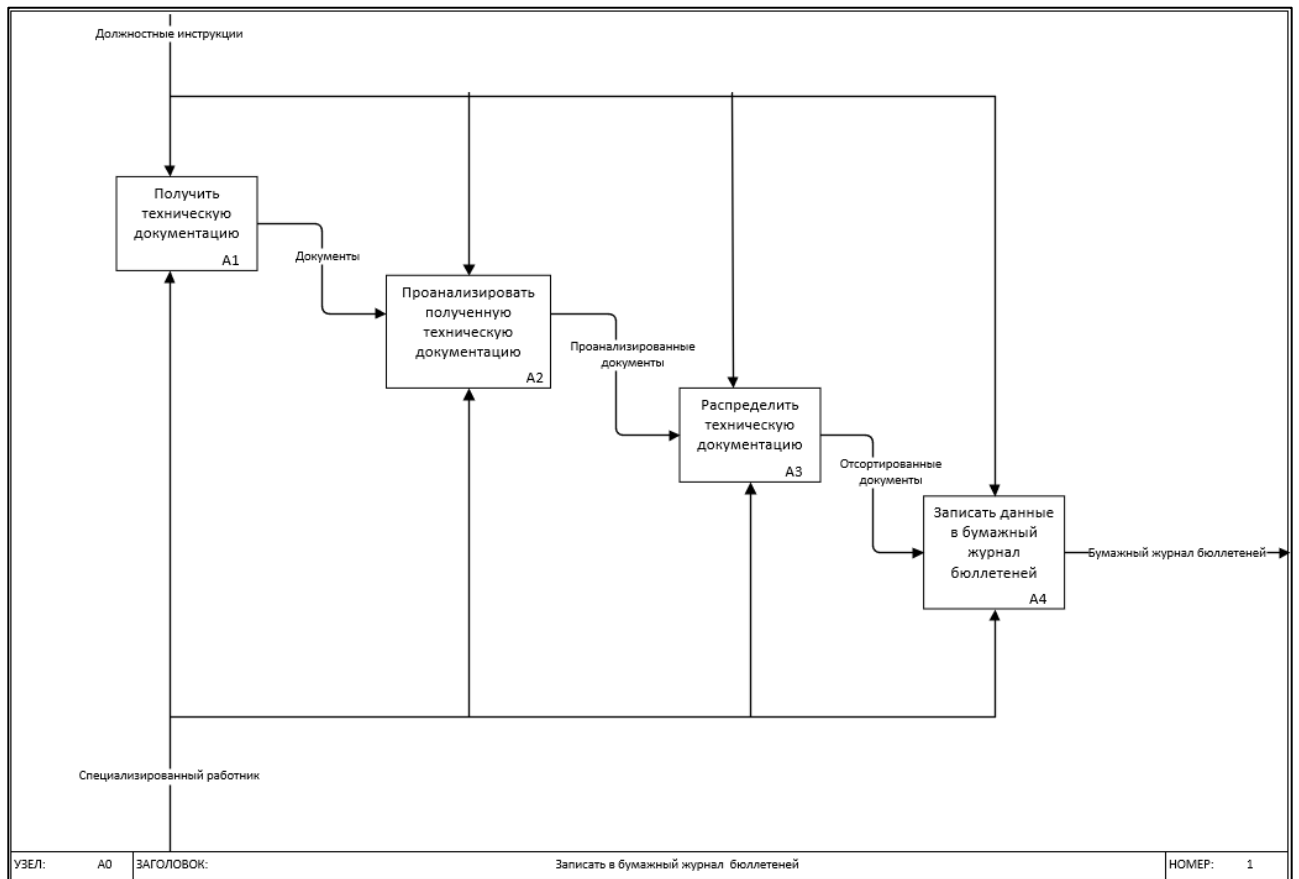


Рис. 1.2. IDEF0 до внедрения программы, узел A0

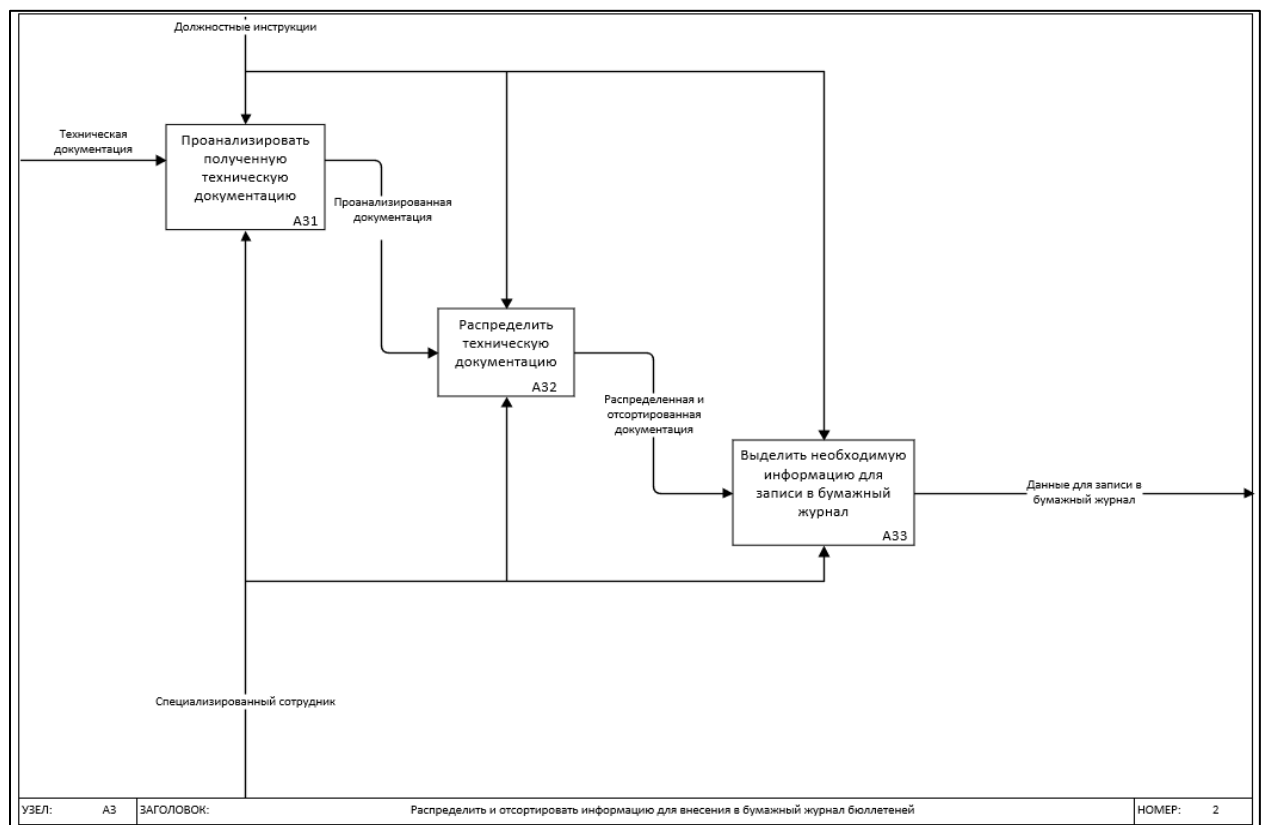


Рис. 1.3. IDEF0 до внедрения программы, узел A3

Вербальное описание бизнес-процессов до внедрения МИС

№ блока	Наименование блока	Описание блока
A1	Получить техническую документацию	Специализированный работник получает бумажный вариант технической документации
A2	Проанализировать полученную техническую документацию	Специализированный работник самостоятельно анализирует техническую документацию
A3	Распределить техническую документацию	Специализированный работник распределяет необходимую документацию
A4	Записать данные в бумажный журнал бюллетеней	Специализированный работник вручную записывает данные в бумажный журнал бюллетеней

Все вышеописанные бизнес-процессы выполняются вручную и являются неавтоматизированными. Бизнес-процессы A2, A3, A4 должны быть автоматизированы.

Объектами автоматизации являются процессы автоматического формирования журнала бюллетеней.

Процессы формирования журнала бюллетеней включают в себя:

- заполнение программных форм;
- использование СУБД MySQL;
- обработка данных;
- сохранение данных.

Данные процессы осуществляются следующими специалистами:

- администратор;
- специализированный работник.

Бизнес-процессы после внедрения программного продукта электронного журнала регистрации и учёта бюллетеней представлены на диаграмме в нотации IDEF0 (Рисунок 1.4 - 1.6). Вербальное описание бизнес-процессов приведено в таблице 1.2.

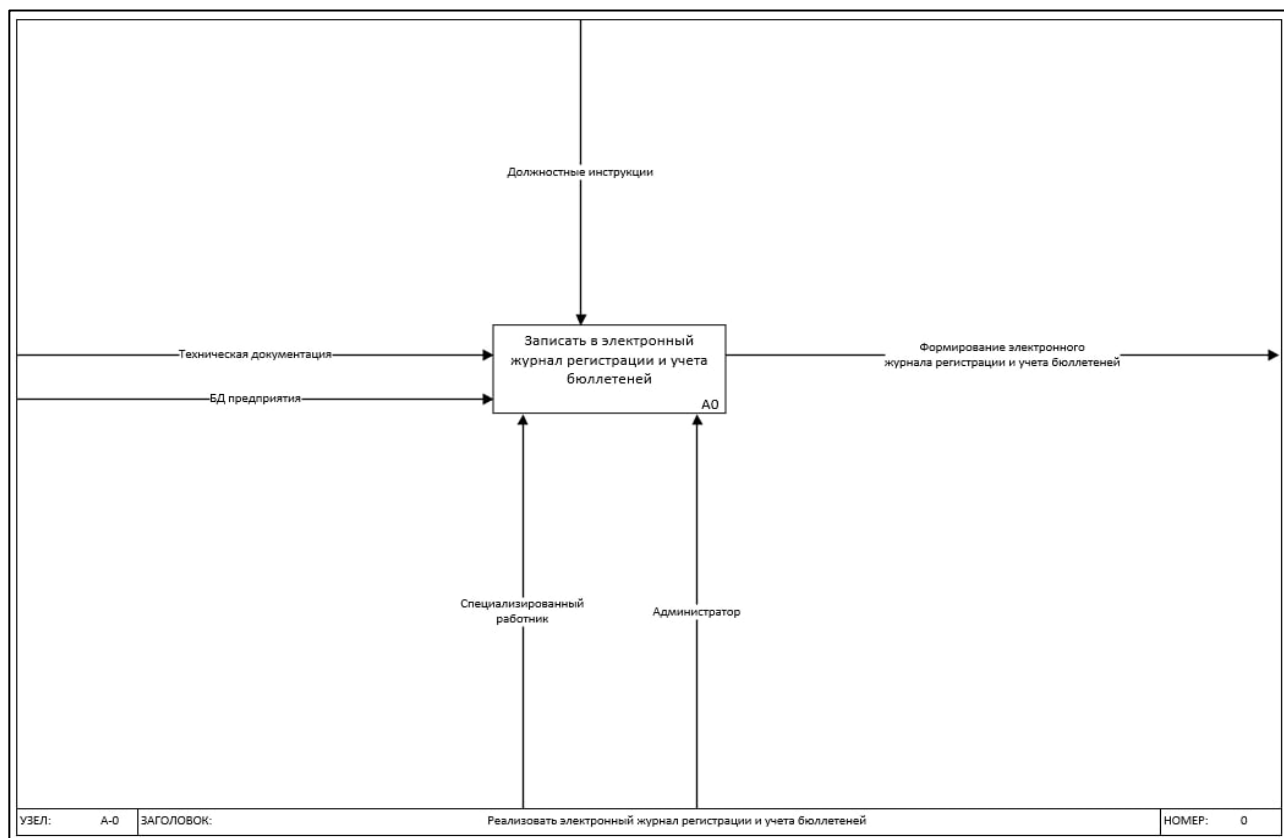


Рис. 1.4. IDEF0 после внедрения программы, узел A-0

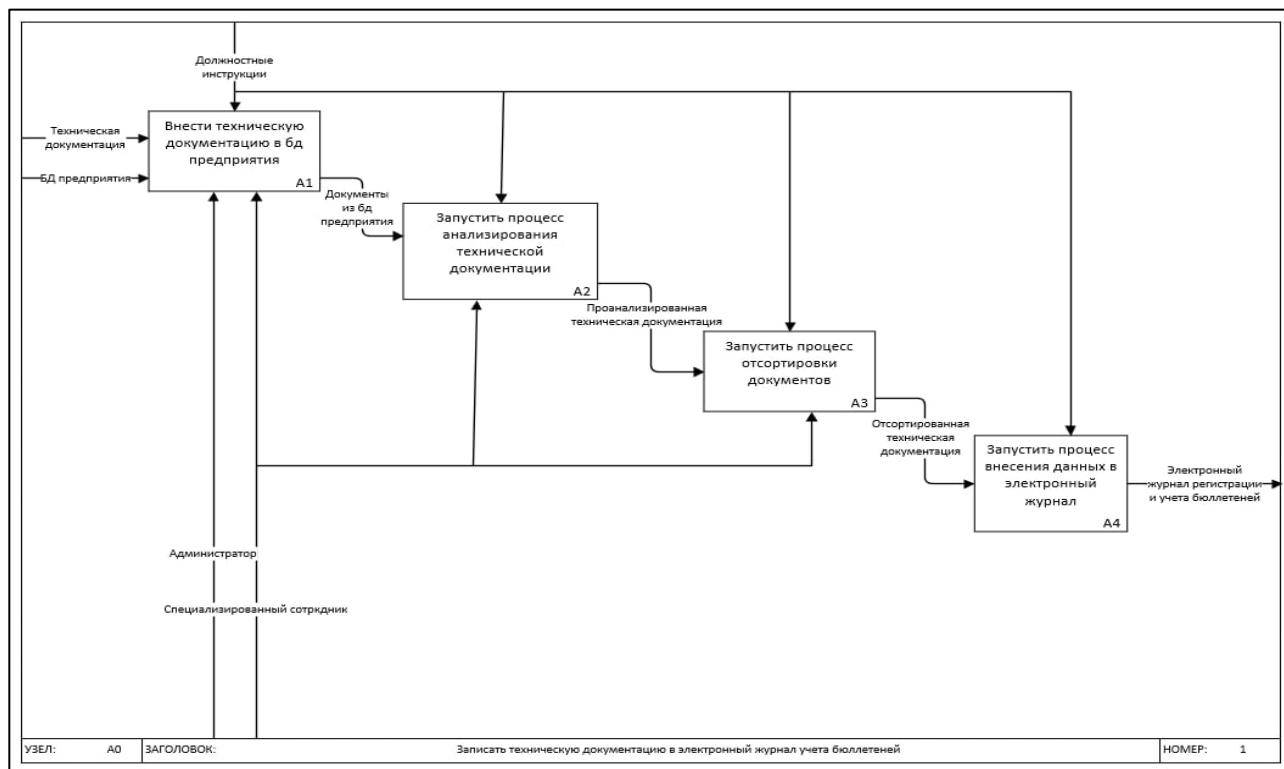


Рис. 1.5. IDEF0 после внедрения программы, узел A0

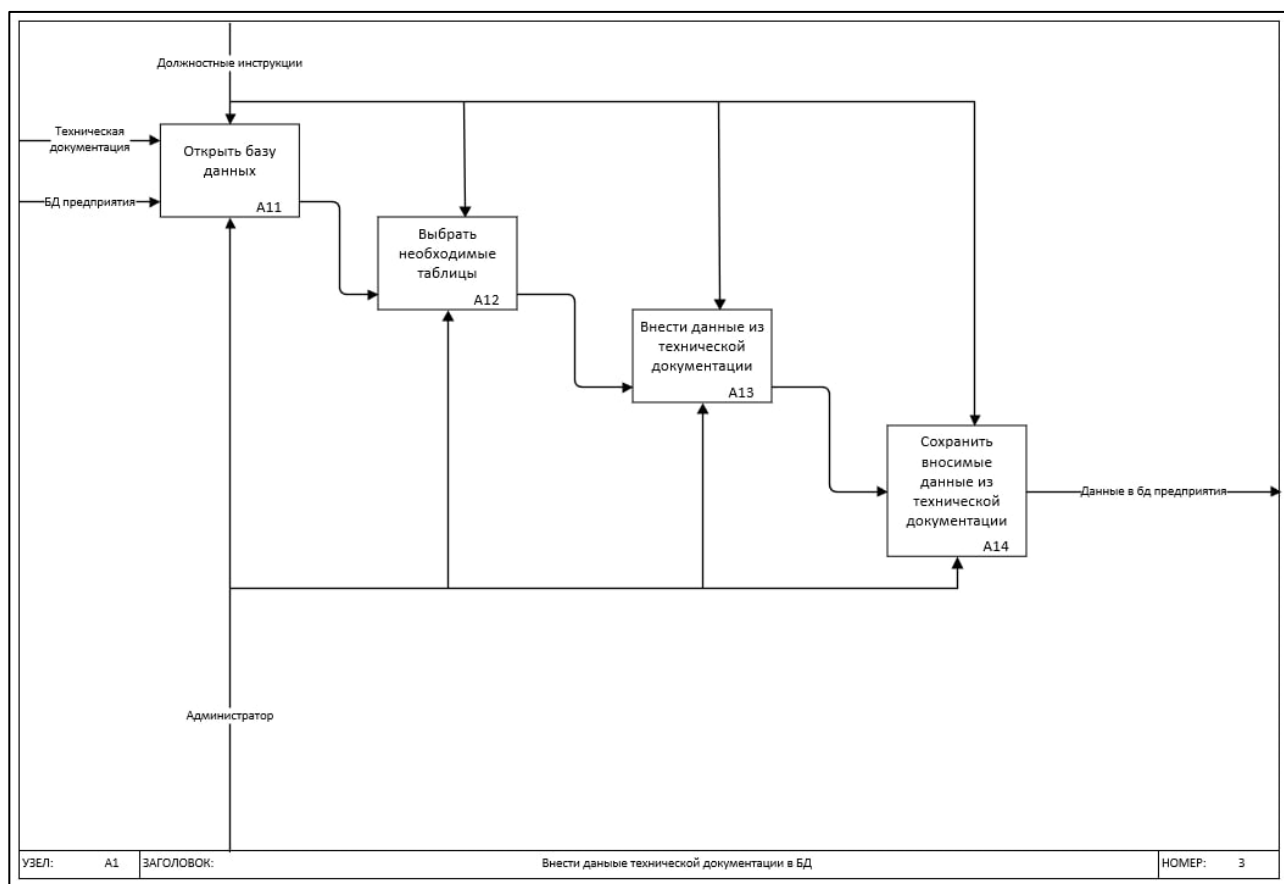


Рис. 1.6 IDEF0 после внедрения программы, узел A1

Таблица 1.2

Вербальное описание бизнес-процессов после внедрения программы

№ блока	Наименование блока	Описание блока
A1	Внести техническую документацию в бд предприятия	Администратор подключается к БД предприятия и вносит туда данные из технической документации
A2	Запустить процесс анализирования технической документации	Система анализирует данные, которые необходимо будет добавить в журнал
A3	Запустить процесс отсортировки документов	Система сортирует данные, которые необходимо внести в журнал
A4	Запустить процесс внесения данных в электронный журнал	Система вносит данные в журнал бюллетеней под руководством специализированного работника

1.3. Общие требования к системе

1.3.1. Требования к структуре и функционированию системы

Структура взаимодействия приложения логирования с администратором и программистом приведена на диаграмме вариантов использования Use Case (Рисунок 1.7).

Ниже приведены типичные ходы событий для каждого из агентов системы в таблицах 1.3, 1.4.

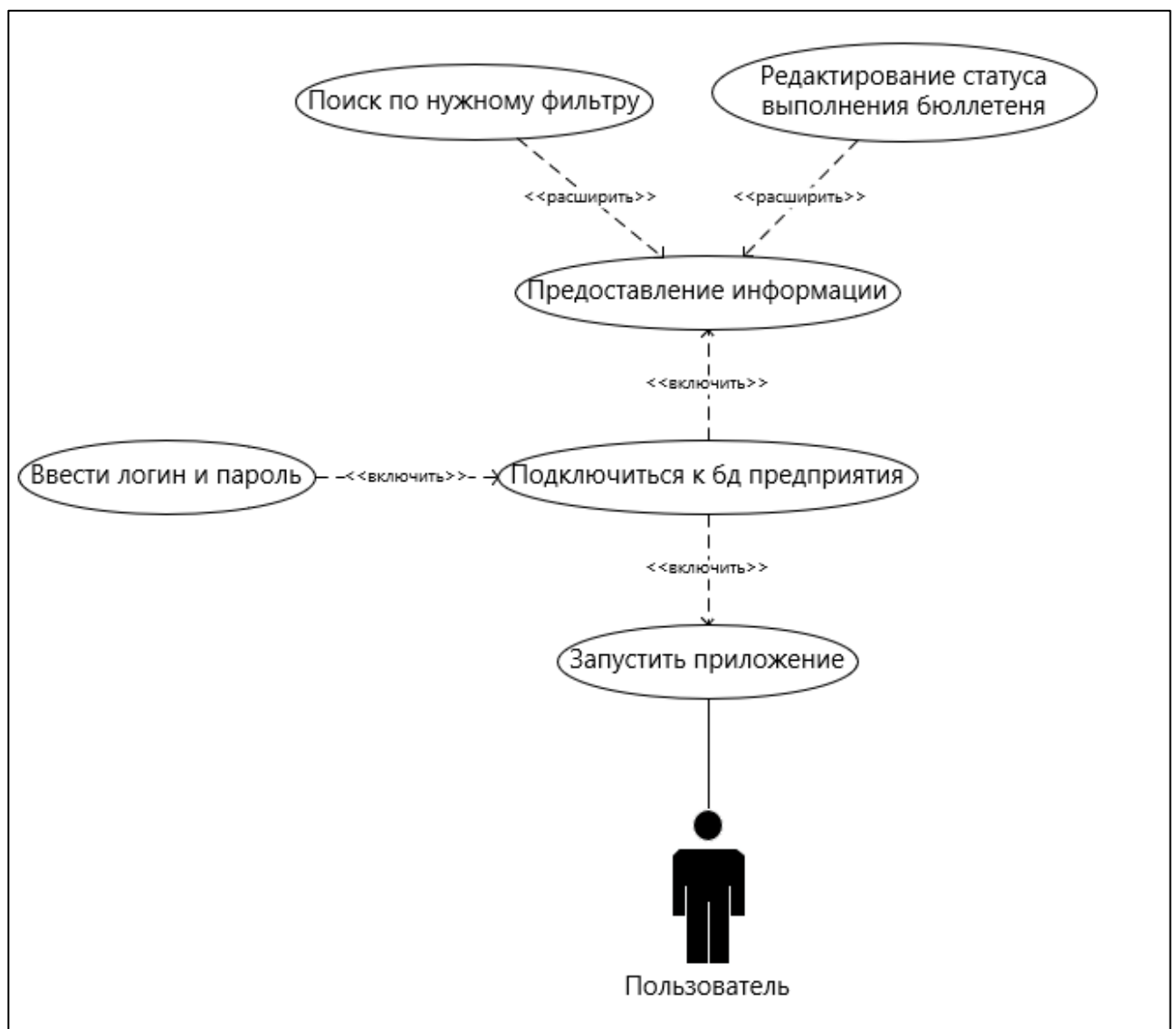


Рис. 1.7. Диаграмма Use Case, агент Пользователь

Типичный ход событий агента – Пользователя

Действия исполнителя	Отклик системы
1. Пользователь запускает приложение	2. Приложение запускается на ПК
3. Пользователь вводит имя пользователя и пароль	4. Система анализирует введенные данные, подключается к БД предприятия, после успешной проверки предоставляет доступ к журналу бюллетеней
5. Пользователь выбирает действие «Поиск по нужному фильтру»	6. Система открывает найденную информацию
7. Пользователь выбирает действие «Удаление ненужной информации»	8. Система удаляет ненужную информацию из бюллетеней
9. Программист выбирает действие «Редактирование информации»	10. Система вносит нужные правки, написанные пользователем

Дополнительная информация:

1. Если время выполнения программы для пользователя велико, то он прерывает процесс выполнения.
2. Необходимо обеспечить возможность выхода из варианта на любом этапе.

Таблица 1.4

Типичный ход событий – оповещение об ошибке

Действия исполнителя	Отклик системы
1. Пользователь пропускает поля для заполнения	2. Система отмечает эти поля и просит заполнить их
3. Пользователь заполняет поля	4. Система снимает с полей метки и продолжает работу
5. Пользователь вводит недопустимое/неверное значение	6. Система отмечает эти поля и просит заполнить их

Типичный ход событий для агента «Администратор» является аналогичным предыдущему.

1.3.1.1. Режимы функционирования системы

Для системы определены следующие режимы функционирования:

- нормальный режим функционирования;
- аварийный режим функционирования.

Основным режимом функционирования системы является нормальный режим. В нормальном режиме функционирования системы:

- клиентское программное обеспечение и технические средства пользователей и администратора системы обеспечивают возможность функционирования в любое время суток;
- серверное программное обеспечение и технические средства серверов обеспечивают возможность круглосуточного функционирования, с перерывами на обслуживание;
- исправно работает оборудование, составляющее комплекс технических средств;
- исправно функционирует системное, базовое и прикладное программное обеспечение системы.

Для обеспечения нормального режима функционирования системы необходимо выполнять требования и выдерживать условия эксплуатации программного обеспечения и комплекса технических средств системы, указанные в соответствующих технических документах.

Аварийный режим функционирования системы характеризуется отказом одного или нескольких компонент программного, технического обеспечения. В случае перехода в аварийный режим необходимо:

- завершить работу всех приложений, с сохранением данных;
- выполнить резервное копирование БД.

После этого необходимо выполнить комплекс мероприятий по устранению причины перехода системы в аварийный режим.

1.3.2. Дополнительные требования

1.3.2.1. Требования к способам и средствам связи для информационного обмена между компонентами модуля информационной системы

Для обмена информацией, входящих в состав системы, должна существовать локальная сеть с минимальной скоростью 1 гбит/с. У сервера, на котором развернута система, должен иметь 1 сетевую карту.

1.3.2.2. Требования к способу связи для информационного обмена между компонентами системы, в которую входит модуль информационной системы

Связь модулей информационных систем, входящих в состав системы, представлена на диаграмме пакетов (Рисунок 1.8).

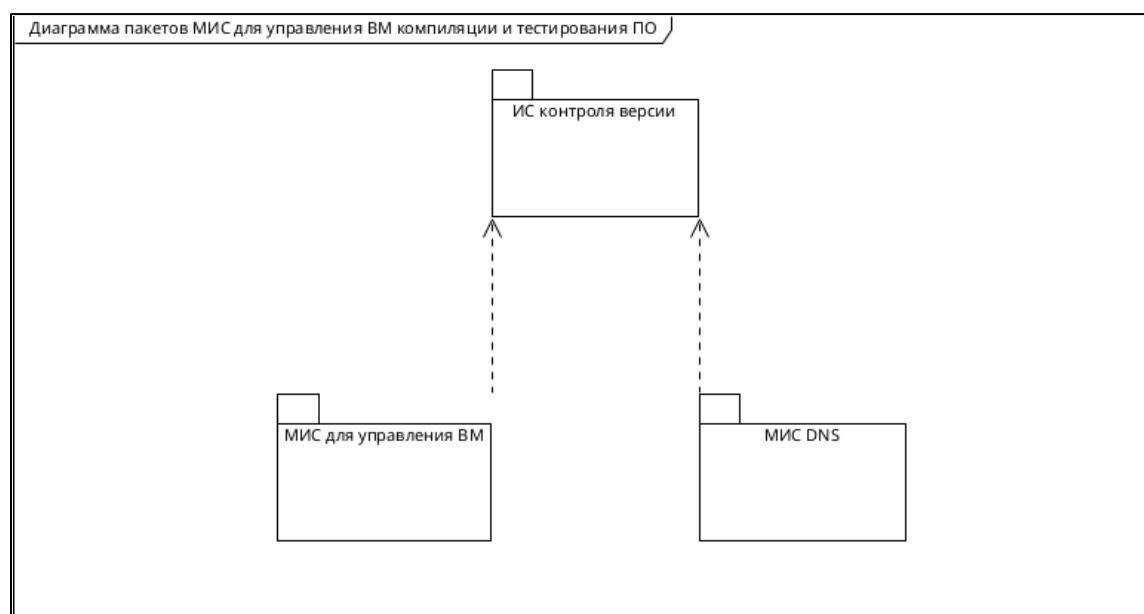


Рис. 1.8. Связь модулей ИС

1.3.2.3. Требования к надёжности

Система должна сохранять работоспособность и обеспечивать восстановление своих функций при возникновении следующих внештатных ситуаций:

- при сбоях в системе электроснабжения аппаратной части, приводящих к перезагрузке ОС, восстановление программы должно происходить после перезапуска ОС.
- при ошибках в работе аппаратных средств восстановление функции системы возлагается на ОС;
- при ошибках, связанных с программным обеспечением (ОС и драйверы устройств), восстановление работоспособности возлагается на ОС.

Для защиты аппаратуры от бросков напряжения и коммутационных помех должны применяться сетевые фильтры.

Время выделенное на восстановление серверов должна быть меньше суток в течение рабочего года.

1.3.2.4. Требования к безопасности

Система электропитания должна обеспечивать защитное отключение при перегрузках и коротких замыканиях в цепях нагрузки, а также аварийное ручное отключение по ГОСТ 12.2.007.0-75.

Общие требования пожарной безопасности должны соответствовать нормам на бытовое электрооборудование. В случае возгорания не должно выделяться ядовитых газов и дымов по СНиП 21-01-97. После снятия электропитания должно быть допустимо применение любых средств пожаротушения по ГОСТ 12.1.004-91.

Факторы, оказывающие вредные воздействия на здоровье со стороны всех элементов системы (в том числе инфракрасное, ультрафиолетовое, рентгеновское и электромагнитное излучения, вибрация, шум,

электростатические поля, ультразвук строчной частоты и т.д.), не должны превышать действующих норм.

1.3.2.5. Требования к эргономике и технической эстетике

Интерфейс должен быть рассчитан на преимущественное использование манипулятора типа «мышь», то есть управление системой должно осуществляться с помощью набора экранных меню, кнопок, значков и т. п. элементов. Клавиатурный режим ввода должен применяться главным образом при заполнении и/или редактировании текстовых и числовых полей экранных форм. Требуемая кодировка UTF-8.

Все надписи экранных форм, а также сообщения, выдаваемые пользователю (кроме системных сообщений), должны быть на русском языке.

Экранные формы должны проектироваться с учетом требований унификации:

- все экранные формы пользовательского интерфейса должны быть выполнены в едином графическом дизайне, с одинаковым расположением основных элементов управления и навигации;
- для обозначения сходных операций должны использоваться сходные графические значки, кнопки и другие управляющие (навигационные) элементы;
- внешнее поведение сходных элементов интерфейса должны реализовываться одинаково для однотипных элементов.

Система должна обеспечивать корректную обработку аварийных ситуаций, вызванных неверными действиями пользователей, неверным форматом или недопустимыми значениями входных данных. В указанных случаях система должна выдавать пользователю соответствующие сообщения, после чего возвращаться в рабочее состояние, предшествовавшее неверной команде или некорректному вводу данных.

1.3.2.6. Требования к защите информации от несанкционированного доступа

Компоненты подсистемы защиты от несанкционированного доступа должны обеспечивать:

- идентификацию пользователя по хэшу пароля MD5;
- проверку полномочий пользователя при работе с системой;
- разграничение доступа пользователей на базе прав пользователя по ролям на уровне задач и информационных массивов;
- «слепые» пароли;
- защиту от внедрения SQL-кода.

1.4. Требования к функциям, выполняемым системой

Программное обеспечение журнала регистрации и учёта бюллетеней должно выполнять следующие функции:

- Связность БД «УКБП» с БД журнала регистрации и учёта бюллетеней
- Просмотр информации в бюллетенях
- Поиск необходимых бюллетеней по фильтру

Вышеуказанные функции подробно разобраны в пунктах 4.4.1. – 4.4.3. и представлены на диаграммах вариантов использования в нотации Use Case с точки зрения пользователя. Вербальное описание и типичный ход событий на каждую из данных функций представлены в таблицах 5 – 7.

1.4.1. Функция «Вход в приложение»

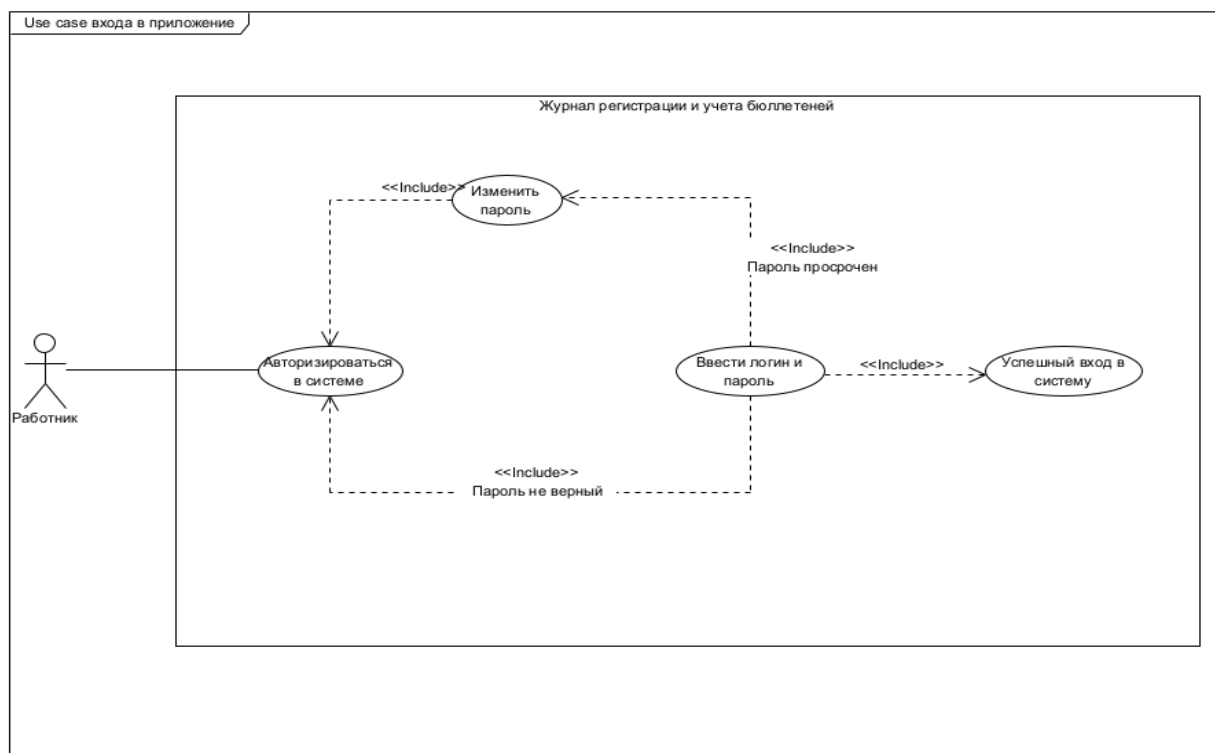


Рис. 1.9. Работник входит в приложение

Таблица 1.5

Типичный ход событий

Действия исполнителя	Отклик системы
1. Пользователь запускает приложение	2. Приложение запускается на ПК
3. Пользователь вводит имя пользователя и пароль и входит в систему	4. Система анализирует введенные данные, подключается к БД, после успешной проверки предоставляет доступ к приложению для дальнейшей работы
	4б. Система анализирует введенные данные, подключается к БД, после проверки сообщает, о неверных введенных данных и система предоставляет форму для повторного ввода данных
	4в. Система анализирует введенные данные, подключается к БД, после проверки сообщает, о просроченном пароле и система предоставляет форму для повторного ввода данных

1.4.2. Функция «Поиск бюллетеней»

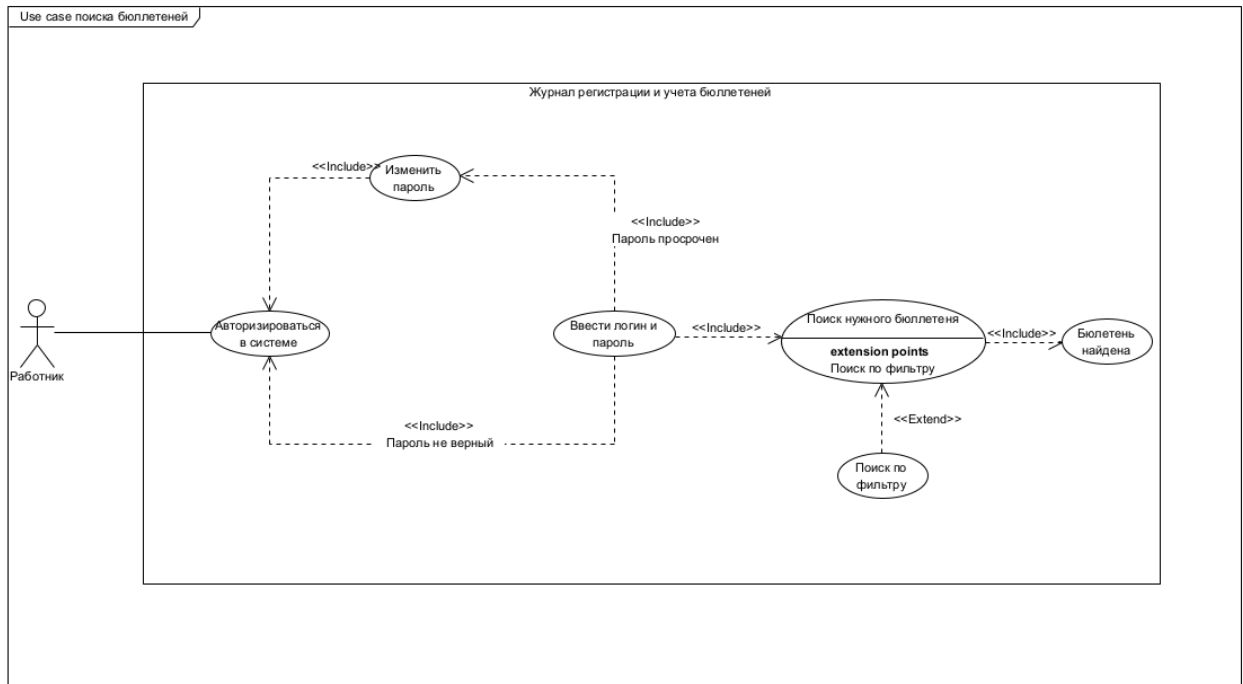


Рис. 1.10. Предоставление информации

Таблица 1.6

Типичный ход событий

Действия исполнителя	Отклик системы
1. Пользователь запускает приложение	2. Приложение запускается на ПК
3. Пользователь вводит имя пользователя и пароль и входит в систему	4. Система анализирует введенные данные, подключается к БД, после успешной проверки предоставляет доступ к приложению для дальнейшей работы
	4б. Система анализирует введенные данные, подключается к БД, после проверки сообщает, о неверных введенных данных и система предоставляет форму для повторного ввода данных
	4в. Система анализирует введенные данные, подключается к БД, после проверки сообщает, о просроченном пароле и система предоставляет форму для повторного ввода данных
5. Работник ищет нужную бюллетень	6. Приложение отображает все доступные бюллетени для их последующего выбора.
	6а. Приложение находит нужную бюллетень, исходя из фильтров, заданных пользователем системы.

1.5. Требования к видам обеспечения

1.5.1. Требования к информационному обеспечению

Уровень хранения данных в системе должен быть построен в СУБД PostgreSQL.

Структура базы данных должна быть организована рациональным способом, исключающим единовременную полную выгрузку информации, содержащейся в базе данных системы.

Технические средства, обеспечивающие хранение информации, должны использовать современные технологии, позволяющие обеспечить повышенную надежность хранения данных и оперативную замену оборудования.

В состав системы должна входить специализированная система резервного копирования и восстановления данных в соответствии с ГОСТ 54471-2011.

1.5.2. Требования к алгоритмическому обеспечению

Особые алгоритмические способы и функции отсутствуют.

1.5.3. Требования к программному обеспечению

Программное обеспечение подразделяется на клиентскую и серверную.

Клиентское ПО имеет следующие требования: PHP, Laravel, ОС Windows, разрядность системы x86/64.

Серверное ПО имеет следующие требования: сервер с СУБД MySQL. При разработке должен использоваться объектно-ориентированный подход программирования. Состав и содержание работ по созданию системы.

1.5.4. Требования к математическому обеспечению

Требования не предъявляются.

1.5.5. Требования к лингвистическому обеспечению

Все прикладное программное обеспечение системы для организации взаимодействия с пользователем должно использовать русский язык.

1.5.6. Требования к информационной безопасности

АО «УКБП» является закрытым предприятием, доступ к данному электронному журналу регистрации и учета бюллетеней может осуществляться людьми, работающими в подразделении ЦАПЖЦИ. Защита от несанкционированного доступа достигается за счёт системы индивидуальных пропусков предприятия.

Электронный журнал регистрации и учета бюллетеней может запускаться строго в режимном помещении, защищённый кодовым замком.

Требования защищённости представлены в таблице 1.7.

Таблица 1.7

Требования защищенности МИС

Требования к модулю	Выполнение требования
Управление доступом	Требование выполняется ИС
Идентификация, проверка подлинности и контроль доступа субъектов	Требование выполняется ИС
Использование аттестованных (сертифицированных) криптографических средств	Требование выполняется ИС
Обеспечение целостности	Требование выполняется ИС

2. ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ

2.1. Выбор средств управления данными

Для эффективной реализации разрабатываемого ПО крайне важно выбрать подходящую систему управления базами данных (СУБД), учитывая множество факторов и параметров. Критически важно, чтобы выбранная СУБД была масштабируемой, надежной, безопасной, обладала широким набором функциональных возможностей и удовлетворяла требованиям оперативной обработки информации. Кроме того, необходимо учитывать расходы на приобретение лицензии СУБД, установку и обучение персонала.

С учетом всех перечисленных требований, наиболее подходящей СУБД для реализации данного модуля информационной системы является MySQL.

Эта кроссплатформенная, реляционная сетевая СУБД свободно распространяется и имеет открытый исходный код. MySQL поддерживает стандартный язык SQL и обладает всем необходимым инструментарием для эффективной работы с базами данных. Она также обеспечивает высокий уровень безопасности паролей и учетных записей пользователей, а утилиты для проектирования таблиц имеют интуитивно понятный интерфейс. Благодаря реализации многопоточности, MySQL обеспечивает высокую скорость работы базы данных и легко масштабируется.

2.2. Проектирование базы данных

Для проектирования базы данных журнала регистрации и учёта бюллетеней было выбрано веб-приложение PHPMyAdmin. PHPMyAdmin-это свободное программное обеспечение с открытым исходным кодом для управления базами данных MySQL через веб-интерфейс. PHPMyAdmin является очень популярным инструментом в веб-разработке, так как он позволяет работать с базами данных без использования командной строки или

других специализированных средств управления базами данных. Он предоставляет простой и понятный интерфейс для работы с базами данных MySQL, что упрощает процесс разработки и управления базами данных.

2.2.1. Концептуальная схема базы данных

Концептуальное проектирование включает создание семантической модели предметной области, которая не зависит от конкретной СУБД или модели данных. Для построения этой модели используются формальные аппараты. В концептуальной модели базы данных описываются информационные объекты и связи между ними, а также устанавливаются ограничения целостности, которые определяют допустимые значения данных и связи между ними. В результате концептуального проектирования создается концептуальная схема базы данных. (Рисунок 2.1).

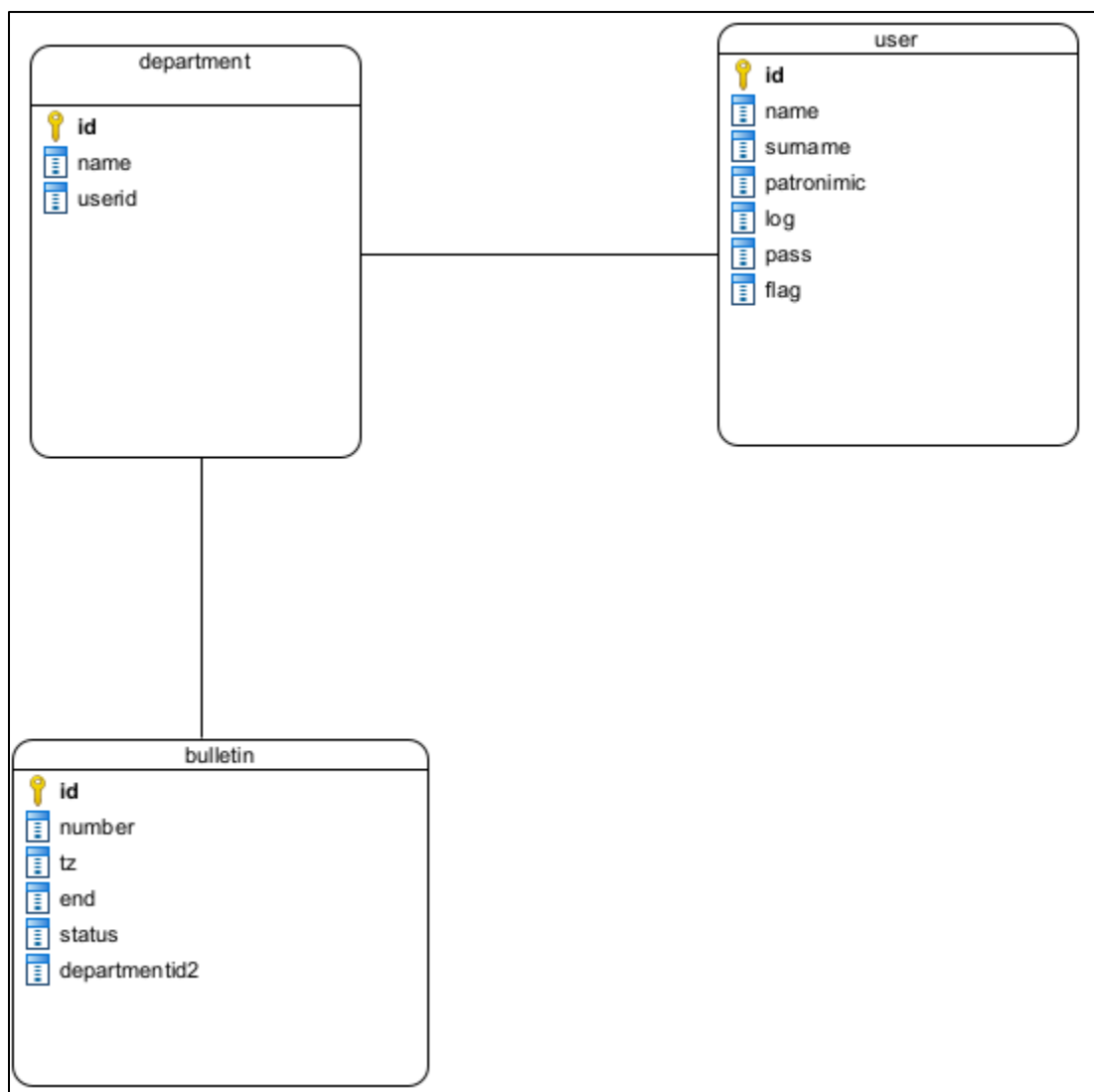


Рис. 2.1. Концептуальная схема БД

Описание реляционных таблиц концептуальной схемы представлены в таблице 2.1.

Таблица 2.1

Описание реляционных таблиц

№	Наименование поля	Описание
1	department	Сущность, которая хранит сведения об отделах и цехах.
2	bulletin	Сущность, которая собирает сведения в бюллетень об отделах, цехах, ТЗ, дате начала и сроках выполнения.
3	user	Сущность, хранящая сведения о пользователях системы.

2.2.2. Логическая схема базы данных

Логическая схема БД представлена на рисунке 2.2.



Рис.2.2. Логическая схема БД

Описание атрибутов сущностей БД представлена в Таблицах 2.2-2.4.

Таблица 2.2

Поля реляционной таблицы department

№	Наименование поля	Тип	Описание
1	id	Числовой	Идентификационный номер
2	name	Текстовый	Название цеха, отдела или лаборатории
3	userid	Числовой	Внешний ключ

Таблица 2.3

Поля реляционной таблицы user

№	Наименование поля	Тип	Описание
1	id	Числовой	Идентификационный номер
2	name	Текстовый	Имя пользователя
3	surname	Текстовый	Фамилия пользователя
4	patronimic	Текстовый	Отчество пользователя
5	log	Текстовый	Логин пользователя для входа в систему
6	pass	Текстовый	Пароль пользователя для входа в систему
7	flag	Числовой	Уровень доступа

Таблица 2.4

Поля реляционной таблицы bulletin

№	Наименование поля	Тип	Описание
1	id	Числовой	Идентификационный номер
2	number	Числовой	Номер бюллетени
3	tz	Массив двоичных данных	Файл с техническим заданием
4	end	Дата/Время	Дата окончания выполнения работы
5	status	Текстовый	Статус выполнения работы
6	departmentid2	Числовой	Внешний ключ

2.2.3. Физическая схема базы данных

На этом этапе сущности, описанные в концептуальной и логической моделях, заменяются реляционными таблицами, а их атрибуты - полями в таблицах. Физическая схема базы данных представлена на Рисунке 2.3.

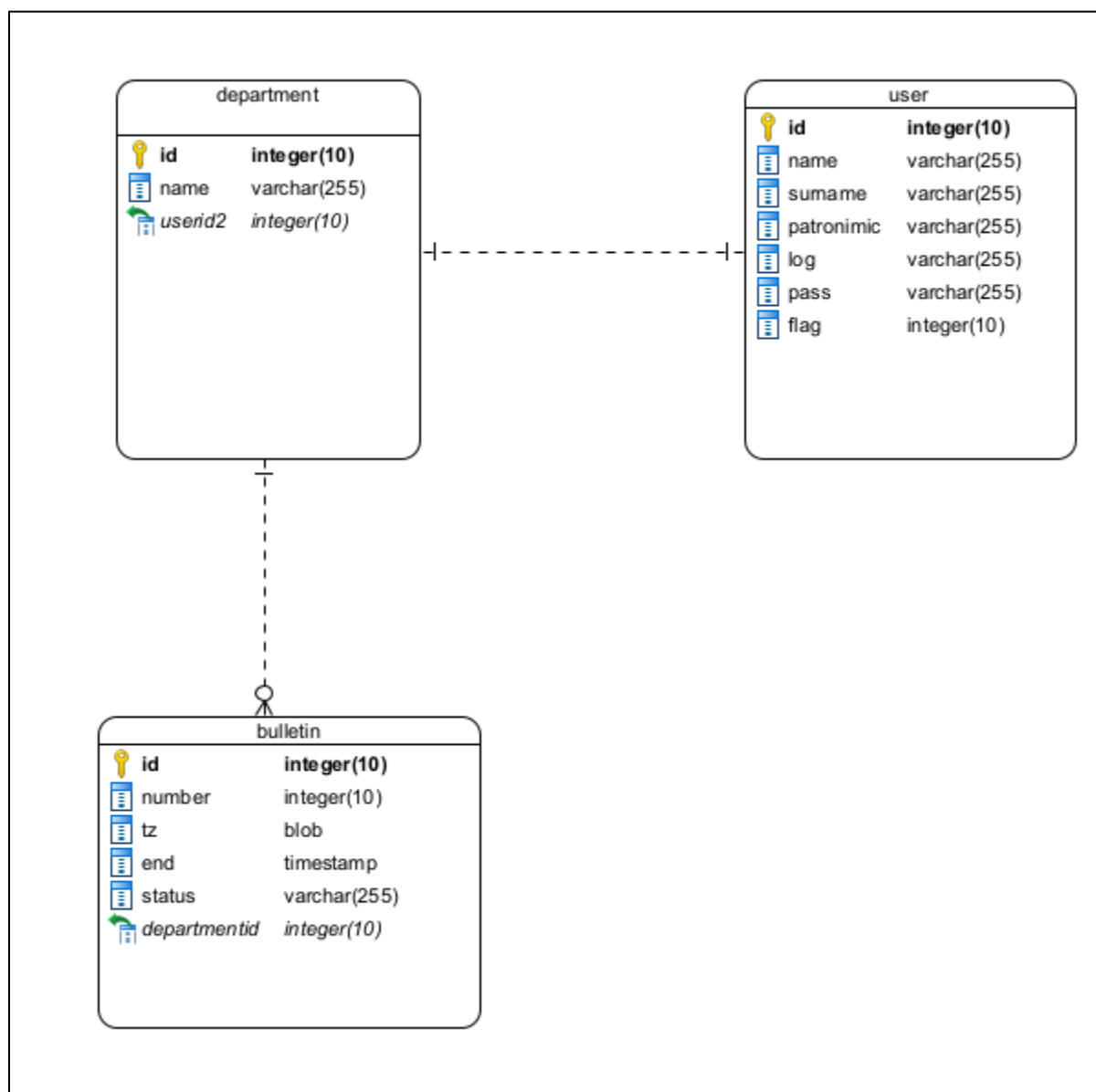


Рис. 2.3. Физическая схема БД

Описание атрибутов сущностей БД представлена в Таблицах 2.5-2.7.

Таблица 2.5

Поля реляционной таблицы department

№	Наименование поля	Тип	Значение по умолчанию
1	id	INTEGER (10)	AUTO_INCREMENT
2	name	VARCHAR (255)	NOT NULL
3	userid	INTEGER (210)	NOT NULL

Таблица 2.6

Поля реляционной таблицы user

№	Наименование поля	Тип	Значение по умолчанию
1	id	INTEGER (10)	AUTO_INCREMENT
2	name	VARCHAR (255)	NOT NULL
3	surname	VARCHAR (255)	NOT NULL
4	patronimic	VARCHAR (255)	NOT NULL
5	log	VARCHAR (255)	NOT NULL
6	pass	VARCHAR (255)	NOT NULL
7	flag	INTEGER (10)	NOT NULL

Таблица 2.7

Поля реляционной таблицы bulletin

№	Наименование поля	Тип	Значение по умолчанию
1	id	INTEGER (10)	AUTO_INCREMENT
2	number	INTEGER (10)	NOT NULL
3	discription	VARCHAR (255)	NULL
4	tz	BLOB	NOT NULL
5	begin	TIMESTAMP	NOT NULL
6	end	TIMESTAMP	NOT NULL
7	status	VARCHAR (255)	NOT NULL
8	departmentid2	INTEGER (10)	NOT NULL

2.1. Проектирование файлов данных

Данные из бюллетеней хранятся на сервере. Наименование файлов отражает их содержание. Передача данных на сервер реализуется при подключении сети Интернет. Доступ к ним обеспечивается посредством приложения phpMyAdmin.

2.2. Организация сбора, передачи, обработки и выдачу информации

Сбор информации происходит через программную форму. Пользователь заполняет поля формы, после чего данные обрабатываются передаются в хранилище данных на базе phpMyAdmin.

3. АЛГОРИТМИЧЕСКОЕ ОБЕСПЕЧЕНИЕ СИСТЕМЫ

Особые алгоритмические способы и функции отсутствуют. Общий алгоритм работы журнала регистрации и учёта бюллетеней представлен на блок-схеме (Рисунок 3.1), а так же на диаграмме последовательностей (Рисунок 3.2).

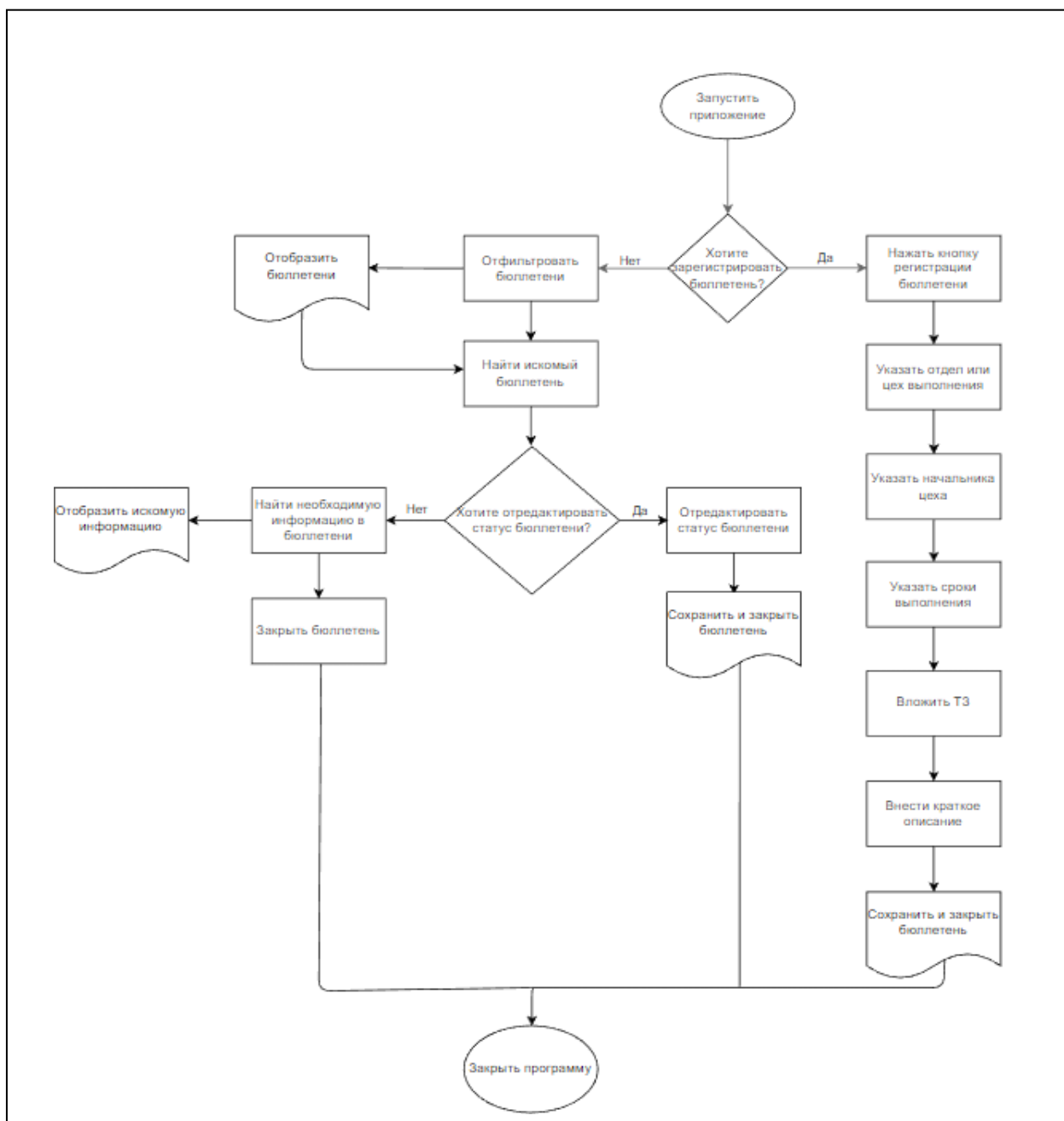


Рис. 3.1. Блок-схема работы программы

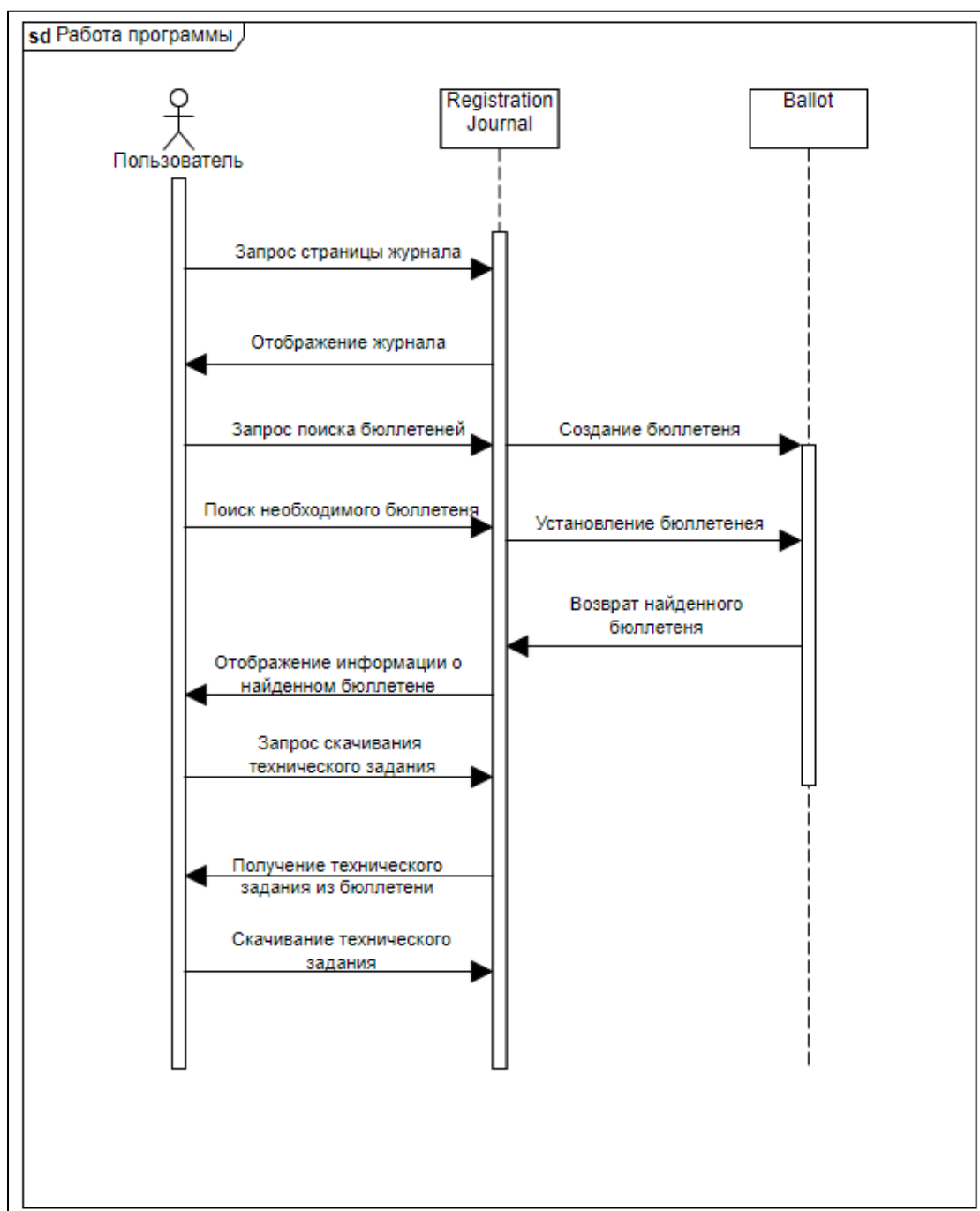


Рис. 3.2. Диаграмма последовательностей программы

4. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ СИСТЕМЫ

4.1. Структура программного обеспечения и функции его компонентов

Для работы МИС необходимо следующее ПО:

- операционная система;
- инструментальное средство разработки;
- дополнительные средства.

Данный журнал регистрации и учёта бюллетеней успешно функционирует с операционной системой Windows 10 и выше.

Инструментальные средства, используемые в разработке модуля информационной системы - Visual Studio Code, интегрированная среда разработки, обеспечивающая запуск программной формы.

Язык программирования PHP.

Дополнительные средства: браузер, Laravel 8, Composer.

4.2. Выбор компонентов программного обеспечения

4.2.1. Операционная система

Для разработки журнала регистрации и учёта бюллетеней была выбрана ОС Windows 10 от компании Microsoft. Данный модуль информационной системы успешно функционирует с ОС Windows, которая поддерживает работу Visual Studio Code и Laravel.

4.2.2. Инструментальное средство разработки и язык программирования

В качестве среды разработки была выбрана Visual Studio Code по следующим причинам:

- бесплатная среда разработки;

- способность реализовать графически интуитивно-понятный пользовательский интерфейс;
- поддержка ЯП PHP, Blade, Java script;
- возможность создания и подключения ряда дополнительных библиотек и плагинов, увеличивающие возможности системы.

Поскольку журнал регистрации и учёта бюллетеней разрабатывается для предприятия, которое использует ОС Windows, наиболее удобно использовать Laravel, как инструментальное средство разработки. Laravel предоставляет ряд возможностей для создания кроссплатформенных приложений, использующих базу данных, работающую по сети, а также позволяет разрабатывать интеллектуальные клиенты, т.е. приложения с полнофункциональным графическим интерфейсом, простые в развертывании и обновлении, способные работать при наличии подключения к Интернету.

Языком разработки был выбран PHP как язык наиболее оптимальный в плане поддержки необходимых операций. PHP - это скриптовый язык программирования, который используется для создания динамических веб-страниц и приложений. Он широко используется в веб-разработке для создания интерактивных веб-страниц, форм, обработки данных и управления базами данных. PHP работает на серверной стороне и может генерировать HTML-код, который посылается клиентскому браузеру в виде обычной веб-страницы. Поэтому этому PHP отлично подходит для разработки данного приложения.

4.2.3. Вспомогательное программное обеспечение

Вспомогательное ПО не требуется.

4.3. Разработка прикладного программного обеспечения

Для написания модуля информационной системы необходимо использовать концепцию ООП, которая основывается на объектах предметной области с определенными свойствами и поведением, где каждый объект является экземпляром класса, а его свойства и методы определены в соответствующем классе.

Структура прикладного программного обеспечения представлена диаграммой классов на рисунке 4.1. Описание классов, необходимых для реализации данного ПО, представлено в таблицах 4.1 – 4.4.

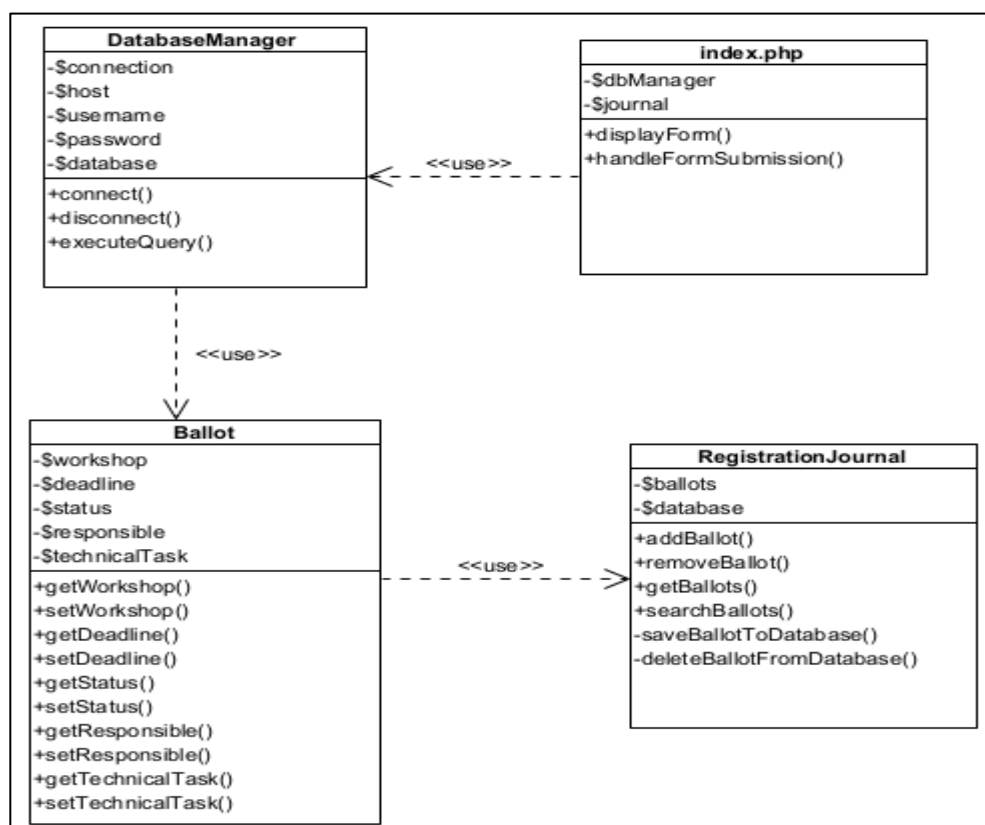


Рис. 4.1. Диаграмма классов

Класс index.php является главным файлом всего приложения.

Таблица 4.1

Модель класса index.php

№	Тип элемента	Название	Модификатор доступа	Описание
1	Поле	\$dbManager	private	Объект класса DatabaseManager для управления базой данных.
		\$journal	private	Объект класса RegistrationJournal для управления журналом бюллетеней.
2	Метод	displayForm()	public	Метод, отображающий форму для регистрации бюллетеня.
		handleFormSubmission()	public	Метод обрабатывающий данные, отправленные формой при регистрации бюллетеня.

Класс DatabaseManager управляет базой данных.

Таблица 4.2

Модель класса DatabaseManager

№	Тип элемента	Название	Модификатор доступа	Описание
1	Поле	\$connection	private	Объект mysqli для установки соединения с базой данных.
		\$host	private	Хост базы данных.
		\$username	private	Имя пользователя базы данных.
		\$password	private	Пароль пользователя базы данных.
		\$database	private	Имя базы данных.
2	Метод	connect()	public	Устанавливает соединение с базой данных.
		disconnect()	public	Закрывает соединение с бд.
		executeQuery(\$query)	public	Выполняет SQL-запрос к бд

Класс Ballot представляет бюллетень.

Таблица 4.3

Модель класса Ballot

№	Тип элемента	Название	Модификатор доступа	Описание
1	Поле	\$workshop	private	Название цеха производства или лаборатории.
		\$deadline	private	Срок выполнения задания.
		\$status	private	Статус выполнения задания.
		\$responsible	private	Ответственное лицо за задание.
		\$technicalTask	private	Техническое задание.
2	Метод	getWorkshop()	public	Метод, возвращающий название цеха производства или лаборатории.
		setWorkshop()	public	Метод, задающий название цеха производства или лаборатории.
		getDeadline()	public	Метод, возвращающий срок выполнения задания.
		setDeadline()	public	Метод, задающий срок выполнения задания.
		getStatus()	public	Метод, возвращающий статус выполнения задания.
		setStatus()	public	Метод, задающий статус выполнения задания.
		getResponsible()	public	Метод, возвращающий ответственное лицо за задание.
		setResponsible()	public	Метод, задающий ответственное лицо за задание.
		getTechnicalTask()	public	Метод, возвращающий техническое задание.
		setTechnicalTask()	public	Метод, задающий техническое задание.

Класс RegistrationJournal представляет журнал регистрации и учёта бюллетеней.

Модель класса RegistrationJournal

№	Тип элемента	Название	Модификатор доступа	Описание
1	Поле	\$ballots	private	Массив бюллетеней, зарегистрированных в журнале.
		\$database	private	Объект класса DatabaseManager для работы с базой данных.
2	Метод	addBallot()	public	Добавляет бюллетень в журнал и сохраняет его в базе данных.
		removeBallot()	public	Удаляет бюллетень из журнала и базы данных.
		getBallots()	public	Возвращает список всех бюллетеней в журнале.
		searchBallots()	public	Ищет бюллетени по заданному фильтру.
		saveBallotToDatabase()	private	Сохраняет бюллетень в базе данных.
		deleteBallotFromDatabase() ()	private	Удаляет бюллетень из базы данных.

4.4. Особенности реализации, эксплуатации и сопровождения системы

Журнал учёта бюллетеней доступен работникам, которые хотят снизить время затраты на поиск необходимых бюллетеней, снизить уровень бумажной номенклатуры, предназначен для руководителей, которые хотят отслеживать ход выполнения работы в установленные сроки. Для работы с проектом необходимо и достаточно владеть ПК на уровне среднего пользователя и иметь подключение к сети Интернет. Основной функционал проекта эффективно и качественно выполняет своё назначение.

Разработчик или администратор занимается сопровождением модуля информационной системы с момента её разработки до момента вывода системы из эксплуатации. Техническая поддержка пользователей осуществляется постоянно.

4.5. Руководство пользователя

4.5.1. Требования к условиям эксплуатации

Минимальные системные требования:

- операционная система: Windows, поддерживающая Laravel;
- Visual Studio;
- доступ в сеть Интернет.

4.5.2. Установка и настройка

Для корректной работы приложения требуется установка Lamp и Composer. Composer - это менеджер зависимостей для языка программирования PHP. Он позволяет управлять зависимостями (библиотеками и фреймворками), необходимыми для работы PHP-проекта, устанавливая их автоматически и обновляя до актуальной версии.

Также пользователь не имеет доступа к настройкам приложения, так как приложение полностью приспособлено для работы пользователей.

Пользователи имеют различные права доступа, в зависимости от данных при авторизации им открываются различные программные формы для дальнейшей работы.

4.5.3. Порядок и особенности работы

Для работы необходимо запустить веб-страницу журнала и учёта бюллетеней. При запуске необходимо авторизоваться, т.е. ввести логин и пароль. После входа пользователь попадает на главную страницу, где и есть весь перечень уже зарегистрированных бюллетеней. На этой странице он может посмотреть их статус, а нажав на специальные кнопки зарегистрировать новые, изменить статус работы и отыскать по фильтрам

необходимые бюллетени.

Результатом работы программы является как перечень бюллетеней, предоставляемый пользователю, так и их регистрация, поиск и редактирование статуса работы.

Примеры интерфейсов МИС представлены на рисунках 4.2-4.5.

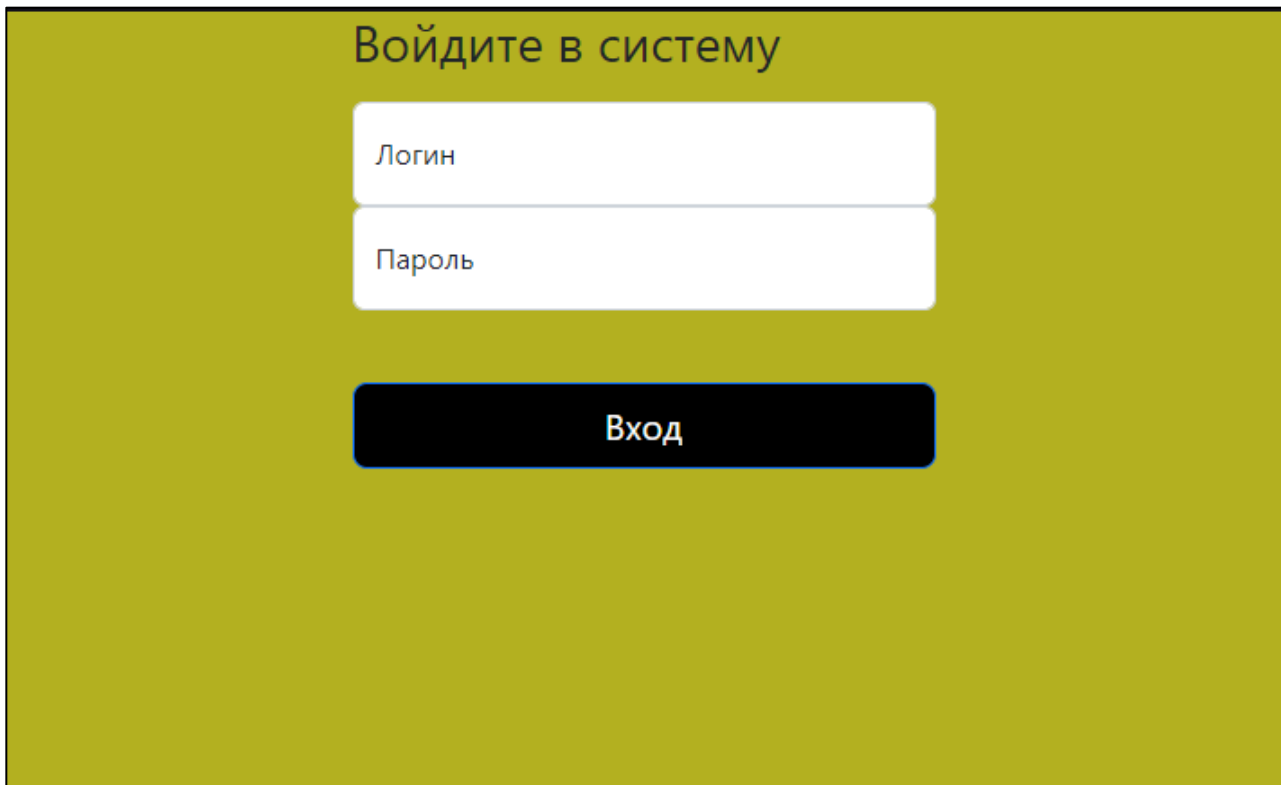
The image shows a login interface with a solid olive-green background. At the top center, the text "Войдите в систему" is displayed in a dark green, sans-serif font. Below this, there are two white rectangular input fields with thin grey borders. The first field is labeled "Логин" and the second is labeled "Пароль". Both labels are in a grey, sans-serif font. Below the password field is a dark blue button with rounded corners and the word "Вход" in white, bold, sans-serif font.

Рис. 4.2. Окно входа

Журнал технических бюллетеней						
			Изменить статус выполнения	Регистрация новых бюллетеней	Поиск бюллетеней по фильтру	
Номер	Статус	Отдел	Начальник отдела	Крайний срок выполнения	Задание	Техническое задание
1	Выполнено	ЦАПЖЦИ	Цызыров Максим Сергеевич	07.12.2003	Гибочный станок	Гибочный станок.doc
Номер	Статус	Отдел	Начальник отдела	Крайний срок выполнения	Задание	Техническое задание
2	В работе	УКБП	Петров Иван Николаевич	11.12.2023	Сверло технологическое	Сверло технологическое.doc
Номер	Статус	Отдел	Начальник отдела	Крайний срок выполнения	Задание	Техническое задание
3	Выполнено	ЦАПЖЦИ	Артемов Артем Артемович	10.12.2023	Прокатный станок	Прокатный станок.doc
Номер	Статус	Отдел	Начальник отдела	Крайний срок выполнения	Задание	Техническое задание
4	Выполнено	ЦАПЖЦИ	Станков Сергей Максимович	12.12.2003	Токарный станок	Токарный станок.doc

Рис. 4.3. Окно технических бюллетеней

Поиск бюллетеней

Название

Руководитель

Номер

Рис. 4.4. Окно поиска необходимых бюллетеней

Статус

Номер

Статус Выберите статус ▼

- Выберите статус
- Выполнено
- В работе**

Изменить статус

Рис. 4.6. Окно для изменения статуса бюллетеней

4.5.4. Исключительные ситуации и их обработка

Все возможные варианты исключительных ситуаций, возникновение которых возможно в период работы приложения, отображены в Таблице 4.5.

Таблица 4.5

Исключительные ситуации, их причины и решения

№	Ситуация	Возможные причины возникновения	Решение
1	Потеря доступа к БД	Ошибка подключения к сети Интернет	Переподключение к сети Интернет; Перезапуск программной формы
2	Ошибка авторизации	Ошибка БД; Ошибка подключения к сети Интернет	Переподключение к сети Интернет; Обращение в тех. Поддержку
3	Отказ запуска приложения	Системная ошибка; Ошибка приложения	Перезапуск приложения; Обращение в тех. Поддержку
4	Ошибка синхронизации данных	Отсутствие подключения к сети Интернет; Ошибка БД	Подключиться к сети Интернет; Обращение в тех. Поддержку

5. ТЕСТИРОВАНИЕ

На каждом этапе разработки модуль информационной системы проходил тестирование, с целью обнаружения программных ошибок.

Для проверки работоспособности приложения были использованы различные схемы и механизмы тестирования.

5.1. Модульное и интеграционное тестирование

5.1.1. Условия и порядок тестирования

Модульное тестирование было проведено с целью проверить на работоспособность отдельные изолированные модули исходного кода программы, а также выявление локализованных в модуле ошибок в реализации алгоритмов и определение степени готовности системы к переходу на следующий этап разработки и тестирования.

Unit tests разрабатывались в приложении Visual Studio Code, примеры которых представлены в п. 5.1.2.

Интеграционные тесты были проведены с целью оценки компонентов приложения на более обширном уровне, чем модульные тесты. Результаты интеграционных тестов подтвердили, что компоненты приложения работают совместно для получения ожидаемого результата.

Интеграционные тесты проводились с использованием базовых функций фреймворка Laravel, для проверки работы с БД и корректного обрабатывания функций.

Примеры интеграционных тестов представлены в п. 5.1.2.

5.1.2. Исходные данные для контрольных примеров

Журнал регистрации и учёта бюллетеней был протестирован при помощи unit tests, с помощью которых была проведена проверка корректности работы каждой функции программы.

Для достижения результата необходимо и достаточно подвергнуть тестированию механизм перехода между формами классов, связанных между собой внутри программы, а также связь с БД.

Рассмотрим примеры модульных и интеграционных тестов:

Unit test «Корректность добавления новых бюллетеней в журнал»

```
public function getRequestTest()
    public function testAddBallot()
    {
        $journal = new RegistrationJournal();
        $ballot = new Ballot();

        $journal->addBallot($ballot);

        $this->assertEquals([$ballot], $journal->getBallots());
    }
```

Unit test «Корректность поиска бюллетеней по фильтру»

```
public function testSearchBallots()
{
    $journal = new RegistrationJournal();
    $ballot1 = new Ballot();
    $ballot1->setWorkshop('Workshop A');
    $ballot2 = new Ballot();
    $ballot2->setWorkshop('Workshop B');
    $journal->addBallot($ballot1);
    $journal->addBallot($ballot2);

    $result = $journal->searchBallots('Workshop A');
```

```

    $this->assertEquals([$ballot1], $result);
}

```

Тестирование остальных методов проведено аналогичным образом.

5.2. Функциональное тестирование

5.2.1. Условия и порядок тестирования

Тестовое покрытие в функциональном тестировании было получено путем анализа граничных значений, т.е. технике тестирования, направленной на проверку поведения системы на граничных значениях входных данных.

5.2.2. Исходные данные для контрольных примеров

В ходе анализа были выявлены следующие классы эквивалентности:

Класс 1: Название бюллетени не может содержать символ «?».

Класс 2: Запись Задание может принимать значение от 1 до 100 символов.

Класс 3: Запись Начальник отдела может принимать значение от 1 до 40 символов.

Класс 4: Запись Отдел может принимать значение от 1 до 20 символов.

Класс 5: Запись Номер может принимать значение от 1 до 99999999.

Исходные данные и результаты проверки классов эквивалентности представлены в таблице 5.1.

Таблица 5.1

Классы эквивалентности

Исходные данные	Ожидаемый результат	Результат
Класс эквивалентности 1.		
«??»	Ошибка. Запись не добавлена.	Ошибка. Запись не добавлена.

Классы эквивалентности

Исходные данные	Ожидаемый результат	Результат
Класс эквивалентности 1.		
«Прокатный Станок»	Успешно	Успешно
«Сверлильный станок»	Успешно	Успешно
«Гибочный станок»	Успешно	Успешно
«Сверло»	Успешно	Успешно
«Лезвие?»	Ошибка. Запись не добавлена.	Ошибка. Запись не добавлена.
Класс эквивалентности 2.		
«»	Ошибка. Запись не добавлена.	Ошибка. Запись не добавлена.
«С»	Успешно	Успешно
«Св»	Успешно	Успешно
«СверлоСверлоСверлоСверлоСверло»	Успешно	Успешно
«СверлоСверлоСверлоСверлоСверло СверлоСверлоСверлоСверлоСверло»	Успешно	Успешно
«СверлоСверлоСверлоСверлоСверло СверлоСверлоСверлоСверлоСверло СверлоСверлоСверлоСверлоСверло СверлоСверлоСверлоСверлоСверло»	Ошибка. Запись не добавлена.	Ошибка. Запись не добавлена.
Класс эквивалентности 3.		
«»	Ошибка. Запись не добавлена.	Ошибка. Запись не добавлена.
«М»	Успешно	Успешно
«Максим»	Успешно	Успешно
«Максим Сергеевич»	Успешно	Успешно
«Максим Сергеевич Цызыров»	Успешно	Успешно
«Максим Максим Сергеевич Сергеевич Цызыров Цызыров»	Ошибка. Запись не добавлена.	Ошибка. Запись не добавлена.
Класс эквивалентности 4.		
«»	Ошибка. Запись не добавлена.	Ошибка. Запись не добавлена.
«Ц»	Успешно	Успешно
«ЦАП»	Успешно	Успешно
«ЦАПЖ»	Успешно	Успешно
«ЦАПЖЦИ»	Успешно	Успешно
«ЦАПЖЦИ ЦАПЖЦИ ЦАПЖЦИ ЦАПЖЦИ»	Ошибка. Запись не добавлена.	Ошибка. Запись не добавлена.
Класс эквивалентности 5.		
«-1»	Ошибка. Запись не добавлена.	Ошибка. Запись не добавлена.
«1»	Успешно	Успешно
«12»	Успешно	Успешно
«165»	Успешно	Успешно
«1889»	Успешно	Успешно
«999999999»	Ошибка. Запись не добавлена.	Ошибка. Запись не добавлена.

5.3. Нагрузочное тестирование

5.3.1. Условия и порядок тестирования

Любое программное обеспечение должно работать под нагрузкой длительное время. Нагрузочное тестирование было проведено с целью определить, как и с какой скоростью работает программа под определенной нагрузкой.

При реализации нагрузочного тестирования диагностике подверглась внедряемая БД.

5.3.2. Исходные данные для контрольных примеров

В БД содержится 1000 записей. Результаты нагрузочного тестирования приведены в таблице 5.2.

Таблица 5.2

Нагрузочное тестирование

Поисковой запрос	Время запроса (сек.)
Перечень бюллетеней, где Номер бюллетеня = «2»	0.0004
Перечень бюллетеней, где Начальник отдела = «Станков Сергей Максимович».	0.0006
Перечень бюллетеней, где Задание = «Сверло технологическое»	0.0008

5.4. Тестирование интерфейсов

При реализации тестирования интерфейсов были исследованы все программные формы на наличие распространенных ошибок и багов.

5.5. Результаты тестирования

Электронный журнал регистрации и учета бюллетеней полностью прошёл диагностику различными механизмами тестирования. На каждом этапе

разработки система прошла тестирование, по результатам которых были сделаны выводы по работе компонентов, входящих в систему, и системы в целом. Подводя итог можно сказать, что электронный журнал регистрации и учета бюллетеней работает с отсутствием ошибок в программе, без перегрузов и багов в стандартном режиме работы.

6. ЭКОНОМИЧЕСКИЙ РАЗДЕЛ

6.1. Экономический анализ внедрения системы

Основной целью разрабатываемой системы было сокращение времени на обработку и учёт документов, которые раньше записывались и хранились в бумажной форме.

Для предсказания возможных экономических выгод, связанных с внедрением журнала регистрации и учета бюллетеней, были произведены расчеты экономических показателей.

6.2. Оценка трудозатрат на разработку модуля информационной системы

Для оценки трудозатрат на разработку журнала регистрации и учета бюллетеней использовалась модель COCOMO II (Constructive Cost Model). Модель использует следующие параметры: количество тысяч строк кода, характеристика продукта, характеристика аппаратного обеспечения, характеристика персонала, характеристика проекта.

Все параметры, кроме количества строк кода, являются константами.

Таблица базовых констант описана в таблице 6.1.

Таблица 6.1

Константы модели COCOMO

Тип проекта	ai	bi
Органический	3.2	1.05
Полураздельный	3	1.12
Встроенный	2.8	1.2

Типы проекта делятся на 3 группы:

- органический - проект, не имеющий жестких требований;
- полураздельный - проект, который имеет смешанные требования;
- встроенный - проект, который имеет жесткие требования.

Журнал регистрации и учёта бюллетеней считается полураздельным проектом, поскольку имеет смешанные требования.

В модели СОСОМО II существуют коэффициенты, которые включают в себя субъективные оценки характеристик продукта, персонала и аппаратного обеспечения. Рейтинг коэффициентов делится на 6 категорий:

- очень низкий;
- низкий;
- средний;
- высокий;
- очень высокий;
- критический.

Выбранные коэффициенты рейтинга описаны в таблице 6.2.

Таблица 6.2

Коэффициенты рейтинга

Факторы стоимости	Рейтинг	Значение
Требуемая надежность ПО	Высокий	1.15
Размер БД приложения	Низкий	0.94
Сложность продукта	Средний	0,85
Требования к быстродействию	Высокий	1.1
Ограничения памяти	Средний	1
Неустойчивость окружения	Средний	0,87
Время восстановления	Высокий	1.07
Аналитические способности	Высокий	0.86
Способности к разработке ПО	Высокий	0.91
Опыт разработки	Высокий	0.86
Опыт использования ВМ	Высокий	0.9
Опыт языка	Высокий	0.95
Инструменты разработки	Очень высокий	0.82
Методы разработки	Высокий	0.91
График разработки	Средний	1

Для оценки трудозатрат в человеко-месяцах необходимо подставить полученные значения в следующую формулу:

$$E = a_i \cdot KLoC^{bi} \cdot \Phi C \quad (1)$$

где: E - трудозатраты в человеко-месяцах; KLoC = 0,5
- оценочный размер программы в тысячах строках кода; $\Phi C = 0.40$ - факторы стоимости коэффициентов.

$$E = 3,2 \cdot 0,5^{1,12} \cdot 0,40 = 0,58$$

Полученное значение переведем в человеко-часы:

$$Eh = E \cdot 22 \cdot 8 \quad (2)$$

где: Eh - трудозатраты в человеко-часах; E - трудозатраты в человеко-месяцах; 22 – количество рабочих дней в месяце; 8 – количество рабочих часов в день.

$$Eh = 0,58 \cdot 22 \cdot 8 = 102,08 \text{ (ч – ч)}$$

На разработку журнала регистрации и учета бюллетеней необходимо 102 человеко-часов.

6.3. Расчёт трудовых и стоимостных показателей

Проведём расчёт и анализ трудовых затрат на разработку журнала регистрации и учёта бюллетеней. В среднем в месяце 22 рабочих дня по 8 часов. Заработная плата инженера-программиста составляет 140 руб/час до вычета налогов.

Трудозатраты на заполнение электронного журнала регистрации и учета бюллетеней представлены в таблице 6.5.

Таблица 6.5

Трудозатраты на процессы предприятия

№	Наименование показателя	Трудоемкость (мин.)		Разница (мин.)
		До внедрения МИС	После внедрения МИС	
1	Сбор технической документации	15	15	0

Трудозатраты на процессы предприятия

№	Наименование показателя	Трудоемкость (мин.)		Разница (мин.)
		До внедрения МИС	После внедрения МИС	
2	Внесение в журнал регистрации и учета бюллетеней	25	5	20
ИТОГО		40	20	20

Исходя из полученных данных был сделан вывод, что при внедрении электронного журнала регистрации и учета бюллетеней время на сбор необходимой технической документации не изменилось, проведение аудита действий пользователей уменьшилось до 150 минут, а время внесения в журнал регистрации и учета бюллетеней сократилось до 5 минут. Благодаря этому эффективность данных операций повысилась в 2 раза.

6.4. Расчёт окупаемости МИС

АО «УКБП» ежедневно получает различные заказы и задания для. Заработная плата сотрудников предприятия представлена в таблице 6.6.

Таблица 6.6

Заработная плата сотрудников

Должность	Заработная плата (руб/час)
Специалист по документообороту	120
Инженер-программист	140

Рассмотрим ситуацию до внедрения МИС.

Специалисты по документообороту, специализирующиеся в области ведения, оформления и регистрации технических документов вручную вносит информацию из технической документации в журнал. Учитывая, что на заполнения бумажного варианта журнала уходит 25 минут, то за полный рабочий день можно будет заполнить всего 19 бюллетеней. Учитывая, что в день

заполняют примерно 10 бюллетеней и на это уходит 250 минут, то в месяц с такими это занимает 5500 минут или же 92 часа. Соответственно, исходя из этих данных, предприятие тратит на это 11 040 рублей.

Рассмотрим ситуацию после внедрения МИС.

На заполнение 1 бюллетеня технической документации в электронном журнале регистрации и учета бюллетеней тратится 5 минут рабочего времени, соответственно на заполнения 10 бюллетеней в рабочий день будет тратиться 50 минут, а в месяц 1100 минут или же 18 часов. Исходя из этих данных, предприятие, после внедрения системы, будет тратить на это 2160 рублей.

Стоимость электронного журнала регистрации и учета бюллетеней при заработной плате инженера-программиста 140 рублей/час и времени на разработку, равную 102 часа составит 14 280 рублей.

Основываясь на математических вычислениях, можно сделать вывод, что срок окупаемости электронного журнала регистрации и учета бюллетеней составит 1,6 месяцев с момента внедрения.

В вышеприведенных расчетах не учитывались амортизационные расходы.

6.5. Выводы экономического раздела

Согласно расчётам, за счёт внедрения электронного журнала регистрации и учета бюллетеней будет достигнуто увеличение эффективности в 2 раза, что способствует достижению поставленных целей по сокращению времени на заполнение технических бюллетеней.

Срок окупаемости составит 1,6 месяцев с момента внедрения.

ЗАКЛЮЧЕНИЕ

В результате разработки электронного журнала регистрации и учета бюллетеней было написано приложение на языке РНР, работающее на персональных компьютерах.

Данная система выполняет поставленные функции без ошибок.

Разработанное программное обеспечение легко интегрируется с существующими бизнес-процессами без необходимости дополнительных усилий для обучения. Взаимодействие с приложением не требует от пользователей навыков работы с персональным компьютером.

Данная система рассматривает возможность расширения и добавления новых функциональных возможностей.

Результатом выпускной квалификационной работы является реализованный электронный журнал регистрации и учета бюллетеней.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Родионов В. В. Выполнение выпускной квалификационной работы бакалавра. Учебно-методическое пособие для студентов направления 09.03.02 «Информационные системы и технологии» / В. В. Родионов. – Ульяновск: УлГТУ, 2017. – 74 с.
2. ГОСТ 34.602-89 Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы.
3. Документация по Visual Studio Code [Электронный ресурс]. - официальный сайт - официальный сайт. – <https://code.visualstudio.com/docs>
4. Документация PHP [Электронный ресурс]. - официальный сайт - <https://www.php.net/docs.php/>.
5. Документация Laravel [Электронный ресурс]. - официальный сайт - <https://laravel.com/docs/10.x/documentation/>.
6. Справочное руководство по MySQL [Электронный ресурс]. - официальный сайт. - <http://www.mysql.ru/docs/man/Reference.html/>.
7. Требования к оформлению и защите бакалаврской работы: практикум / сост.: Т. В. Корсакова, Е. Н. Згуральская. – Ульяновск: УлГТУ, 2021. – 37 с.
8. Федеральный Государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 09.03.02 информационные системы и технологии: утвержден приказом Министерства образования и науки Российской Федерации от 19 сентября 2017 г. № 926 / Министерства образования и науки Российской Федерации. - Москва: 2013 г.
9. Чистов Д. В. Проектирование информационных систем. Учебник и практикум для академического бакалавриата / Д. В. Чистов. – Москва: Юрайт, 2016. – 257 с.