

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего
образования «МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ»
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)
филиал «РКТ» МАИ в г. Химки Московской области

Специальность 09.02.03 «Программирование в компьютерных системах»

ОТЧЕТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ
ПМ.02 Разработка и администрирование баз данных

Студент
группы МП 41-18 _____ / Коробков И. Д.

Руководитель
практики от филиала _____ / Никаноров М.С.

Руководитель
практики от организации _____ / Марченко А.В.

2021г.

ПРОГРАММА ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ

По специальности 09.02.03 «Программирование в компьютерных системах»

Дата	Наименование выполняемых работ	Количество дней практики

Руководитель практики от филиала «РКТ» МАИ преподаватель

Никаноров Михаил Сергеевич_____

(фамилия, имя, отчество)

Дата _____

(подпись)

АТТЕСТАЦИОННЫЙ ЛИСТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ

студент Коробков Иван Дмитриевич

ФИО

обучающийся на 4 курсе по специальности СПО 09.02.03 «Программирование в компьютерных системах» успешно прошел(ла) производственную практику по профессиональному модулю

ПМ.03.01 «Участие в интеграции программных модулей» в объеме часов с « »
2021 г. по «30» 2021 г. в организации /на предприятии МКБ «Факел»

Виды и качество выполнения работ

Виды работ, выполненных обучающимся(ейся) во время практики	Объем работ	Качество выполнения работ в соответствии с технологией и (или) требованиями организации, в которой проходила практика

Руководитель практики от организации / предприятия (должность, Ф.И.О.)

Марченко Антон Васильевич

Дата

подпись

Расшифровка подписи

МП.

Руководитель практики от филиала «РКТ» МАИ в (должность, Ф.И.О.)

Никаноров Михаил Сергеевич

Дата

подпись

Расшифровка подписи

ХАРАКТЕРИСТИКА

Обучающийся (аяся) Коробков Иван Дмитриевич Группы МП-41-18
специальности СПО 09.02.03 «Программирование в компьютерных системах»
проходил производственную (учебную) практику по профессиональному модулю
ПМ.03.01«Участие в интеграции программных модулей»

в период с _____ по _____

в организации /на предприятии МКБ «Факел».

За время работы проявил себя как ответственный/безответственный,
исполнительный /неисполнительный, коммуникабельный/замкнутый,
доброжелательный/наглый сотрудник(студент) и т.д.

Обучающийся (аяся) обладает общими компетенциями (ОК) включающими в себя
способность:

ОК 1. Понимать сущность и социальную значимость своей будущей профессии, проявлять
к ней устойчивый интерес.

ОК 2. Организовывать собственную деятельность, определять методы и способы
выполнения профессиональных задач, оценивать их эффективность и качество.

ОК3. Решать проблемы, оценивать риски и принимать решения в нестандартных
ситуациях.

ОК 4. Осуществлять поиск, анализ и оценку информации, необходимой для постановки и
решения профессиональных задач, профессионального и личностного развития.

ОК 5. Использовать информационно-коммуникационные технологии для
совершенствования профессиональной деятельности.

ОК 6. Работать в коллективе и команде, обеспечивать ее сплочение, эффективно общаться
с коллегами, руководством, потребителями.

ОК 7. Ставить цели, мотивировать деятельность подчиненных, организовывать и
контролировать их работу с принятием на себя ответственности за результат выполнения
заданий.

ОК 8. Самостоятельно определять задачи профессионального и личностного развития,
заниматься самообразованием, осознанно планировать повышение квалификации.

ОК 9. Быть готовым к смене технологий в профессиональной деятельности.

В соответствии с ПМ. __ овладел профессиональными компетенциями (ПК),

К работе относился _____

Цели и задачи практики достигнуты, (достигнуты не в полном объеме).

Оценка за практику _____

Руководитель от организации (должность, фамилия, имя, отчество)

Ведущий программист Марченко Антон Васильевич

Дата _____

дата

подпись

Расшифровка подписи

ДНЕВНИК ПРОХОЖДЕНИЯ ПРАКТИКИ

[illegible]

Содержание

1. Общие сведения об объекте.	7
1.1. Описание места прохождения практики.	7
2. Сбор и анализ исходных данных.	8
2.1. Анализ задачи.....	8
2.2. Назначение задачи.	8
2.3. Актуальность.....	8
2.4. Целевая аудитория.....	8
2.5. Функции веб-приложения.....	8
2.6. Характеристики веб-приложения.....	8
3. Определение метрик программного обеспечения.....	9
3.1. Состав задач тестирования.	9
4. Планирование работ по созданию программного обеспечения.....	10
4.1. Требования к техническому и программному обеспечению.....	10
4.2. Функциональные характеристики программного обеспечения.....	10
5. Определение стадий и этапов разработки программ и документации.....	11
5.1. Стадии и этапы разработки ПО.	11
5.1.1. Определение целей и определение альтернативных решений.	11
5.1.2. Выявление и устранение рисков.....	11
5.1.3. Разработка следующей версии продукта.	11
5.1.4. Обзор и планирование следующего этапа.	11
5.2. Документирование ПО.	11
5.2.1. Используемые стандарты.	11
5.2.2. Руководство ПО.....	12
6. Определение формы и структуры входных и выходных данных.....	13
7. Составление необходимых диаграмм UML и/или структурной схемы программы.....	14
7.1. Диаграмма прецедентов (вариантов).	14
7.2. Диаграмма классов.	15
7.3. Диаграмма деятельности.....	16
7.4. Диаграмма состояний.....	17
7.5. И/Или структурная схема ПО.....	18
Заключение.....	20
Список литературы.....	21

1. Общие сведения об объекте.

1.1. Описание места прохождения практики.

Место прохождения практики Акционерное общество «Машиностроительное конструкторское бюро «Факел» имени академика П.Д. Грушина.

МКБ «Факел» - первопроходец отечественного ракетостроения и лидер в разработке ракет для систем ПВО–ПРО, предприятие, обладающее уникальными технологиями, постоянно развивающее традиции, заложенные его основателем - дважды Героем социалистического труда академиком Петром Дмитриевичем Грушиным.

За 66 лет деятельности МКБ «Факел» разработало свыше 50 типов зенитных ракет и их модификаций и обеспечило боевыми средствами поражения эффективные системы ПВО-ПРО, которые приняты на вооружение более чем в 60 странах мира.

Организация российской ракетно-космической промышленности, расположенная в г. Химки. Основная продукция — управляемые ракеты класса «земля-воздух» для зенитных ракетных комплексов. С 2002 года входит в АО «Концерн ВКО «Алмаз - Антей», финансово-промышленную группу «Оборонительные системы».

АО «МКБ «Факел» является головной организацией по разработке зенитных управляемых ракет для зенитных ракетных комплексов С-75 «Двина», 9К33 «Оса», 9М330 «Тор», С-125, С-200, С-300П, С-400 и многих других.

2. Сбор и анализ исходных данных.

2.1. Анализ задачи.

Для прохождения производственной практики, было получено задание – создать веб-приложение в сфере финансов. Веб-приложение должно информировать клиента о состоянии его финансов и активов. На сайте должна быть реализована база данных для хранения информации о клиентах и финансов/активов. Для создания серверной части веб-приложения будет использоваться язык Php. Для клиентской части – JavaScript, Html, CSS, Bootstrap.

2.2. Назначение задачи.

Назначение сайта состоит в том, чтобы информировать теоретических клиентов о состоянии его финансов и активов и возможность взаимодействия с ними.

2.3. Актуальность.

Актуальность заключается в необходимости веб-приложения в сфере финансов для оказания услуг учета и управления финансами клиентов.

2.4. Целевая аудитория.

Активные пользователи Интернета, интересующиеся учетом и управлением своих финансов.

2.5. Функции веб-приложения

К функциям веб-приложения можно отнести: 1. просмотр текущего состояния активов; 2. добавление активов; 3. редактирование активов; 4. удаление активов.

2.6. Характеристики веб-приложения

К характеристикам веб-приложения можно отнести: 1. доступность - приложения доступны с любой точки планеты, где есть Интернет, с любого устройства; 2. адаптивность - В отличие от нативных продуктов, web-application не требуют конкретной операционной системы. Они работают на всех ОС и во всех существующих браузерах; 3. отсутствие программного обеспечения для клиентов - не нужно ничего скачивать и устанавливать. Это упрощает процесс пользования.

3. Определение метрик программного обеспечения.

Метрика Чепина: суть метода состоит в оценке информационной прочности отдельно взятого программного модуля с помощью анализа характера использования переменных из списка ввода-вывода. Все множество переменных, составляющих список ввода-вывода, разбивается на 4 функциональные группы: 1. Р — вводимые переменные для расчетов и для обеспечения вывода, 2. М — модифицируемые, или создаваемые внутри программы переменные, 3. С — переменные, участвующие в управлении работой программного модуля (управляющие переменные), 4. Т — не используемые в программе («паразитные») переменные. Поскольку каждая переменная может выполнять одновременно несколько функций, необходимо учитывать ее в каждой соответствующей функциональной группе. Метрика Чепина: $Q = a_1 * P + a_2 * M + a_3 * C + a_4 * T$, где a_1 , a_2 , a_3 , a_4 — весовые коэффициенты. Весовые коэффициенты использованы для отражения различного влияния на сложность программы каждой функциональной группы. По мнению автора метрики, наибольший вес, равный 3, имеет функциональная группа С, так как она влияет на поток управления программы. Весовые коэффициенты остальных групп распределяются следующим образом: $a_1=1$, $a_2=2$, $a_4=0.5$. Весовой коэффициент группы Т не равен 0, поскольку «паразитные» переменные не увеличивают сложность потока данных программы, но иногда затрудняют ее понимание. С учетом весовых коэффициентов: $Q = P + 2M + 3C + 0.5T$.

3.1. Состав задач тестирования.

Тестирование программного обеспечения (Software Testing) — проверка соответствия реальных и ожидаемых результатов поведения программы, проводимая на конечном наборе тестов, выбранном определенным образом. Цель тестирования — проверка соответствия ПО предъявляемым требованиям, обеспечение уверенности в качестве ПО, обнаружение проблем на более ранних этапах разработки и предотвращение повышения стоимости продукта, обнаружение вариантов использования, которые не были предусмотрены при разработке. Для тестирования ПО будут использоваться фреймворки PHPUnit, Ava. Благодаря им будут протестированы классы управляющие бизнес-логикой и связью с клиентской частью.

4. Планирование работ по созданию программного обеспечения.

4.1. Требования к техническому и программному обеспечению.

В таблице 4.1 перечислены минимальные и рекомендуемые требования к оборудованию для веб-приложения.

Компонент	Минимальная конфигурация	Рекомендуемая конфигурация
Процессор	x86- или x64-разрядный двухъядерный процессор с тактовой частотой 1,9 ГГц, с набором инструкций SSE2	x64-разрядный двухъядерный процессор с тактовой частотой 3,3 ГГц и выше, с набором инструкций SSE2
Память	ОЗУ объемом 2 Гб	ОЗУ объемом 4 Гб и более
Дисплей	Super VGA с разрешением 1024 x 768	Super VGA с разрешением 1024 x 768

Таблица 4.1 – Требования к оборудованию

При запуске приложений на основе модели на компьютере, не соответствующем рекомендуемым требованиям, производительность работы может оказаться недостаточной. Дополнительно удовлетворительную производительность можно получить, выполнив системы, которые используют аппаратную конфигурацию. Например, система с современным четырехъядерным процессором, более низкой тактовой частоты и большей ОЗУ. Приложения на основе модели предназначены для оптимальной работы в сетях со следующими элементами: 1. пропускная способность более 50 КБ/с (400 КБ/с); 2. задержка менее 150мс. Эти значения являются рекомендованными и не гарантируют удовлетворительную производительность. Рекомендуемые значения актуальны для систем, в которых используются готовые формы без пользовательской настройки. Если в готовые формы внесены значительные изменения, рекомендуется проверить реагирование формы, чтобы оценить потребности в пропускной способности. Веб-приложение может выполняться в любом из следующих веб-браузеров, запущенных в указанных операционных системах: 1. Microsoft Edge (последняя открыто выпущенная версия) работает в Windows 10, Windows 8.1, Windows 8, Windows 7; 2. Mozilla Firefox (последняя публично выпущенная версия) работает в Windows 10, Windows 8.1, Windows 8 или Windows 7; 3. Google Chrome (последняя открыто выпущенная версия) работает в Windows 10, Windows 8.1, Windows 8, Windows 7.

4.2. Функциональные характеристики программного обеспечения.

Веб-приложение в сфере финансов предназначено для электронного ведения учета личных средств, их характеристик и движения. Каждый клиент имеет возможность добавлять, изменять, удалять данные о ресурсах и их движениях. Главная задача веб-приложения – создание онлайн-отчетов о наличии и состоянии финансов, а также операциями над ними.

5. Определение стадий и этапов разработки программ и документации.

5.1. Стадии и этапы разработки ПО.

Этапы разработки ПО: 1. определение целей и определение альтернативных решений; 2. выявление и устранение рисков; 3. разработка следующей версии продукта; 4. обзор и планирование следующего этапа.

5.1.1. Определение целей и определение альтернативных решений.

Требования собираются от клиентов, а цели определяются, разрабатываются и анализируются в начале каждого этапа. Затем предлагаются альтернативные решения, возможные для фазы.

5.1.2. Выявление и устранение рисков.

Оцениваются все возможные решения, чтобы выбрать наилучшее из возможных. Затем выявляются риски, связанные с этим решением, и риски устраняются с использованием наилучшей стратегии. В конце создается прототип для наилучшего возможного решения.

5.1.3. Разработка следующей версии продукта.

Выявленные функции разрабатываются и проверяются посредством тестирования. В конце доступна следующая версия программного обеспечения.

5.1.4. Обзор и планирование следующего этапа.

Заказчики оценивают уже разработанную версию программного обеспечения. В конце, начинается планирование следующего этапа.

5.2. Документирование ПО.

5.2.1. Используемые стандарты.

В проекте были использованы стандарты серии ISO 9000. Стандарты международной организации по стандартизации ISO являются наиболее известными и распространенными в мире. Стандарты ISO универсальны, их можно применять в качестве моделей независимо от отрасли, в которой функционирует компания. В следствие этого у модели ISO есть свои неоспоримые преимущества и недостатки. Но основным преимуществом модели ISO является известность, распространенность, признание на мировом уровне. Сейчас стандарты ISO являются обязательным минимумом который должна иметь любая организация существующая на рынке. Но конечно же, вследствие своей универсальности, модель на основе стандартов ISO серии 9000 получилась достаточно “высокоуровневой”.

5.2.2. Руководство ПО.

Предполагается, что пользователь имеет опыт работы с ОС Windows(7 или выше), а также с браузерами (Yandex, Google Chrome и т.д.).

Работоспособность приложения обеспечивается за счет стабильного подключения к интернету, соблюдения минимальных требований к оборудованию, стабильной работы сервера приложения.

Для начала работы пользователю необходимо авторизоваться, нажав на кнопку “Авторизоваться”. После появится всплывающее окно, в полях которого нужно ввести в соответствующие поля паспорт и пароль, и нажать кнопку “Войти”.

На данном этапе могут возникнуть различные проблемы.

“Ошибка, введенные паспорт или пароль неверны” возникает, если введенные данные были некорректны. Для исправления достаточно ввести в поля формы корректные данные.

“Ошибка, пользователь не зарегистрирован” возникает, когда незарегистрированный пользователь пытается авторизоваться. Пользователю необходимо зарегистрироваться, нажав на кнопку “Регистрация” и в модальном окне заполнить все поля и нажать на кнопку “Зарегистрироваться”. Может возникнуть ошибка, информирующая о том, что введенные данные некорректны. Для исправления достаточно ввести в поля формы корректные данные.

Для выхода из профиля нужно нажать на кнопку “Выход”.

Для просмотра записей отчетов необходимо успешно войти в систему, после чего перейти на страницу “Отчеты”, нажав на кнопку “Отчеты”.

Для изменения записей в отчете авторизованному пользователю, на странице “Отчеты”, нужно нажать на кнопку “Изменить”. После чего перейдете на страницу “Редактирования записи”, на ней отображена форма с соответствующими полями для редактирования записи. Чтобы изменения вступили в силу, нужно нажать на кнопку “Изменить” или на кнопку “Назад” для отмены изменений.

Для внесения данных Авторизованный пользователь должен перейти на страницу “Финансы”, нажав кнопку “Финансы”. На странице отображена форма с полями, в которые нужно ввести соответствующие данные.

Во время изменения и ввода записей может возникнуть ошибка, что данные некорректны. Для исправления нужно ввести в поля формы исправленные данные.

6. Определение формы и структуры входных и выходных данных.

В работе приложения участвует база данных, взаимодействие с которой осуществляется посредством ввода и вывода данных. Входными данными являются данные, которые пользователь вводит в специальные формы. Выходные данные – данные, появляющиеся на мониторе во время работы с приложением.

В качестве входных данных служат формы регистрации и авторизации, пополнения счетов. В роле выходных данных выступают отчеты о состоянии и движении средств, информация о пользователе.

7. Составление необходимых диаграмм UML и/или структурной схемы программы.

7.1. Диаграмма прецедентов (вариантов).

Диаграмма поведения, на которой показано множество прецедентов и актеров, а также отношения между ними.

Субъект (actor) - любая сущность, взаимодействующая с системой извне или множество логически связанных ролей, исполняемых при взаимодействии с прецедентами.

Прецеденты (use case) - это описание множества последовательностей действий (включая их варианты), которые выполняются системой для того, чтобы актер получил результат, имеющий для него определенное значение.

Между субъектами и прецедентами - основными компонентами диаграммы прецедентов - могут существовать различные отношения, которые описывают взаимодействие экземпляров одних субъектов и прецедентов, с экземплярами других субъектов и прецедентов.

Отношение включения (include) - указывает, что некоторое заданное поведение для одного прецедента включает в качестве составного компонента поведение другого прецедента.

Отношение расширения (extend) - определяет взаимосвязь экземпляров отдельного прецедента с более общим прецедентом, свойства которого определяются на основе способа совместного объединения данных экземпляров.

Отношение ассоциации (association) - определяет наличие канала связи между экземплярами субъекта и прецедента.

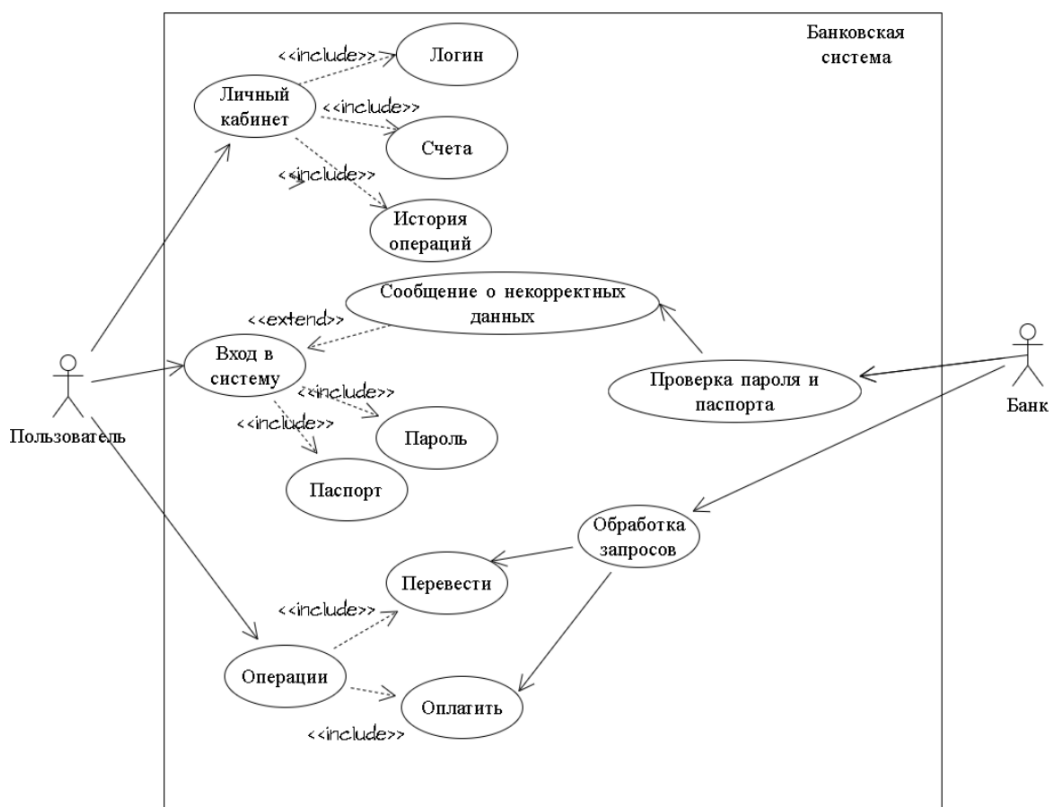


Рисунок 7.1 - Диаграмма прецедентов

На диаграмме (смотреть рисунок 7.1) отображены два актера: Пользователь и Банк. Каждый из актеров может пользоваться системой по-своему.

Пользователь имеет доступ в личный кабинет, который содержит информацию о его логине, счетах и истории операций. Предоставляется возможность входа в систему включающая в себя входные данные в виде паспорта и пароля. В случае предоставления пользователем неверных данных, Банк отправляет сообщение о некорректных введенных данных. Пользователь может совершать операции перевода и оплаты.

Банк проверяет входные данные пользователя при входе в систему на действительность, и в случае ошибки со стороны пользователя банк информирует о ней. Также Банк занимается обработкой запросов Пользователя по операциям перевода и пополнения.

7.2. Диаграмма классов.

Структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов, методов, интерфейсов и взаимосвязей между ними.

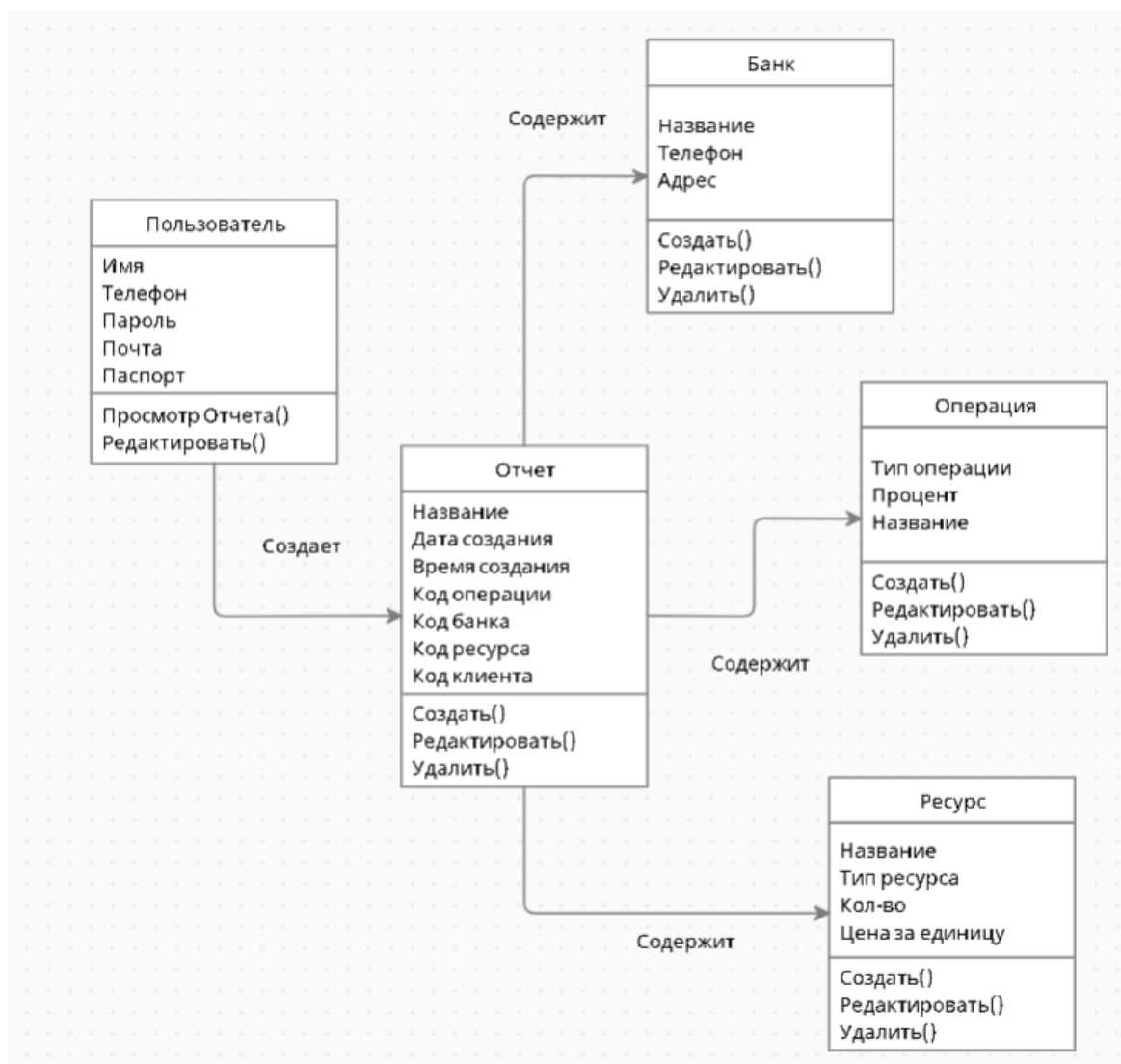


Рисунок 7.2 - Диаграмма классов

На диаграмме(смотреть рисунок 7.2) отображено 5 классов: Ресурс, Операция, Банк, Отчет, Пользователь.

Класс Отчет реализуется за счет наследования от классов Ресурс, Операция, Банк. Это класс определяет, как и из чего должен состоять любой отчет в веб-приложении. В качестве параметров выступают переменные: Название, Дата создания, Время создания, Код операции, Код ресурса, Код Клиента. Методами данного класса являются функции: Создать – создает новую запись, Редактировать – обновляет информацию о записи, Удалить – удаляет запись.

Класс Ресурс содержит информацию о характеристиках имеющихся ресурсов. В качестве параметров выступают переменные: Название, Тип ресурса, Кол-во(количество), Цена за единицу. Методами данного класса являются функции: Создать – создает новую запись, Редактировать – обновляет информацию о записи, Удалить – удаляет запись.

Класс Операция содержит информацию о доступных операциях. В качестве параметров выступают переменные: Название, Тип операции, Процент. Методами данного класса являются функции: Создать – создает новую запись, Редактировать – обновляет информацию о записи, Удалить – удаляет запись.

Класс Банк содержит информацию о используемых клиентом банках. В качестве параметров выступают переменные: Название, Адрес, Телефон. Методами данного класса являются функции: Создать – создает новую запись, Редактировать – обновляет информацию о записи, Удалить – удаляет запись.

Класс Пользователь содержит информацию о пользователе. В качестве параметров выступают переменные: Имя, Телефон, Пароль, Почта, Паспорт. Методами данного класса являются функции: Просмотр отчета – показывает текущее состояние записей, Редактировать – обновляет информацию о пользователе.

7.3. Диаграмма деятельности.

UML-диаграмма, на которой показаны действия, состояния которых описаны на диаграмме состояний.

Данная диаграмма (смотреть рисунок 7.3) разделена на две части: деятельность пользователя, деятельность сервера. Отображает действия Пользователя и Системы при попытке первого выполнить вход или регистрацию.

Деятельность берет начало с выбора Пользователем выполнить Вход или Регистрацию. Заканчивается либо успешным входом в систему с отображением главной страницы, либо вход в систему не удастся.

Деятельность пользователя состоит из входа и регистрации. Заканчивается входом в систему или выходом из нее.

В деятельность системы входит проверка вводимых пользователем данных при регистрации или входе в систему. Заканчивается перемещением пользователя на главную страницу и успешным входом в систему или невозможностью выполнить вход.

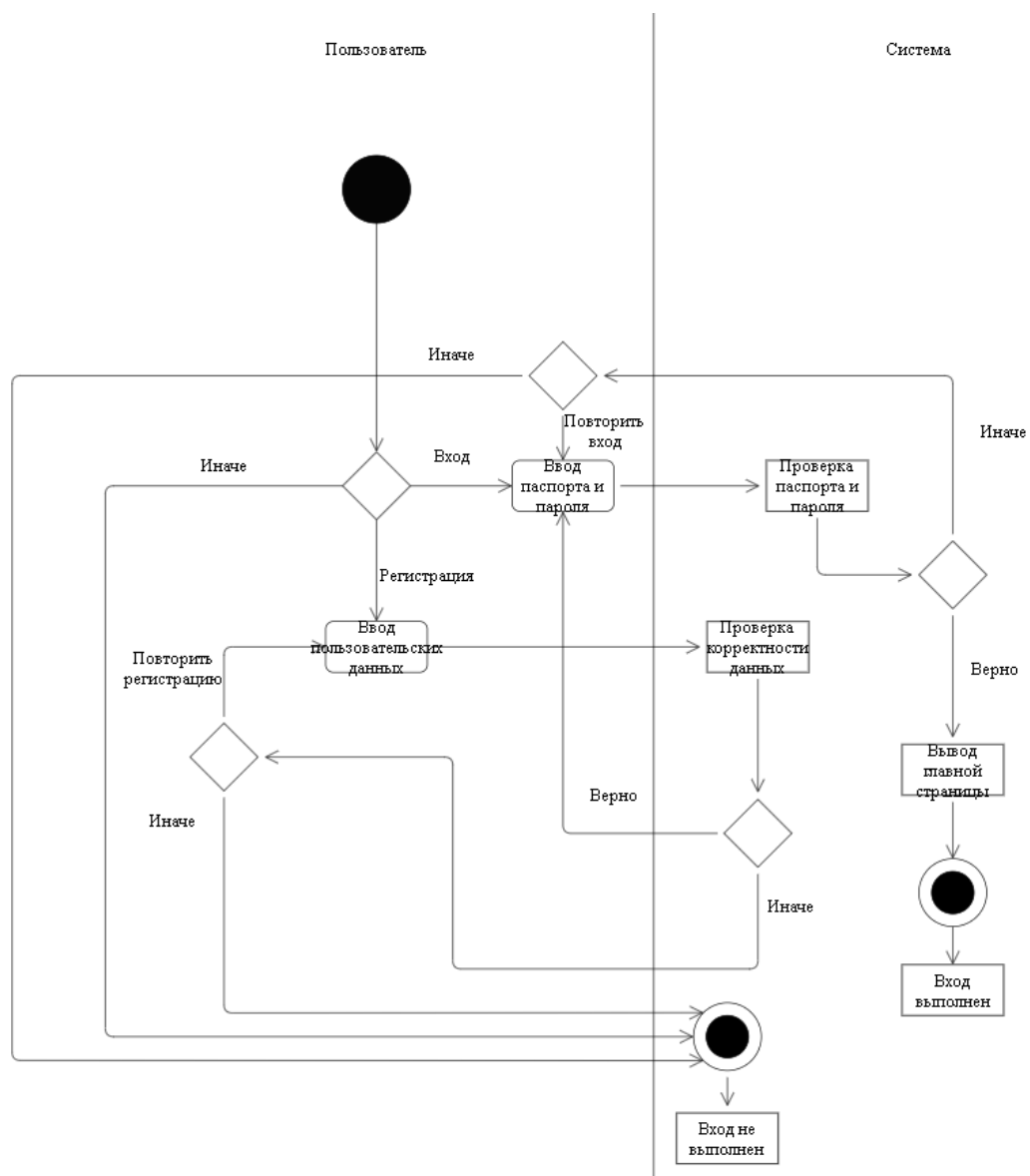


Рисунок 7.3 – Диаграмма деятельности

7.4. Диаграмма состояний

Определяет все возможные состояния, в которых может находиться конкретный объект, а также процесс смены состояний объекта в результате влияния некоторых событий (смотреть рисунок 7.4). Для начала работы с программой необходимо авторизоваться. После Пользователь может получить доступ к отчету для просмотра. Пользователю предоставляется возможность выбрать действия: ничего не делать (завершить работу в приложении), обновить данные. В свою очередь обновление данных можно разделить на добавление, изменение, удаление. После чего происходит возврат на страницу просмотра отчета.

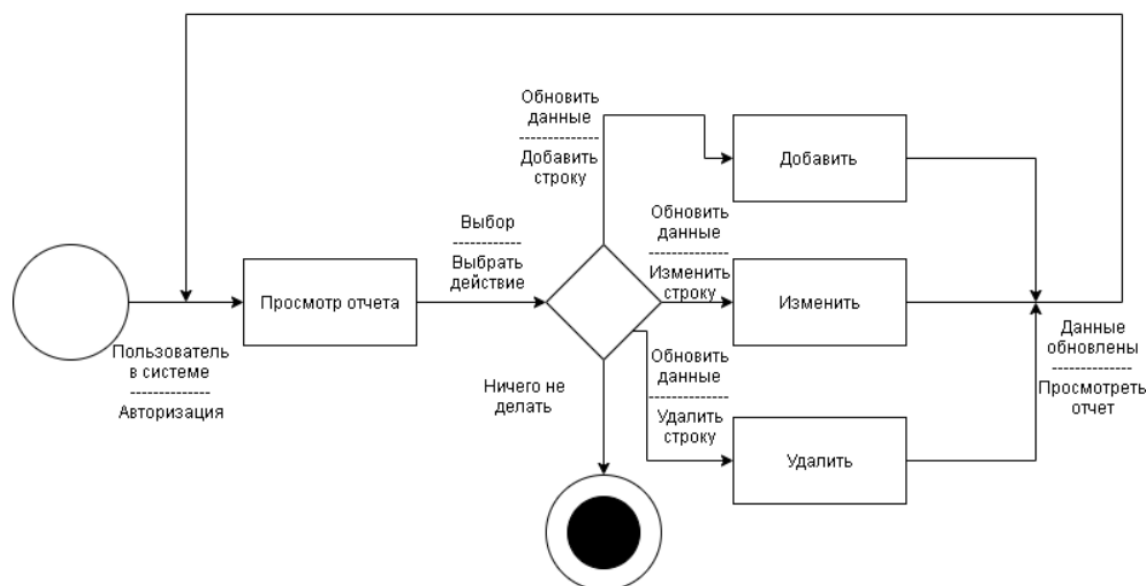


Рисунок 7.4 – Диаграмма состояний

7.5. И/Или структурная схема ПО.

Структурной называют схему, отражающую состав и взаимодействие по управлению частей разрабатываемого программного обеспечения.

Система состоит из пяти подсистем: Движение финансов, Регистрация/Вход, Отчет, Профиль, О нас.

Подсистема Движение финансов предназначена для управления финансов благодаря функциям добавить, удалить, просмотра/изменения записей базы данных.

Действие добавить осуществляет вставку записи о новой операции в базу данных. Действие удалить осуществляет удаление строки из базы данных. Действие просмотр/изменить осуществляет вывод записи из базы данных для дальнейшего редактирования записи.

Подсистема Регистрация/Вход предназначена для регистрации новых пользователей и авторизации существующих благодаря функциям добавить, авторизация.

Действие добавить осуществляет вставку записи о новом пользователе в базу данных. Действие авторизация осуществляет вход в систему и получение доступа к личным данным.

Подсистема Профиль предназначена просмотра и изменения личной информации о пользователе, а также возможности выйти из системы благодаря функциям просмотр/изменение и выход соответственно.

Действие добавить осуществляет вставку записи о новом пользователе в базу данных. Действие просмотр/изменить осуществляет вывод записи из базы данных для дальнейшего редактирования записи. Действие выход позволяет пользователю покинуть систему.

Подсистема Отчет предназначена для управления отчетами с возможностью экспорта за счет функций просмотр, добавить, удалить, экспорт.

Действие добавить осуществляет вставку записи о новом отчете в базу данных. Действие просмотр осуществляет вывод записи из базы данных. Действие удалить

осуществляет удаление строки из базы данных. Действие экспорт сохраняет данные об отчете и загружает в виде файла на устройство.

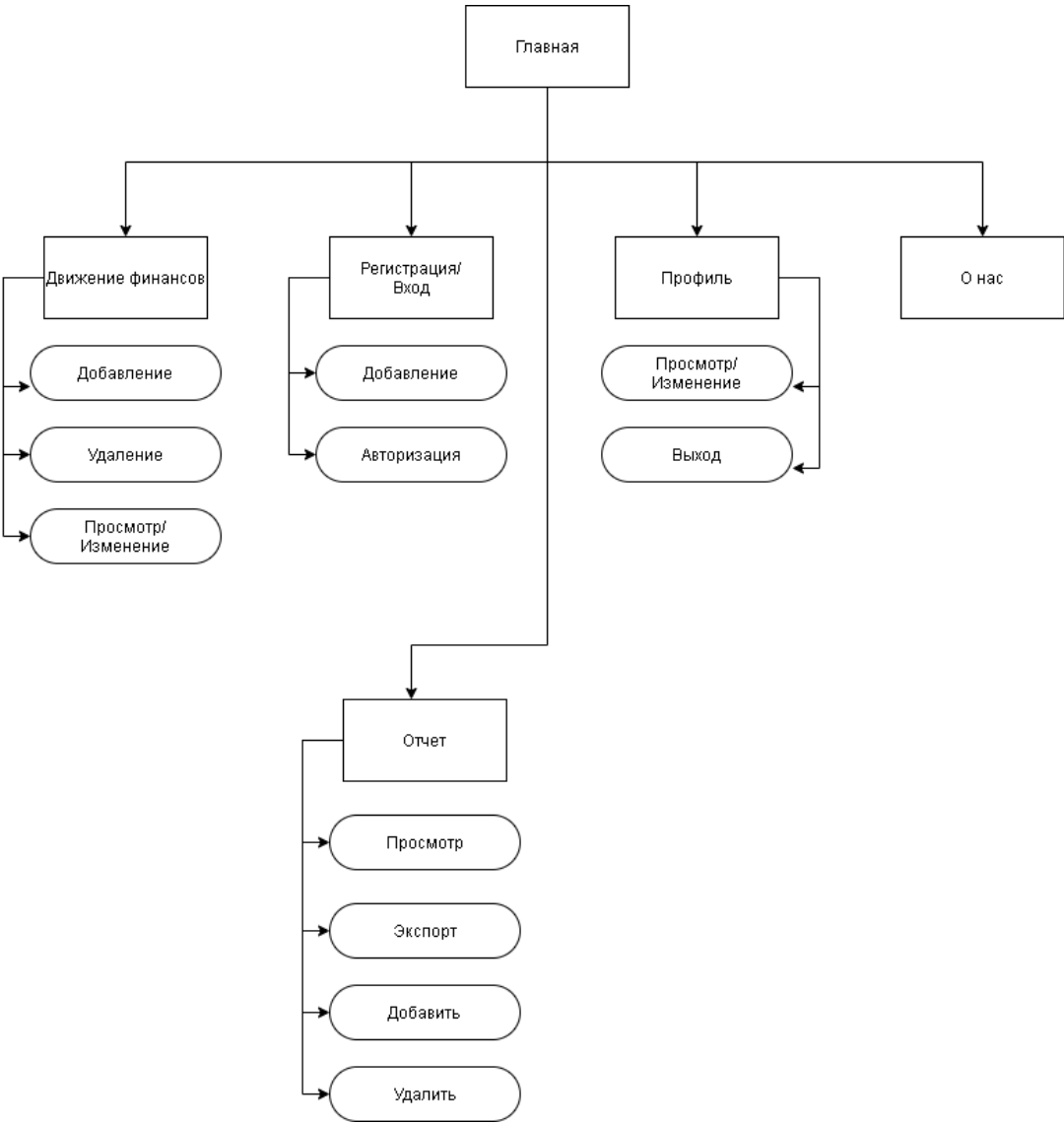


Рисунок 7.5 – Структурная схема ПО

Заключение

В ходе прохождения производственной практики были получены навыки проектирования веб-приложения. Эти навыки включают в себя сбор и анализ исходных данных, определение метрик программного обеспечения, планирование работ по созданию программного обеспечения, определение стадий и этапов разработки программ и документации, определение формы и структуры входных и выходных данных, Составление необходимых диаграмм UML и/или Структурной схемы программы. Изученный материал будет содействовать в разработке проекта.

Список литературы

1. Сайт по самостоятельному изучению веб-разработки URL:
<https://schoolsw3.com/php>
2. Руководство по PHP URL: <https://www.php.net/manual/ru>
3. Простое руководство по диаграммам активности UML URL:
<https://createlly.com/blog/ru/uncategorized-ru/учебник-по-диаграмме-активности/>