

ИНСТИТУТ КОМПЬЮТЕРНЫХ НАУК И КИБЕРБЕЗОПАСНОСТИ
ВЫСШАЯ ШКОЛА ПРОГРАММНОЙ ИНЖЕНЕРИИ

ПРАКТИЧЕСКАЯ РАБОТА №2

«ГОСТ 19.201-78»

по дисциплине «Авторское право, метрология и стандартизация программного обеспечения»

Выполнили студенты группы: 5130904/00104:

Почернин В. С.
Шиляев В. С.
Мурзаканов И. М.
Разукрантов В. Е.

Преподаватель:

Юсупова О. А.

Содержание

1	Постановка задачи	2
2	Теория	3
2.1	Международные стандарты	3
2.2	Российские стандарты	3
2.3	ГОСТ 34	3
2.4	ГОСТ 19	4
2.5	ГОСТ 19.201-78 ЕСПД. Техническое задание. Требования к содержанию и оформлению	5
2.5.1	Общие положения	5
2.5.2	Содержания разделов	6
3	Ход работы	8
3.1	Введение	8
3.1.1	Наименование программы	8
3.1.2	Краткая характеристика области применения программы . .	8
3.2	Основания для разработки	8
3.3	Назначение разработки	8
3.3.1	Функциональное назначение	8
3.3.2	Эксплуатационное назначение	8
3.4	Требования к программе или программному изделию	8
3.4.1	Требования к функциональным характеристикам	8
3.4.2	Требования к надежности	12
3.4.3	Условия эксплуатации	12
3.4.4	Требования к составу и параметрам технических средств . .	12
3.4.5	Требования к информационной и программной совместимости	13
3.4.6	Требования к маркировке и упаковке	13
3.4.7	Требования к транспортированию и хранению	13
3.4.8	Специальные требования	13
3.5	Требования к программной документации	13
3.6	Технико-экономические показатели	14
3.7	Стадии и этапы разработки	14
3.8	Порядок контроля и приемки	14
3.8.1	Виды испытаний	14
3.8.2	Общие требования к приемке работы	15

1 Постановка задачи

Необходимо:

- Составить техническое задание выбранного проекта по ГОСТ 19.201-78.
- Заполнить все разделы.
- Оформить титульный лист по правилам СПбГУ, текст отчета оформить в соответствии с ГОСТ 19.201-78.

2 Теория

2.1 Международные стандарты

- IEEE Std 1063-2001 «IEEE Standard for Software User Documentation» - стандарт для написания руководства пользователя.
- ISO/IEC 26514:2008 «Requirements for designers and developers of user documentation» - стандарт для дизайнеров и разработчиков пользовательской документации.
- IEEE Std 1016-1998 «Recommended Practice for Software Design Descriptions» - стандарт для написания технического описания программы.
- ISO/IEC FDIS 18019:2004 «Guidelines for the design and preparation of user documentation for application software» - стандарт для написания руководства пользователя.

2.2 Российские стандарты

- ГОСТ (государственный стандарт) - это нормативно-правовой документ, в соответствии с требованиями которого производится стандартизация производственных процессов и оказания услуг.
- Государственный стандарт обязательно проходит процедуру регистрации, которая проводится специальным государственным органом - Госстандартом.
- ГОСТы могут заменяться или отменяться. Действующие ГОСТы обязательны к исполнению.
- Утвержденный ГОСТ содержит ключевые требования, которым должны соответствовать товары, работы и услуги, в отношении которых он принимается, для обеспечения их эффективной и безопасной эксплуатации.
- ГОСТ 19 Единая система программной документации (ЕСПД).
- ГОСТ 34 Информационные технологии. Комплекс стандартов на автоматизированные системы.

2.3 ГОСТ 34

- Автоматизированная система (АС) - это система, состоящая из персонала и комплекса средств автоматизации его деятельности, реализующая информационную технологию выполнения установленных функций. В зависимости от вида деятельности выделяют следующие виды АС:
 - Автоматизированные системы управления (АСУ).
 - Системы автоматизированного проектирования (САПР).
 - Автоматизированные системы научных исследований (АСНИ).
 - Другие.
- ГОСТ 34 разделяет виды обеспечения АС:
 - Организационное.
 - Методическое.
 - Техническое.
 - Математическое.
 - Программное обеспечение.
 - Информационное.
 - Лингвистическое.

- Правовое.
- Эргономическое.
- Автоматизированная система - это не программа, а целый комплекс видов обеспечения.
- 34.201-89 Виды, комплектность и обозначения документов при создании автоматизированных систем.
- 34.320-96 Концепции и терминология для концептуальной схемы и информационной базы.
- 34.321-96 Информационные технологии. Система стандартов по базам данных. Эталонная модель управления.
- 34.601-90 Автоматизированные системы. Стадии создания.
- 34.602-89 Техническое задание на создание автоматизированной системы.
- 34.603-92 Информационная технология. Виды испытаний автоматизированных систем.
- РД 50-34.698-90 Автоматизированные системы. Требования к содержанию документов.
- Р ИСО/МЭК 8824-3-2002 Информационная технология. Абстрактная синтаксическая нотация.
- Р ИСО/МЭК 10746-3-2001 Управление данными и открытая распределенная обработка.

2.4 ГОСТ 19

- ЕСПД (единая система программной документации) - комплекс государственных стандартов (ГОСТ), устанавливающих взаимосвязанные правила разработки, оформления и обращения программ и программной документации.
- Стандарты ЕСПД устанавливают требования, регламентирующие разработку, сопровождение, изготовление и эксплуатацию программ.
- Определения из ЕСПД:
 - Программа - данные, предназначенные для управления конкретными компонентами системы обработки информации в целях реализации определенного алгоритма.
 - Программное обеспечение - совокупность программ системы обработки информации и программных документов, необходимых для эксплуатации этих программ.
- 19.001-77 Общие положения.
- 19.005-85 Р-схемы алгоритмов и программ. Обозначения условные и графические и правила выполнения.
- 19.101-77 Виды программ и программных документов.
- 19.102-77 Стадии разработки.
- 19.103-77 Обозначения программ и программных документов.
- 19.104-78 Основные надписи.
- 19.105-78 Общие требования к программным документам.
- 19.106-78 Требования к программным документам, выполненным печатным способом.
- 19.201-78 **Техническое задание. Требования к содержанию и оформлению.**
- 19.202-78 Спецификация. Требования к содержанию и оформлению.

- 19.301-79 Программа и методика испытаний. Требования к содержанию и оформлению.
- 19.401-78 Текст программы. Требования к содержанию и оформлению.
- 19.402-78 Описание программы.
- 19.403-79 Ведомость держателей подлинников.
- 19.404-79 Пояснительная записка. Требования к содержанию и оформлению.
- 19.501-78 Формуляр. Требования к содержанию и оформлению.
- 19.502-78 Описание применения. Требования к содержанию и оформлению.
- 19.503-79 Руководство системного программиста. требования к содержанию и оформлению.
- 19.504-79 Руководство программиста. Требования к содержанию и оформлению.
- 19.505-79 Руководство оператора. Требования к содержанию и оформлению.
- 19.506-79 Описание языка. Требования к содержанию и оформлению.
- 19.507-79 Ведомость эксплуатационных документов.
- 19.508-79 Руководство по техническому обслуживанию. Требования к содержанию и оформлению.
- 19.601-78 Общие правила дублирования, учета и хранения.
- 19.602-78 Правила дублирования, учета и хранения программных документов, выполненных печатным способом.
- 19.603-78 Общие правила внесения изменений.
- 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом.
- 19.701-90 (ИСО 5807-85) Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения.
- 19.781-90 Обеспечение систем обработки информации программное. Термины и определения.

2.5 ГОСТ 19.201-78 ЕСПД. Техническое задание. Требования к содержанию и оформлению

Настоящий стандарт устанавливает порядок построения и оформления технического задания на разработку программы или программного изделия для вычислительных машин, комплексов и систем независимо от их назначения и области применения.

2.5.1 Общие положения

- 1) Техническое задания оформляют в соответствии с ГОСТ 19.106-78 на листах формата 11 и 12 по ГОСТ 2.301-68, как правило, без заполнения полей листа. Номера листов (страниц) проставляют в верхней части листа над текстом.
- 2) Лист утверждения и титульный лист оформляют в соответствии с ГОСТ 19.104-78. Информационную часть (аннотацию и содержание), лист регистрации изменений допускается в документ не включать.

- 3) Для внесения изменений или дополнений в техническое задание на последующих стадиях разработки программы или программного изделия выпускают дополнение к нему. Согласование и утверждение дополнения к техническому заданию проводят в том же порядке, который установлен для технического задания.
- 4) Техническое задание должно содержать следующие разделы:
- Введение.
 - Основания для разработки.
 - Назначение разработки.
 - Требования к программе или программному изделию.
 - Требования к программной документации.
 - Техничко-экономические показатели.
 - Стадии и этапы разработки.
 - Порядок контроля и приемки.

В техническое задание допускается включать приложения. В зависимости от особенностей программы или программного изделия допускается уточнять содержания разделов, вводить новые разделы или объединять отдельные из них.

2.5.2 Содержания разделов

- 1) В разделе «Введение» указывают наименование, краткую характеристику области применения программы или программного изделия и объекта, в котором используют программу или программное изделие.
- 2) В разделе «Основания для разработки» должны быть указаны:
- Документ (документы), на основании которых ведется разработка.
 - Организация, утвердившая этот документ и дата его утверждения.
 - Наименование и (или) условное обозначение темы разработки.
- 3) В разделе «Назначение разработки» должно быть указано функциональное и эксплуатационное назначение программы или программного изделия.
- 4) Раздел «Требования к программе или программному изделию» должен содержать следующие подразделы:
- Требования к функциональным характеристикам.
 - Требования к надежности.
 - Условия эксплуатации.
 - Требования к составу и параметрам технических средств.
 - Требования к информационной и программной совместимости.
 - Требования к маркировке и упаковке.
 - Требования к транспортированию и хранению.
 - Специальные требования.
- (а) В подразделе «Требования к функциональным характеристикам» должны быть указаны требования к составу выполняемых функций, организации выходных и входных данных, временным характеристикам и т. д.
- (б) В подразделе «Требования к надежности» должны быть указаны требования к обеспечению надежного функционирования (обеспечение устойчивого функционирования, контроль входной и выходной информации, время восстановления после отказа и т. д.).

- (с) В подразделе «Условия эксплуатации» должны быть указаны условия эксплуатации (температура окружающего воздуха, относительная влажность и т. д. для выбранных типов носителей данных), при которых должны обеспечиваться заданные характеристики, а также вид обслуживания, необходимое количество и квалификация персонала.
- (d) В подразделе «Требования к составу и параметрам технических средств» указывают необходимый состав технических средств с указанием их основных технических характеристик.
- (e) В подразделе «Требования к информационной и программной совместимости» должны быть указаны требования к информационным структурам на входе и выходе и методам решения, исходным кодам, языкам программирования и программным средствам, используемым программой. При необходимости должен обеспечиваться защита информации и программ.
- (f) В подразделе «Требования к маркировке и упаковке» в общем случае указывают требования к маркировке программного изделия, варианты и способы упаковки.
- (g) В подразделе «Требования к транспортированию и хранению» должны быть указаны для программного изделия условия транспортирования, места хранения, условия хранения, условия складирования, сроки хранения в различных условиях.
- 5) В разделе «Требования к программной документации» должны быть указаны предварительный состав программной документации и, при необходимости, специальные требования к ней.
- 6) В разделе «Технико-экономические показатели» должны быть указаны:
- Ориентировочная экономическая эффективность.
 - Предполагаемая годовая потребность.
 - Экономические преимущества разработки по сравнению с лучшими отечественными и зарубежными образцами или аналогами.
- 7) В разделе «Стадии и этапы разработки» устанавливают необходимые стадии разработки, этапы и содержание работы (перечень программных документов, которые должны быть разработаны, согласованы и утверждены), а также, как правило, сроки разработки и определяют исполнителей.
- 8) В разделе «Порядок контроля и приемки» должны быть указаны виды испытаний и общие требования к приемке работы.
- 9) В приложениях к техническому заданию, при необходимости приводят:
- Перечень научно-исследовательских и других работ, обосновывающих разработку.
 - Схемы алгоритмов, таблицы, описания, обоснования, расчеты и другие документы, которые могут быть использованы при разработке.
 - Другие источники разработки.

3 Ход работы

3.1 Введение

3.1.1 Наименование программы

- Наименование программы на английском языке: «Tesseract».
- Наименование программы на русском языке: «Тессеракт».

3.1.2 Краткая характеристика области применения программы

- Приложение «Тессеракт» предназначено для агрегации цифровых финансовых активов от различных операторов информационных систем.

3.2 Основания для разработки

- Основанием для разработки является Курсовая работа по дисциплине «Технологии разработки качественного программного обеспечения» под руководством старшего преподавателя ВШПИ ИКНТ СПбПУ Маслакова Алексея Павловича.
- Наименование темы разработки - «Агрегатор цифровых финансовых активов «Тессеракт»».

3.3 Назначение разработки

3.3.1 Функциональное назначение

- Функциональным назначением является осведомление пользователей о существовании цифровых финансовых активов от всех существующих операторов информационных систем, а также предоставление возможности автоматического создания портфелей инвестирования по заданным параметрам (стоимость портфеля, уровень рискованности активов в портфеле).

3.3.2 Эксплуатационное назначение

- Эксплуатационным назначением является возможность запуска и использования приложения на мобильных устройствах, работающих на операционной системе **Android**.

3.4 Требования к программе или программному изделию

3.4.1 Требования к функциональным характеристикам

Идентификатор	Требование
F_LoginPage_1	Страница входа содержит логотип приложения, кнопки Войти, Войти с помощью Google, Зарегистрироваться, Информация, поля Логин, Пароль. Dev-версия приложения допускает наличие кнопки Изменить API URL.
F_LoginPage_2	При нажатии на кнопку Информация происходит переход на страницу информации.
F_LoginPage_3	Кнопка Войти неактивна, пока хотя бы одно из полей Логин или Пароль пустое.
F_LoginPage_4	При нажатии на кнопку Войти и возникновении ошибки аутентификации, на экране отображается соответствующее уведомление об ошибке.
F_LoginPage_5	При нажатии на кнопку Войти и корректной аутентификации происходит переход на страницу активов.

Идентификатор	Требование
F_LoginPage_6	При нажатии на кнопку Войти через Google и возникновении ошибки аутентификации, на экране отображается соответствующее уведомление об ошибке.
F_LoginPage_7	При нажатии на кнопку Войти через Google и корректной аутентификации через Google OAuth API происходит переход на страницу активнов.
F_LoginPage_8	При нажатии на кнопку Регистрация происходит переход на страницу регистрации.
F_LoginPage_9	При нажатии на кнопку Изменить API URL открывается окно, в котором можно установить URL сервера.

Таблица 1: Функциональные требования для страницы входа

Идентификатор	Требование
F_InfoPage_1	Страница информации содержит информацию о приложении, кнопку Назад.
F_InfoPage_2	При нажатии на кнопку Назад будет открыт предыдущий экран.

Таблица 2: Функциональные требования для страницы информации

Идентификатор	Требование
F_RegistrationPage_1	Страница регистрации содержит кнопки Назад, Зарегистрироваться, поля Логин, Email, Пароль, Подтверждение пароля.
F_RegistrationPage_2	При нажатии на кнопку Назад будет открыт предыдущий экран.
F_RegistrationPage_3	Кнопка Зарегистрироваться неактивна, пока хотя бы одно из полей Логин, Email, Пароль, Подтверждение пароля пустое.
F_RegistrationPage_4	Длина пароля должна составлять от 6ти до 30ти символов включительно.
F_RegistrationPage_5	Пароль может состоять из цифр, латинских букв верхнего и нижнего регистра, а также специальных символов.
F_RegistrationPage_6	Специальными символами являются: «! @ # \$ % & * () - _ + = ; : , . / ? \ [] { }».
F_RegistrationPage_7	Пароль должен содержать хотя бы одну букву (в любом регистре), а также хотя бы одну цифру или специальный символ (т.е. Буква И (Цифра ИЛИ Специальный символ)).
F_RegistrationPage_8	При нажатии на кнопку зарегистрироваться и возникновении ошибки регистрации, на экране отображается соответствующее уведомление об ошибке.
F_RegistrationPage_9	При нажатии на кнопку зарегистрироваться и успешной регистрации происходит переход на страницу входа, а на экране отображается уведомление об успешной регистрации.
F_RegistrationPage_10	Длина логина должна составлять от 3х до 16ти символов включительно.
F_RegistrationPage_11	Логин может состоять из цифр и латинских букв верхнего и нижнего регистра.
F_RegistrationPage_12	Логин должен быть уникальным.
F_RegistrationPage_13	Email должен содержать имя пользователя, разделитель (@), имя почтового сервера.

Идентификатор	Требование
F_RegistrationPage_14	Email должен быть уникальным.
F_RegistrationPage_15	Поля Пароль и Подтверждение пароля должны совпадать.

Таблица 3: Функциональные требования для страницы регистрации

Идентификатор	Требование
F_AssetsPage_1	Страница активов содержит список всех цифровых активов.
F_AssetsPage_2	Каждый элемент списка (актив) содержит базовую информацию об активе (название актива, название компании, стоимость актива, изменение стоимости актива за последний день).
F_AssetsPage_3	На каждом элементе списка (активе) есть кнопка В избранное, позволяющая добавить/удалить актив в избранное.
F_AssetsPage_4	При нажатии на актив открывается страница конкретного актива.
F_AssetsPage_5	За один раз на страницу загружается 10 элементов списка, которые затем могут подгружаться по мере его прокручивания (динамическая загрузка).

Таблица 4: Функциональные требования для страницы активов

Идентификатор	Требование
F_AssetPage_1	Страница актива содержит в себе кнопки Назад, В избранное, а также полную информацию об активе (базовая информация + уровень рискованности + описание компании + описание актива + оператор ЦФА).
F_AssetPage_2	При нажатии на кнопку Назад будет открыт предыдущий экран.
F_AssetPage_3	При нажатии кнопки В избранное происходит добавление/удаление актива в избранное.

Таблица 5: Функциональные требования для страницы конкретного актива

Идентификатор	Требование
F_FavouritesPage_1	Страница избранных активов полностью повторяет страницу активов, за исключением того, что там хранятся только избранные активы пользователя.

Таблица 6: Функциональные требования для страницы избранных активов

Идентификатор	Требование
F_PortfoliosPage_1	Страница портфелей содержит в себе кнопку Создать портфель, а также список портфелей пользователя.
F_PortfoliosPage_2	Каждый элемент списка (портфель), содержит дату и время создания, уровень рискованности, текущую стоимость, а также изменение стоимости с момента создания портфеля.
F_PortfoliosPage_3	При нажатии на кнопку Создать портфель происходит переход на экран создания портфеля.
F_PortfoliosPage_4	При нажатии на элемент списка (портфель) происходит переход на страницу конкретного портфеля.

Идентификатор	Требование
F_PortfoliosPage_5	За один раз на страницу загружается 10 элементов списка, которые затем могут подгружаться по мере его прокручивания (динамическая загрузка).

Таблица 7: Функциональные требования для страницы портфелей

Идентификатор	Требование
F_PortfolioCreatePage_1	Страница создания портфеля содержит в себе кнопки Назад, Создать портфель, поле Общая стоимость, радио-кнопку Уровень рискованности.
F_PortfolioCreatePage_2	При нажатии на кнопку Назад будет открыт предыдущий экран.
F_PortfolioCreatePage_3	Кнопка Создать портфель неактивна, пока поле Стоимость пустое.
F_PortfolioCreatePage_4	Стоимость должна быть целым числом, не меньшим, чем минимальная стоимость актива выбранной рискованности и не большим, чем 10_000_000.
F_PortfolioCreatePage_5	Выбор рискованности является радио-кнопкой, состоящей из выборов Высокорискованный, Среднерискованный, Низкорискованный, Комбинированный.
F_PortfolioCreatePage_6	При нажатии на кнопку Создать портфель и возникновении ошибки, на экране отображается соответствующее уведомление об ошибке.
F_PortfolioCreatePage_7	При нажатии на кнопку Создать портфель и корректном создании, происходит переход на страницу портфелей.
F_PortfolioCreatePage_8	Портфель создается на основе данных, введенных пользователем, а также данных об активах с помощью специального алгоритма.

Таблица 8: Функциональные требования для страницы создания портфеля

Идентификатор	Требование
F_PortfolioPage_1	Страница портфеля содержит в себе кнопку Назад, а также полную информацию о портфеле (Дата и время создания, Уровень рискованности, Общая стоимость, Изменение стоимости с момента создания портфеля, Список всех цифровых активов портфеля с указанием общей стоимости, изменения стоимости с момента создания портфеля и количества каждого актива).
F_PortfolioPage_2	При нажатии на кнопку Назад будет открыт предыдущий экран.
F_PortfolioPage_3	При нажатии на какой-либо актив будет совершен переход на страницу конкретного актива.

Таблица 9: Функциональные требования для страницы конкретного портфеля

Идентификатор	Требование
F_SettingsPage_1	Экран настроек содержит в себе кнопки Изменить пароль, кнопка с диалоговым окном Тема, поля Старый пароль, Новый пароль, Подтверждение пароля, Выйти из аккаунта.

Идентификатор	Требование
F_SettingsPage_2	Кнопка изменить пароль неактивна, если хотя бы одно из полей Старый пароль, Новый пароль, Подтверждение пароля пустое, а также если пользователь вошел через Google.
F_SettingsPage_3	При нажатии на кнопку Изменить пароль и ошибки изменения пароля, на экране отображается соответствующее уведомление об ошибке.
F_SettingsPage_4	При нажатии на кнопку Изменить пароль и успешном изменении пароля, на экране отображается соответствующее уведомление.
F_SettingsPage_5	При нажатии на кнопку Переключить тему появляется диалоговое окно с предложением выбора темы (Системная, Темная, Светлая).
F_SettingsPage_6	При нажатии на кнопку выйти из аккаунта происходит выход из аккаунта и переход на страницу входа.
F_SettingsPage_7	Старый пароль должен совпадать с текущим паролем пользователя.
F_SettingsPage_8	Поля Новый пароль и Подтверждение пароля должны совпадать.

Таблица 10: Функциональные требования для страницы настроек

Идентификатор	Требование
F_Navigation_1	Между страницами активов, избранных активов, портфелей, настроек навигация происходит с помощью нижнего меню.
F_RiskDegree_1	Каждый актив имеет свой уровень рискованности, рассчитываемый на основе информации об эмитенте и об активе с помощью специального алгоритма.

Таблица 11: Остальные функциональные требования

3.4.2 Требования к надежности

- Сервер, обрабатывающий бизнес-логику приложения должен обладать высокой доступностью уровня 99% (two nines), то есть быть недоступным для обработки запросов не более 3.65 суток в год.
- Клиентское приложение должно корректно обрабатывать возможные ошибки, не завершая аварийно свою работу.

3.4.3 Условия эксплуатации

Специальные условия эксплуатации не требуются. Обслуживание не требуется.

3.4.4 Требования к составу и параметрам технических средств

Требования к серверу:

- Поддержка виртуализации,
- Доступ в интернет,
- Программное обеспечение: Docker.

Требования к устройству клиента:

- Мобильное устройство на операционной системе Android,

- Версия Android SDK не менее 26 (Android 8.0),
- Доступ в интернет,
- Актуальная версия сервисов Google (опционально, используются для входа через аккаунт Google).

3.4.5 Требования к информационной и программной совместимости

Серверное приложение:

- Язык программирования – Java,
- Веб-фреймворк – Spring,
- СУБД – PostgreSQL.

Клиентское приложение:

- Язык программирования – Kotlin,
- Версия Android SDK не менее 26 (Android 8.0),
- Актуальная версия сервисов Google (опционально, используются для входа через аккаунт Google).

3.4.6 Требования к маркировке и упаковке

Клиентское приложение собирается в APK-архив для дальнейшего распространения и установки на мобильные устройства. Маркировка не требуется.

3.4.7 Требования к транспортированию и хранению

Специальные требования к транспортировке и хранению не предъявляются.

3.4.8 Специальные требования

- Клиентское приложение должно поддерживать следующие языки: русский, английский;
- Пользовательский интерфейс приложения должен быть разработан в соответствии с дизайн-системой Material 3.

3.5 Требования к программной документации

Предварительный состав программной документации:

- функциональная спецификация,
- high-level design,
- спецификация Application Programming Interface,
- описание используемых алгоритмов,
- план модульного тестирования,
- план интеграционного тестирования,
- план end-to-end тестирования,
- код программы.

Функциональная спецификация должна содержать функциональные и нефункциональные требования к разрабатываемому приложению. В документации High-Level Design должны быть представлены: макет дизайна интерфейса приложения, архитектура приложения, стек используемых технологий для frontend и backend частей приложения, диаграмма классов, схема базы данных и API. Все схемы должны сопровождаться описанием.

3.6 Техничко-экономическис показатели

Ориентировочная экономическая эффективность приложения: Учитывая, что данное приложение разрабатывается в рамках учебного проекта, ориентировочная экономическая эффективность не только заключается в финансовых показателях, но и в педагогических целях. Процесс разработки приложения предоставляет студентам возможность практически применить полученные знания и навыки в области разработки Android приложений. В качестве дополнительной экономической выгоды можно рассматривать возможность привлечения инвестиций на будущие проекты или участие в конкурсах и выставках студенческих проектов.

Предполагаемая годовая потребность: Годовая потребность в использовании учебного приложения будет зависеть от его актуальности и востребованности среди студентов и преподавателей. Предполагается, что в первый год эксплуатации приложения оно будет использоваться студентами из различных курсов и направлений, что может привести к формированию постоянной аудитории в последующие годы.

Экономические преимущества разработки по сравнению с лучшими отечественными и зарубежными образцами или аналогами: Одним из главных экономических преимуществ учебного проекта является возможность минимизировать затраты на разработку приложения благодаря использованию свободных или студенческих лицензий для программного обеспечения и инструментов разработки. Кроме того, такой проект предоставляет возможность для студентов проявить свои навыки и творческий потенциал, что в долгосрочной перспективе может привести к их успешной карьере в сфере информационных технологий.

3.7 Стадии и этапы разработки

Проект имел следующие стадии разработки:

- Разработка функциональной спецификации - Требуется сформулировать функциональные и нефункциональные требования на разрабатываемый программный продукт.,
- Разработка High-Level Design(HLD) - Требуется разработать High-Level Design (HLD). В документе должны быть описаны макет дизайна интерфейса (при наличии), общая архитектура приложения, стек технологий, спроектированы диаграммы классов/модулей/etc, схемы баз данных, API. Все схемы должны сопровождаться описанием.,
- Разработка программного продукта - Необходимо разработать программный продукт, опираясь на FS и HLD.,
- Доработка программного продукта - Необходимо доработать программный продукт, в том числе:
 - Произвести рефакторинг кода.
 - Протестировать работу программного продукта в строгом соответствии со всеми пунктами FS.
 - Добавить новые фичи.
- Покрыть код модульными тестами - Необходимо покрыть код приложения модульными тестами. Покрытие кода должно составлять не менее 80 процентов.,
- Покрыть код интеграционными тестами,
- Покрыть код системными тестами,

3.8 Порядок контроля и приемки

3.8.1 Виды испытаний

Проект проверялся следующими видами испытаний:

- Необходимо выполнить модульное тестирование ПО, покрытие должно составлять более 80
- Необходимо выполнить интеграционное тестирование нескольких модулей. Минимальное количество сценариев - 10, сценарии должны быть согласованы с руководителем.
- Необходимо протестировать ПО по бизнес-требованиям, сформированным перед началом проектирования ПО. В качестве тестовых сценариев выбираются основные сценарии использования ПО в полностью рабочем окружении.

3.8.2 Общие требования к приемке работы

- Обязательно наличие репозитория на любом хостинге(github/gitlab/bitbucket/etc). Репозиторий ведется согласно github flow/gitlab flow/git flow(По выбору команды).
- Система контроля за разработкой, где есть возможность создать kanban-доску - любая из Github Projects, Trello, Notion