МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ им. А. И. ГЕРЦЕНА»



**ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И   
ТЕХНОЛОГИЧЕСКОГО ОБРАЗОВАНИЯ**

**Кафедра информационных технологий и электронного обучения**

Основная профессиональная образовательная программа

Направление подготовки 09.03.01 Информатика и вычислительная техника

Направленность (профиль) «Технологии разработки программного обеспечения»

форма обучения – очная

**Курсовая работа**

«Разработка программного проекта сервиса для управления задачами»

Обучающейся 3 курса

Беленко Анастасии Витальевны

Научный руководитель:

Кандидат физико-математических наук,   
доцент кафедры ИТиЭО

Жуков Николай Николаевич

Санкт-Петербург

2021

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 3](#_heading=h.30j0zll)

[ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 4](#_heading=h.1fob9te)

[1.1 Определение понятий 4](#_heading=h.3znysh7)

[1.2 Различные модели и методологии 4](#_heading=h.2et92p0)

[1.3 Инструменты и фреймворки гибкой методологии разработки 7](#_heading=h.tyjcwt)

[ПРАКТИЧЕСКАЯ ЧАСТЬ 8](#_heading=h.3dy6vkm)

[2.1 Техническое задание 8](#_heading=h.1t3h5sf)

[2.2 Диаграмма Ганта 9](#_heading=h.4d34og8)

[2.3 Макет 10](#_heading=h.2s8eyo1)

[2.4 Инструмент управления проектом 11](#_heading=h.17dp8vu)

# ВВЕДЕНИЕ

На первых этапах создания проекта любой человек задумывается об эффективном распределении своего времени и времени своих коллег по проекту. Вообще создание проекта – это ресурсо- и времязатратно, поэтому проблема правильного описания всех процессов, рационального выставления сроков выполнения каждого этапа остаётся актуальной. В данной курсовой работе я описываю основные методы разработки программных проектов, выделяю их достоинства и недостатки и применяю выбранный мной метод на практике – при создании моего проекта «разработка приложения для трекинга привычек».

# ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

## 1.1 Определение понятий

Процесс разработки программного продукта – это процесс преобразования потребностей пользователей в программный продукт. От правильного понимания запросов пользователей зависит успешность проекта. На основе этого запроса выстраивается техническое задание – документ или документы, определяющие цель и структуру любого проекта. Техническое задание на программный продукт составляется строго в соответствии с ГОСТ 19.201–78 «Единая система программной документации. Техническое задание. Требования к содержанию и оформлению.» и ГОСТ 34.602-89 «Информационная технология (ИТ). Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы.». Процесс разработки разделяют на отдельные фазы, из которых складывается жизненный цикл разработки ПО (англ. SDLC – Software development lifecycle). Прохождение ступеней этого цикла зависит от выбранной методологии разработки – это система принципов, а также совокупность идей, понятий, методов, способов и средств, определяющих стиль разработки программного обеспечения.

## 1.2 Различные модели и методологии

Для эффективного управления проектом руководителю стоит выбрать одну из моделей разработки основываясь на различных факторах и ограничениях. Например, при возможности выставить чёткие сроки и окончательно утвердить требования проекта руководители останавливают свой выбор на каскадной или V-образной моделях. Каскадная модель используется в проектах с чёткими датами, утверждёнными требованиями и заранее установленной точной стоимости всех работ. Данная модель подразумевает под собой последовательное прохождение всех этапов разработки – к уже пройденным этапам нельзя вернуться, следующий наступит только после полного завершения текущего, а для повторного прохождения необходимо завершить весь проект, в т.ч. этап тестирования и поддержки (рис. 1). Довольно сильным «плюсом» здесь является именно фиксированная стоимость, она в большинстве случаем перевешивает недостатки этого подхода.

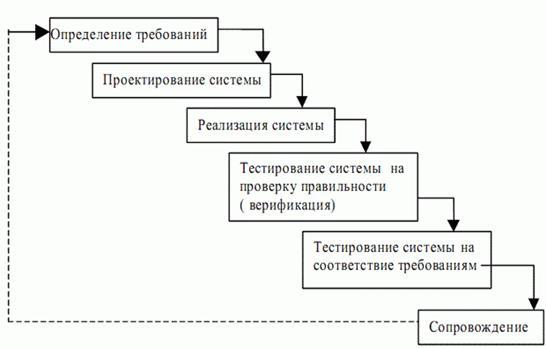


Рисунок 1. Графическое представление каскадной модели разработки.

«V-model» позаимствовала последовательное прохождение у каскадной модели. Отличительной чертой данного подхода является то, что этапу тестирования выделяется значительное время в разработке проекта. V-образная модель применяется в проектах, где важно непрерывное функционирование ПО.

Если же у проекта сжатые сроки реализации продукта, ограниченный бюджет и нечётко определённые требования к продукту, то для такого проекта отлично подойдёт методология RAD (от англ. rapid application development — быстрая разработка приложений) – это подход к разработке программного обеспечения в рамках спиральной модели жизненного цикла. Жизненный цикл программного обеспечения делится на фазы. RAD предлагает вести разработку так, чтобы заказчик мог увидеть практические результаты на самых ранних этапах — и скорректировать техническое задание, если это будет необходимо. Очередной цикл разработки начинается не раньше, чем пользователь оценил результаты предыдущего. Благодаря частому взаимодействию пользователя и разработчика снижаются риски допущения ошибок и недопониманий, переделываний с нуля и т.п. Преимущества такой методологии: интерфейс, устраивающий пользователя, лёгкая адаптируемость проекта к изменяющимся требованиям, быстрота продвижения программного продукта на рынок. Недостатки: сильная зависимость от вовлечённости пользователя в проект, жёсткие временные ограничения, отсутствие полной документации на начальных этапах разработки, необходимость в высококвалифицированных разработчиках для работы в сжатые сроки.

В иных ситуациях прибегают к методу «гибкого программирования» (Agile). В отличие от моделей, рассмотренных выше, разработка по этому методу будет делится на итерации. После каждой итерации выпускается новая версия ПО. Каждая последующая версия шире и с устранёнными ошибками предыдущих версий.

Основное преимущество такого подхода в том, что внесение изменений не является такой существенной проблемой, как в каскадной модели, т.е. разработчики ещё на первых версиях могут выявить и исправить ошибки. Также достоинством будут являться ранние релизы, так как с их помощью становится возможным собрать и проанализировать обратную связь, и на основе этого анализа улучшить следующую версию ПО.

У Agile есть и недостатки перед другими методами. Так как метод не подразумевает составление чётких и полных требований в начале разработки, проблемой становится оценка необходимого количества усилий, средств и ресурсов. И по той же причине довольно сложно составить документацию по разрабатываемому ПО, а это создаёт трудности для новых пользователей и коллег, которые только пришли в проект и не могут знать всех особенностей.

## 1.3 Инструменты и фреймворки гибкой методологии разработки

Фреймворк для управления проектами — это набор инструментов, задач и процессов, используемых для организации и выполнения проекта от начала и до завершения. Фреймворк описывает все, что вам нужно для успешного планирования ваших проектов, контроля и управления ими. Agile-фреймворки отличаются от традиционных инструментов простотой, так как сводят к минимуму объёмы правил и документации.

Самым популярным фреймворком гибкой разработки является методология Scrum, разработанная в 1990-х годах. Она строится на идее разбиения проекта на спринты от одной до четырёх недель. Каждый спринт заканчивается созданием работоспособной версии черновика продукта. Также Scrum требует определенных ролей и обязанностей для участников Agile-команды: владелец продукта (выступает в роли клиента, заказчика), scrum-мастер (ответственен за организацию ежедневных совещаний, улучшение взаимодействия в команде и повышение продуктивности) и команда разработки (включает в себя всех сотрудников, необходимых для проектирования, производства, тестирования и выпуска финального продукта). Все вместе они scrum-команда. Scrum удобен для сложных проектов, требующих дисциплины и навыков, проектов по разработке ПО, требования к которому могут изменятся непосредственно во время разработки.

# ПРАКТИЧЕСКАЯ ЧАСТЬ

## 2.1 Техническое задание

Техническое задание – технический документ, оговаривающий набор требований к системе и утверждённый как заказчиком (пользователем), так и исполнителем (производителем) системы. Такая спецификация может содержать также системные требования и требования к тестированию.

Техническое задание позволяет исполнителю понять суть задачи, спланировать выполнение проекта, а заказчику сформировать чёткое представление о готовом продукте и даёт основание требовать от исполнителя соответствия продукта всем условиям, оговорённым в техническом задании.

Как уже было сказано выше, техзадание составляется в соответствии ГОСТов. После рассмотрения данных стандартов составим требования к системе:

1) Функциональным назначением программы является предоставление пользователю возможности отслеживания необходимых действий в течение нескольких дней.

Программа должна эксплуатироваться на устройствах с операционной системой Android. Пользователями должны являться любые заинтересованные в данной программе люди, не обязанные обладать какими-либо специальными навыками.

2) Требования к структуре: результатом должно являться мобильное приложение, устанавливаемое из apk-файла, написанное на совместимом с данной ОС языке.

3) Требования к функционалу: начальное окно приветствия при первом запуске приложения, ознакомление с базовыми функциями. Основное окно с текущими отслеживаниями и дополнительное меню. Добавление новых "привычек", интеграция календаря с возможностью отмечать дни и время. Пользователь должен иметь возможность выставлять "цель" - это количество "успешных" дней подряд. Включение напоминаний по определённым дням и во указанное время.

4) Требования к совместимости: приложение должно работать на операционной системе Android 5.0+.

## 2.2 Диаграмма Ганта

После составления технического задания стоит распределить время, которое выделяется на проект.

Диаграмма Ганта (график Ганта) – это один из наиболее удобных и популярных способов графического представления времени выполнения задач. Как средство планирования используется в личном и корпоративном тайм-менеджменте; управлении проектами.

Для составления диаграммы Ганта я использовала сервис Instagantt, потому что данный сервис имеет возможность интеграции с Asana.

Я выделила несколько основных этапов работы над проектом: сбор нужной информации, составление техзадания, создание макетов, заполнение диаграммы Ганта, разработка минимального функционала, настройка интерфейса и отрисовывание кнопок, тестирование функций и рефакторинг ТЗ, реализация авторизации, доработка функционала, финальное тестирование и релиз.

Полученная диаграмма представлена на рисунке 2.

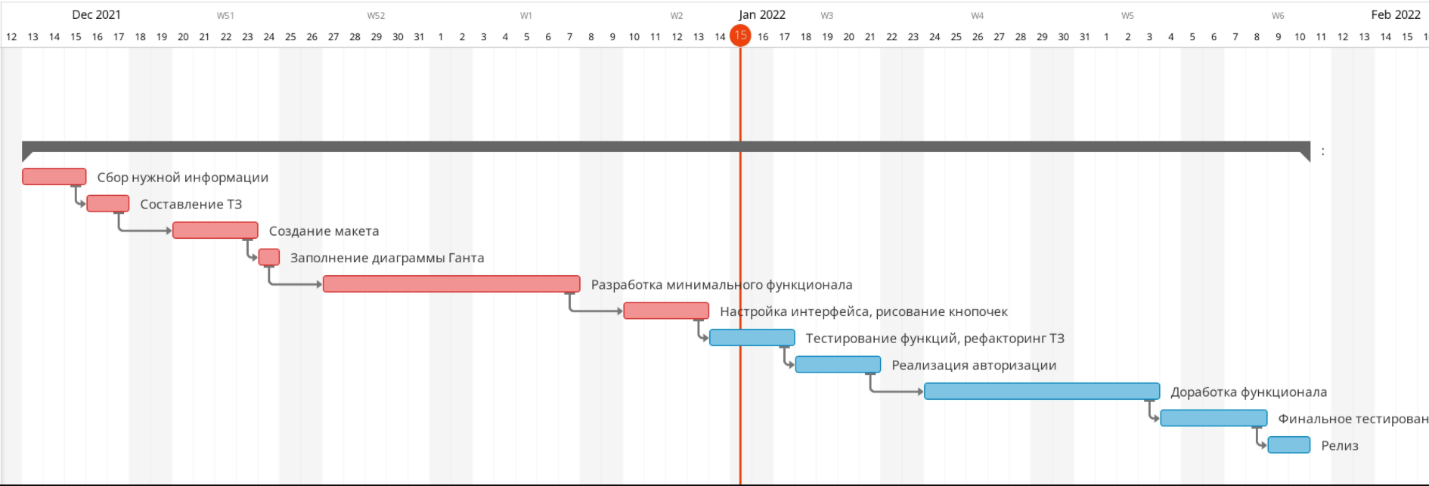


Рисунок 2. Диаграмма Ганта по проекту «Трекер привычек»

## 2.3 Макет

Дизайн-макет сайта — это визуальный образ будущего сайта, разработанный с учетом технических возможностей HTML верстки. Такой макет является демонстрацией того, как визуально будет выглядеть сайт после верстки и наполнения.

Я делала макеты в Photoshop из-за привычки использования графических редакторов вместо специализированных утилит. На рисунке 3 представлена пара макетов главного экрана и экрана отслеживания.

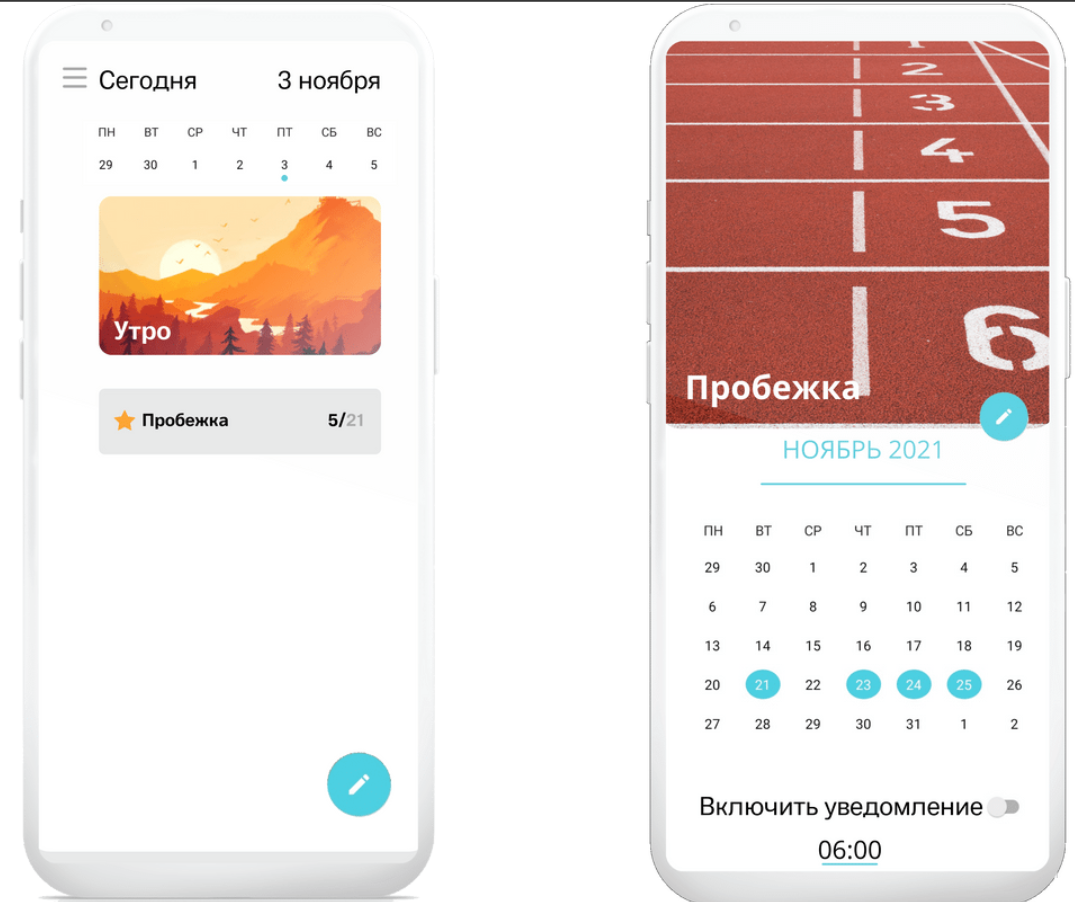


Рисунок 3. Макет главного экрана и экрана отслеживания «трекинга привычек»

## 2.4 Инструмент управления проектом

Так как проект подразумевает разработку мобильного приложения на операционной системе Android, выбор пал на Android Studio, а для управления версиями я выбрала Git. Код приложения будет хранится в репозитории GitHub, так как этот сервис самый удобный и изученный.

Во время выполнения одной из самостоятельных работ я подробно изучила сервис Asana, который показался мне удобным для данного проекта. Asana объединяет в себе широкий функционал, для проекта наиболее полезными из них станут сборка ресурсов и источников полезной информации по проекту, отслеживание ход выполнения задач и в виде доски, и в виде хронологии.

## 2.5 UML- диаграмма

Диаграмма вариантов использования – диаграмма, отражающая отношения между акторами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Прецедент — возможность моделируемой системы (часть её функциональности), благодаря которой пользователь может получить конкретный, измеримый и нужный ему результат.

Актор - множество логически связанных ролей в UML, исполняемых при взаимодействии с прецедентами или сущностями.

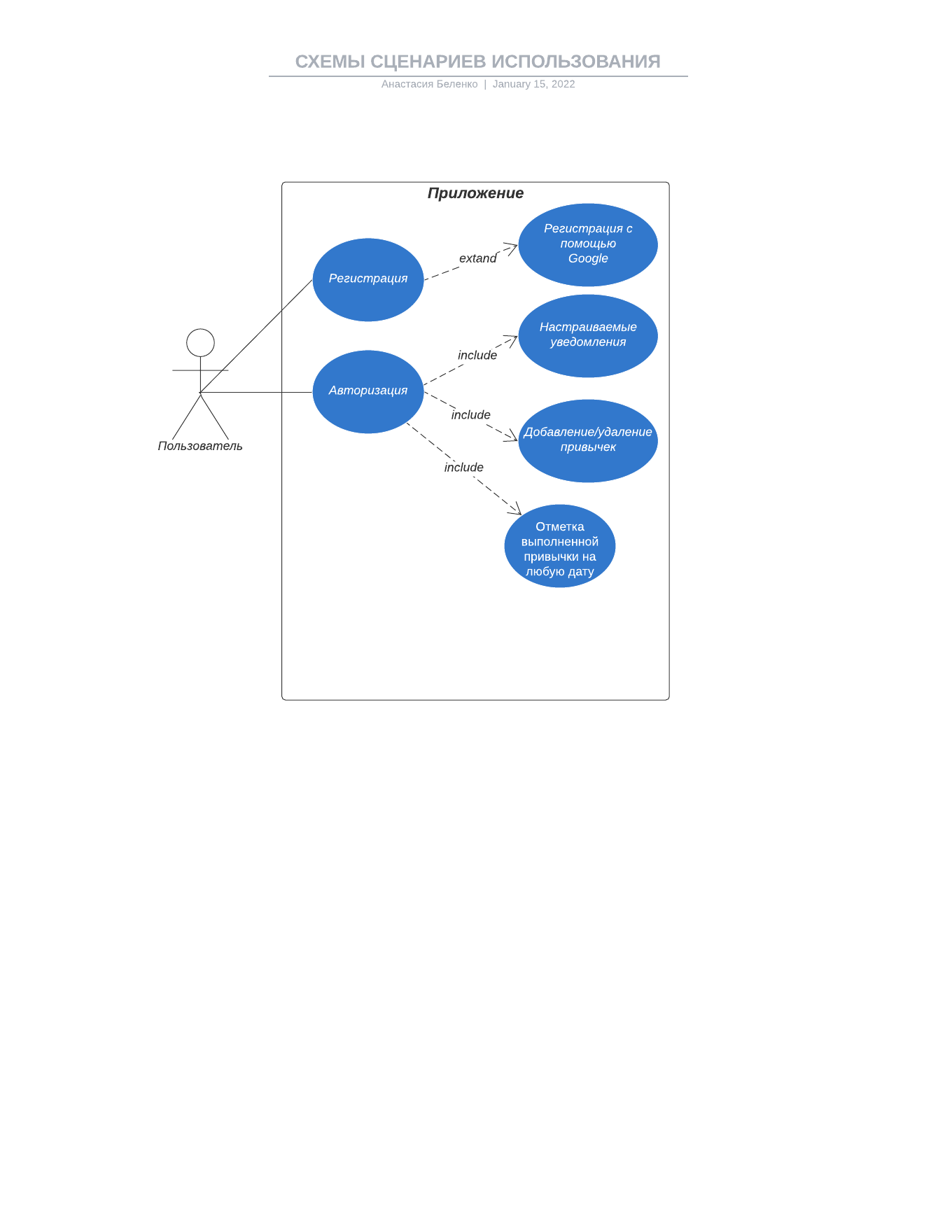


Рисунок 4. Диаграмма вариантов использования приложения

Опишу последовательность действий пользователя. При первом входе в приложение выходит приветственное меню, помогающее пользователю в регистрации и/или авторизации. После успешной авторизации пользователь видит всплывающие подсказки, объясняющие минимальный функционал приложения: добавление/удаление привычки, отметка выполненной привычки и установка уведомлений на конкретное время. При добавлении привычки она появляется на главном экране.

## 2.6 Тестирование

Юзабилити-тестирование (Usability – удобство) – это проверка программного продукта на соответствие с требованиями в плане удобности использования приложения. Таким образом, с помощью юзабилити-тестирования мы можем определить эргономичность программы.

Проверка юзабилити приложения заключается в:

* Оценке соответствия дизайна приложения к его функциональности, заданной заказчиком.
* Анализе используемых графических элементов, цветового оформления с точки зрения восприятия.
* Оценке удобства навигации и ссылочной структуре.
* Анализе текстового наполнения сайта.
* Оценке удобства использования функциями приложения (сервисами, если это сайт).
* Анализе шрифтового оформления текста.