Техническое задание

На разработку мобильного приложения Социальной сети

Составлено на 34 страницах, включая Приложение А

Составитель Иванцов И.П.

/ /

/ /

/ /

Москва, 2023

1. Термины и определения

**API (*Application Programming Interface*) –** набор программных модулей и протоколов, позволяющих Серверу и Приложению обмениваться данными.

**GIF-анимация** – файл в формате GIF, являющимся преобразованным фрагментом видеоролика, доступным для использования в социальных сетях

**Push-уведомления** – способ доставки сообщений от сервера клиенту по инициативе сервера для отображения в Приложении уведомлений Пользователю.

**Rest** ***(Representational State Transfer)*** – метод клиент-серверного взаимодействия, при котором вызов удаленной процедуры представляет собой HTTP-запрос, а необходимые данные передаются в качестве параметров запросов

**Авторизация** – предоставление Пользователю прав на использование ресурсов Приложения в соответствии с его статусом и на основе данных, полученных при Регистрации

**Комментарий** – короткая текстовая заметка, не содержавшая изображений или видеороликов, размещаемая пользователями в отношении Поста.

**Комплекс** – информационная система, построенная по модели клиент-сервер, состоящая из совокупности Мобильного Приложения и Управляющего Сервера совместно с Панелью Управления.

**Мобильное приложения** ***(Приложение)*** – программное обеспечение, предназначенное для работы на мобильных устройствах: смартфонах и планшетах. Мобильное Приложение является неотъемлемой частью Комплекса и выступает в роли клиента, обращающегося к серверу для получения различных информационных материалов и инструкций по работе. В данном ТЗ под Мобильным Приложением подразумевается приложение Kaplys для IOS и Android.

**ПУ** ***(Панель Управления)*** – информационная система, графическая настройка над Сервером, предоставляющая возможность удобного ручного управления содержимым Приложения и настройками Сервера.

**Регистрация** – процедура создания пользователям учетной записи в Приложении.

**Социальная сеть** – сторонний веб-сайта в Интернете, предназначенных для публикации различной информации пользователя данной социальной сети. Приложения интегрируются с Социальными сетями, указанными в данном ТЗ.

**Тег** – текстовая метка, присеваемая Посту для облегчения поиска и навигации в Приложении. Пост может обладать одним или нескольким тегами, а также не иметь тегов.

**Пост** – запись, создаваемая и размещенная Пользователям в Приложении, доступная для просмотра другими пользователями, состоящая из графического изображения или видеороликов, сопровождающего тексом и ряд дополнительных параметров.

**Управляющий Сервер *(Сервер)*** –программно-аппаратный комплекс, обеспечивающий работу Комплекса, размещаемый у хостинг-провайдера. Сервер выступает в роли программного обеспечения, дающего возможность управлять содержимым Приложения на всех мобильных устройствах удаленно.

**Каскадная модель** ***(Waterfal*l)** – модель жизненного цикла ПО все этапы зависят друг от друга. Этап начинается тогда, когда закончился предыдущий – все его артефакты полностью готовы. При такой модели движение к цели выглядит как каскадный водопад.

**Артефакт** – это рукотворный объект. В первом приближении артефактами можно назвать и код программы, и документы написанные программистами. При более тщательном подходе мы можем значительно расширить круг примеров и отнести к артефактам электронные письма, фрагменты чатов и даже наброски фломастеров на доске.

**Программное обеспечение** ***(ПО)*** – программа или множество программ, используемых для управления компьютером или иных аппаратных обеспечений.

**Частное техническое задание** ***(ЧТЗ)*** – Система состоящая из нескольких подсистем, то в основном ТЗ следует привести перечень функций для подсистемы, описывающая функциональные требования к подсистемам в отдельных ТЗ на подсистемы.

**Пользовательские требования** – декомпозируется на какое-то число функциональных. К каждому функциональному требованию нужно продумать нефункциональные требования и ограничения. Также нефункциональные требования и ограничения могут напрямую вытекать как из пользовательских требований, так и из бизнес-требований и правил.

**Тред** – последовательность ответов на сообщение, то есть **“ветвь обсуждения”**. Сообщения показываются в виде связанной последовательности, если их объединяет общая тема или общий идентификатор ветви.

1. Анализ
   1. Назначение данного Технического задания

Основное назначение настоящего Технического задания – определение объема работ по разработке Социальной сети и согласование его с Клиентом. Этот документ должен также поставить четкие задачи в соответствие документации разработчика проекта. Документ описывает границы, внешний вид, поведение Мобильной Приложения.

* 1. Обязательность выполнения положений данного Технического задания

Положение данного Технического задания являются обязательным к выполнению, соблюдение всех пунктов выполнения Технического задания по Каскадной модели разработки ПО, если с Клиентом не согласовано иное дополнительное соглашением к ТЗ, Дизайном, перепиской посредством электронной почты либо иным способом. В ходе разработки не получится внести изменения. Также Заказчик имеет белую видимость и прозрачность процесса разработки. Всегда будет понятно, где команда находится сейчас, что еще осталось сделать и когда продукт будет готов.

* 1. Пользовательские требования в виде Use Case
     1. Пользовательские требования “Войти в мобильное приложение”

|  |  |
| --- | --- |
| Вариант использования | Войти в мобильное приложение |
| Контекст использования | Для просмотра своих данных в мобильном-приложение и прочтение/создание тредов, не вводя повторно персональные данные |
| Область действия | Мобильное-приложение Kaplys |
| Уровень | Цели пользователя |
| Основное действующие лицо(актор) | Зарегистрированный пользователь приложения |
| Участники (доп. действующие лица) | - |
| Предусловие | Пользователь зашел в мобильное- приложение Kaplys |
| Минимальные гарантии успеха | Система запомнила точку, до которой дошел пользователь |
| Гарантии успеха | Пользователь авторизован |
| Триггер | В любой момент, если пользователь находится в приложение и аутентифицированный, и решает создать ветвь обсуждения |
| Описание (основной сценарий) | 1.Пользователь переходит к авторизации  2.Пользователь вводит электронную почту и пароль  3.Пользователь нажимает кнопку “Войти”  4.Система подтверждает введенные данные  5.Система открывает страницу личного кабинета |
| Расширения(альтернативный сценарий) | 4.1 Пользователь получает сообщение о том, что электронный адрес еще не зарегистрирован  4.2 Система возвращается к шагу 1 |
| Изменения в технология и данных | - |

* + 1. Пользовательское требование “Создание треда”

|  |  |
| --- | --- |
| **Вариант использования** | Создать тред |
| **Контекст использования** | Для создания сообщения в том или ином разделе, для ведения диалога с другими пользователями |
| **Область действия** | Мобильное-приложение Kaplys |
| **Уровень** | Цели пользователя |
| **Основное действующие лицо(актор)** | Зарегистрированный пользователь мобильного приложения |
| **Участники(доп. действующие лица)** | - |
| **Предусловие** | Пользователь зашел в мобильное приложение Kaplys |
| **Минимальные гарантии успеха** | Система запомнила точку, до которой дошел пользователь |
| **Гарантии успеха** | Пользователь создал собственный тред ветвь обсуждения |
| **Триггер** | В любой момент, если пользователь находится в приложение и аутентифицированный, и решает создать ветвь обсуждения |
| **Описание(основной сценарий)** | 1.Пользователь переходит на нужную категорию тем обсуждения.  2.Пользователь нажимает кнопку “Создать тред”  3.Пользователь заполняет необходимую информацию для создания треда  4.Система получает данный запрос  5.Система создает тред на главной странице |
| **Расширение(альтернативный сценарий)** | 4.1 Пользователь получается ошибку на экране мобильного приложения 501 Not Implemented  4.1 Система возвращает к шагу 1 |
| **Изменения в технологиях и данных** | **-** |

* + 1. Пользовательское требование “Ответить в ветвь обсуждения”

|  |  |
| --- | --- |
| Вариант использования | Ответить в ветвь обсуждения |
| Контекст использования | Для активного участия в интересующей теме общения, добавив свой ответ в определённый тред сообщения. |
| Область действия | Мобильное-приложение Kaplys |
| Уровень | Цели пользователя |
| Основное действующие лицо(актор) | Зарегистрированный пользователь мобильного приложения |
| Участники(доп. действующие лицо) | - |
| Предусловие | Пользователь находится в мобильном-приложение и в нужном ему разделе ветви обсуждения |
| Минимальные гарантии успеха | Система запомнила точку, до которой дошел пользователь |
| Гарантии успеха | Пользователь ответил в треде |
| Триггер | В определённом моменте, если пользователь желает оставить свое сообщение в интересующей ветви обсуждения |
| Описание(основной сценарий) | 1.Пользователь переход в интересующий ему раздел обсуждения  2.Пользователь находит тред,в котором пожелал оставить свой комментарий или сообщение  3.Пользователь нажимает на кнопку “Ответить”  4.Пользователю открывается ветвь обсуждения, с прокруткой вниз до самого последнего.  5.Пользователь оставляет свой комментарий, по желанию добавляю фотографию в формате png и нажимает кнопку “Отправить” |
| Расширение(альтернативный сценарий) | 3.Пользователю выдается ошибку, что на данный момент не доступно комментирование тред.  3.1 Система возвращает к шагу 2. |
| Изменения в технологиях и данных | - |

1. Проектирование
   1. Цели и задачи разрабатываемого продукта

Разрабатываемое программное обеспечение предоставляет собой связку Мобильного Приложения и Управляющего Сервера.

*Задача Мобильного Приложения –* Дать пользователям возможность ведения активной беседы с различного пользователя Мобильного Приложения, осуществление активных бесед и дискуссий, по интересующим тематик разделов. Приложение должно обладать коммуникационными функционалом для создания ветвей общения, создание того или иного контента для обсуждения, ведение общения и дискуссий.

*Общий сценарий использования -*  Пользователь создает один или несколько ветвей сообщений в интересующих ему категорий. Все пользователи приложения имеют полный доступ прочтения и создания тредов, к любой интересующей категории. Пользователям при ответе в треде открывается меню комментирование и полный функционал возможностей украсить свое сообщение, открывается возможность прикреплять существующие в приложение дополнительные функции комментирования: смайл, gif, жирный текст, курсив, зачёркнутый, встроенный код, спойлер. Также пользовать может прикрепить свою сохранную фотографию на телефоне, в формате jpg и png, и видеоматериал в формате mp4. Приложение должно обладать функциями социальной сети. После регистрации в приложение, пользователь получает свой личный профиль, с функциями и возможностями редактирования профиля и открытия настроек встроенных в приложение. Пользователи приложения могут подписываться друг на друга, ставить лайки и писать комментарии.

*Дизайн приложения –* должен быть создан с учетом существующего брендбука, предоставленного Клиентом. Интерфейс приложения должен быть оригинальным, нестандартным, полу-игровым, удобным и простым для своей целевой аудитории.

*Целевая аудитория –* Клиенты iOS и Android. Подробные технические требования указаны в разделе – данного ТЗ.

* 1. Нефункциональные требования

За качество ПО отвечает совокупность всех характеристик, которые влияют на то, как пользователи мобильного приложения смогут решать свои задачи с помощью использования этого ПО.

* + - 1. Внешние атрибуты

1. Доступность – Приложение должно быть доступно на все 100% времени в сутки, без исключения времени на плавные и технические обслуживания. Вся серверная часть происходит в реальном времени, без нарушение доступности пользователям к использование приложения.
2. Удобство установки – *Установка* мобильного приложения Kaplys должна осуществляется в маркетплейсах AppStore и Google Play. *При обновлении* версии приложения должна сохраниться авторизация пользователя и все настройки его профиля. *При удалении* приложения, у пользователя сохраняется все настройки его профиля, но теряется сохранение в авторизации.
3. Целостность – Система должна хранить бессрочно все устаревшие треды сообщения во всех категориях.
4. Совместимость – Система должна иметь возможность получать список друзей из системы “Друзья”. Система должна иметь возможность получать список категорий из системы “Категории”
5. Производительность – Система должна уведомлять пользователя об получение уведомлений, сразу же после отправки сообщения, не позднее 3 секунд после оповещения.

Система должны отправить пользователю сообщение на указанную почту, об успешной регистрации в приложение не позднее 10 секунду после регистрации.

1. Надёжность – Количество сообщение об отправки или ответа, завершенных по причине ошибок в работе системы или ее компонентов, не должна превышать более 0.1% от всех сообщений в сутки.
2. Устойчивость – Если при взаимодействие с приложением произошел серверный сбой и был выполнен автоматический перезапуск приложения, система должны восстановить архитектурную последовательность действий и вернуться к тому, где у пользователя находился во время сбоя.
3. Защита - Физический вред не предоставляется во время использования приложения.
4. Безопасность – При подключение дополнительной защиты профиля, входя в приложение система должна запрашивать подтверждение в виде четырехзначных цифр и ответа на секретное слово.
5. Удобство использовании - Пользователь, авторизованный в приложение должен словесно ознакомиться со всеми видами категорий не более, чем за 20 секунд.
   * + - 1. Внутренние атрибуты
6. Эффективность – При размещение готовой версии приложения в AppStore и Google Play размер приложения должен быть менее 300 Мб.
7. Возможность модификации – Система обновляется по запросу пользователь и решения руководства, управление версиями продукта должно осуществляться с помощью Git.
8. Переносимость – Мобильное приложение должно запускаться в операционных системах IOS 13 и более, Android 9 и более.
9. Возможность повторного использования – Функциональность создания определенной категории в приложение, позволяет переиспользовать программный код для создания идентичной категории, заменив некоторые атрибуты.
10. Масштабируемость – Необходимо обеспечить быструю модерацию сообщений в тредах во всех категориях, для поддержания соблюдения правил использования приложения и планового увеличения регистрации новых пользователей.
11. Тестируемость – Должна быть предоставлена возможность отслеживания результата тестирования каждого отдельного функционального требования в одном рабочем пространстве.
    1. Требования к внешним интерфейсам
       1. Пользовательский интерфейс

4.3.1.1 Категория раздела должна содержать кратную информацию о себе, при переходе на нее.

4.3.2. Программный интерфейс

4.3.2.1. Система регистрации пользователя через почту Gmail

4.3.2.2. Система авторизации через почту Gmail и Apple

4.3.3. Аппаратный интерфейс

4.3.3.1. Неприменимо

4.4.3. Коммуникационные интерфейсы

4.4.3.1. Система должна отправлять пользователю сообщение на почту указанную в профиле, с уникальным кодом для подтверждения смены пароля учётной записи

4.4.3.2. Система должна отправлять пользователю сообщение на почту об успешной регистрации в приложение, с указанием его логина и пароля.

* 1. Ограничения проектирования и реализации
     1. Ограничения в использование ЯП

Продукт должен быть разработан с использованием библиотек Python и использование Фреймворка Kivy, для детального и качественного создания мобильного приложения, а клиенская часть должна быть написана на Objective-C.

* + 1. Ограничения в использование БД

Серверное-хранилище должно быть разработано и использовать базу данных SQLite, для хранения сообщений, которую не очень просто прочитать, но которая в целом лишена шифрованию.

* + 1. Ограничения в использование Библиотек

Продукт должен быть разработан в использовании языка программирования Python, но также должны быть использованы два фрейморвка Robot Framework и Pytest, для проведения быстро тестирования новой версии приложения.

* 1. Функциональные требования

Каждый экран должен быть подробно описан, как по назначению, так и по функционалу от Заказчика. Сначала мы описываем задачу экрана, условие перехода и возможные ограничения. После переходим к функционалу экрана, ссылаясь на макет и обозначение на нем элементов интерфейса.

*\*Дизайн страницы, название полей, расположение элементов может быть изменено на этапе технического проектирования и дополнительно согласовано с Заказчиков.*

*\* Иллюстрации к функциональным требованиям описания элементов экрана будет находиться в дополнительном файле, для удобства и легкости прочтение Технического задания*

|  |  |  |
| --- | --- | --- |
| № | Наименование элементов | Описание |
| 1 | Логотип приложения | Содержит img элемент экрана авторизации |
| 2 | Блок приветствия пользователя | Текстовое сообщение |
| 3 | Кнопки Api авторизации | При нажатие кнопки, происходит log in по Api Google или Apple |
| 4 | Кнопка зарегистрироваться | При нажатие на кнопку, происходит переход на экран регистрации аккаунта |
| 5 | Кнопка авторизация | При нажатие на кнопку, происходит переход на экран авторизоваться в систему |

Таблица 1 Экран авторизации пользователь

|  |  |  |
| --- | --- | --- |
| № | Наименование элементов | Описание |
| 1 | Кнопка крестик “Закрыть” | При нажатие на кнопку, происходит закрытие экрана регистрации пользователя и перенаправляет на экран авторизации |
| 2 | Логотип приложения | Содержит img элемент экрана регистрации |
| 3 | Блок приветствия нового пользователя | Текстовое сообщение |
| 4 | Блок ввода email | Поле для текстового ввода персональных данных для регистрации |
| 5 | Блок ввода password | Поле для текстового ввода пароль пользователя, для дальнейшей авторизации в систему |
| 6 | Блок ввода Имя пользователя | Поля для текстового ввода придуманного имя пользователя |
| 7 | Кнопка “Продолжить” | При нажатие на кнопку, происходит переход на экран профиль пользователя |

Таблица 2 Экран регистрации нового пользователя

|  |  |  |
| --- | --- | --- |
| № | Наименование элементов | Описание |
| 1 | Кнопка “Возвращение назад” | При нажатие на кнопку, происходит возвращение пользователя на предыдущий экран,c которого перешел в Профиль. |
| 2 | Аватар пользователя | Содержит img элемент в профиле, виден всем пользователь приложения, как при переходе на профиль определённый или активность в приложение. |
| 3 | Имя пользователя | Индивидуальный идентификатор пользователя в приложение, обозначающие имя самого пользователя. |
| 4 | Количество подписчиков | Функциональный количественный счетчик, показывающий число следящий пользователь за профилем. |
| 5 | Текстовое живой счетчик активности | Функциональный текст сообщений, показывающий пользователю активность его в приложение, выдающий информацию об последней навигации в категориях, количество в сети друзей и последняя активность. |
| 6 | Кнопка "Настройки” | При нажатии на кнопку, происходит открытие нового экрана приложения с настройками приложения и его взаимодействия. |
| 7 | Кнопка “Изменить профиль” | При нажатии на кнопку, происходит открытие нового экрана приложения с личными настройками профиля и его редактирования |
| 8 | Навигации кнопки | При нажатие на кнопки, происходит открытие определенных элементов экрана и нужной информации, желающий пользователь увидеть или создать |
| 9 | Блок с изображением | Содержит img элемент экрана пустоты информации |
| 10 | Текстовое сообщение | Сообщение о том, что на данный момент в категориях пусто и нет информации |
| 11 | Навигационные кнопки | При нажатие на кнопки, происходит переход к одноименным страницам приложения |

Таблица 3 Экран профиля пользователя

|  |  |  |
| --- | --- | --- |
| № | Наименование элементов | Описание |
| 1 | Логотип приложения | Содержит img элемент экрана категории |
| 2 | Текстовое сообщение | Сообщение о самом мобильном приложение и информация об категориях, и словесные правила взаимодействия и общения |
| 3 | Блок с изображением | Содержит img элемент экрана категории |
| 4 | Навигация категорий | Общий список категорий, с возможностью сделать scroll вниз-вверх, для нахождение нужной категории и перехода по ней |
| 5 | Навигация кнопок | При нажатие на кнопки, происходит переход к одноименным страницам приложения |

Таблица 4 Экран категорий

|  |  |  |
| --- | --- | --- |
| № | Наименование | Описание |
| 1 | Кнопка “Главная” | При нажатие на кнопку, происходит переход на страницу приложения Главная. |
| 2 | Чекбокс категорий | На нажатие выпадающий список, появляются сокращенное наименование категорий, по нажатие на категорию определённую, осуществляет переход по ней |
| 3 | Кнопка “Темная тема” | При нажатие на кнопку, меняется светлая тема(автоматически сперва устанавливается) на темную тему. Становится темное приложение. |
| 4 | Текстовое сообщение | Наименование категории, на которой находится пользователь на данный момент. |
| 5 | Кнопка “Создать тред” | При нажатие на кнопку, происходит перенаправление на другую страницу, где пользователь может оставить комментарий в треде или ответить пользователю, с дополнительными функционалом. |
| 6 | Блок с изображением темы | Содержит img элемент экрана определенной категории, изображение добавляется в тематику самой категории. |
| 7 | Кнопка “Каталог” | При нажатие на кнопку, происходит перенаправление на другую страницу этой же категории, но расширяется список имеющейся тредов на данный момент, треды изображены блоками для быстрого ознакомления |
| 8 | Текстовое сообщение темы | Наименование темы ветвяного сообщения |
| 9 | Текстовое сообщение информации | Текстовое наименование содержащийся информацию об создателей треда, прописывается Имя пользователя и дата создания ветви обсуждения |
| 10 | Блок с изображение аватара | Содержит img изображения пользователя(создателя треда) |
| 11 | Текстовое сообщение треда | Текстовое описание самого треда |
| 12 | Текстовое сообщение оповещение | Функциональный счётчик, показывающий сколько пользователь пропустил сообщений в ветви обсуждения. Измеряется в цифрах-числах без ограничений |
| 13 | Кнопка “В тред” | При нажатие на кнопку, пользовать перенаправляется в сам ветвь обсуждения сообщений |
| 14 | Кнопка “В ленту” | При нажатие на кнопку, данный тред добавляется в главную ленту пользователь(главная страница), имея тот же функционал, как и кнопка “В тред” |
| 15 | Навигация кнопка | При нажатие на кнопки, происходит переход к одноименным страницам приложения. |

Таблица 5 Экран категория-тред

* 1. Диаграммы UML схемы

Архитектурные схемы предоставляют с собой графическое описание для объектное моделирования в области системного проектирования и разработки программного приложения.

Разработка диаграммы вариантов использования – диаграмма прецедентов(вариантов использования) отражает отношения между актером и прецедентами системы функциональных требованиях, и позволяет описать систему на концептуальном уровне. Прецедент – возможность моделируемой системы, часть ее функциональности, благодаря которой актер может получить конкретный ему результат. Прецедент соответствует отдельному сервису системы, определяет один из вариантов ее использования и описывает типичный способ взаимодействия пользователя с системой. Диаграмма прецедентов обычно применяется для спецификации внешних требований к системе.

В ходе проектирования был выделен следующий актер.

*Пользователь* – пользователь приложения, которому доступна возможность использования всех его функций.

На основе функциональных требований к мобильному приложению была создана диаграмма прецедентов, которая представлена на рис.1

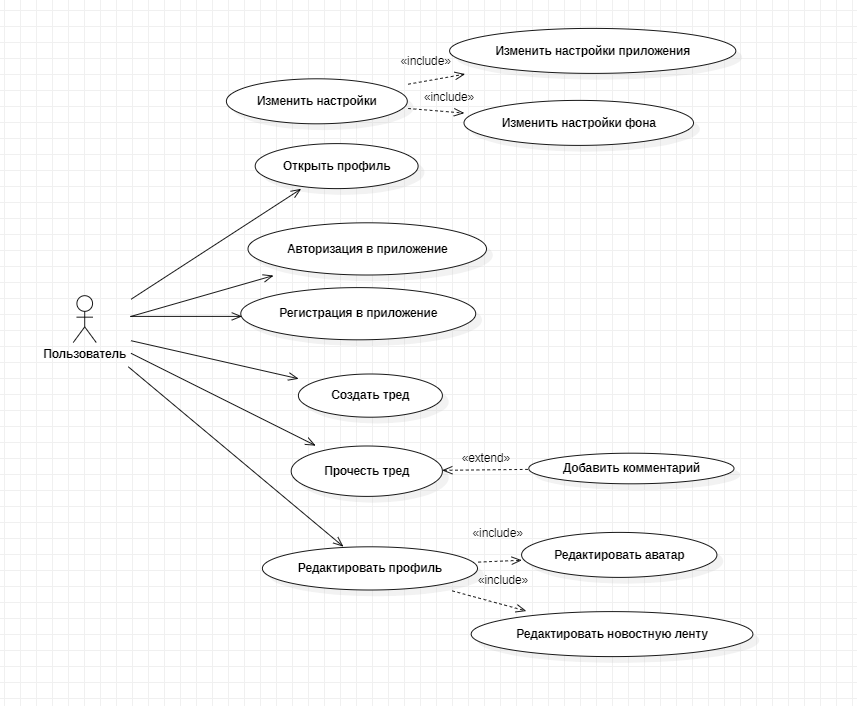


Рисунок 1 Схема диаграмма прецедентов

*Авторизация в приложение –* авторизация в приложение позволяет уже зарегистрированным пользователям войти в систему, введя логин и пароль.

*Регистрация в приложение –* регистрация в приложение служит для регистрации новых пользователь, где необходимо указать email, name, password – для успешной регистрации, и для дальнейшего взаимодействия с мобильным приложением и осуществление всех функциональных требований, заложенных в систему.

*Открыть профиль –* открыть профиль позволяет пользователь увидеть свои персональные настройки, как выглядит его профиль, изменить его функциональность, открыть навигационные панели профиля, для комментирования, посты и обо мне информация.

*Изменить настройки –* открывается страницы со всеми настройками мобильного приложения, начиная от include изменить само приложения под свои личные настройки и пожелания внешнего интерфейса, также с персональными системными настройками.

*Создать тред –* функция служит для того, чтобы пользовать открыл заранее интересующий ему категорию, после чего по желанию пользователя создаст тред(ветвь обсуждения), где выполнив рекомендацию по построению треда, он будет виден людям и активен.

*Прочесть тред –* прочесть тред позволяет ознакомиться с содержимой информацию, донесенная от другого пользователя, зависящая от интересов и выбранной категории пользователя, с возможность прочтения комментария и добавление собственного комментария.

Работа мобильного приложения зависит от сервера .Все данные и информация, имеющееся в приложения – информация об тредах, категориях , новостная лента – постоянно актуализируются с сервера. Также в ходе работы приложения на сервер отправляются данные: личные диалоги пользователей, активность по отношению к профилю другого пользователя, создания тредов и его каждая ветвь обсуждения, внешние настройки профиля.

**Разработка модели диаграмма классов -** класс – элемент диаграммы, обозначающий множество объектов, обладающих одинаковой внутренней структурой, поведением и отношениями с объектами других классов. Изображается класс на диаграмме в виде прямоугольника, разделенного на три секции:

1. Имя класса
2. Список атрибутов класса
3. Список методов класса.

На рис.2 представлена схема модели классов компонентов системы

Система содержит 9 различных структур данных – Регистрация, Пользователь, Система, Лента новостей, Категория, Тред, Ветви сообщений, Профиль и Настройки профиля.

*Регистрация –* класс имеющий главную задачу и функционал, давая возможность пользовать зарегистрироваться и взаимодействовать с приложением.Из публичных(+) данный для пользователя, для осуществление регистрации, в атрибутах представлены email, password, name с определённым типом данных. В методах класса указывается addUserSystem(добавление пользователя в систему), getUserSystem(получать зарегистрированного статус пользователя),errorRegistrationUserEmail(ошибка регистрации из-за поля email),errorRegistationUserName(ошибка регистрации из-за поля username)

*Пользователь –* класс имеющий персональные данные пользователя и хранящие их в атрибутах. Атрибуты класса пользователь содержится userId(персональный id номер пользователя), userName(имя пользователя), password(пароль пользователя) и email. В методах класса указываются EditUser(пользователь может изменить персональные настройки), DeleteUser(пользователь может удалить себя из системы) и ValidateUser( валидация пользователь при авторизации)

*Система –* само мобильное приложение, входящие все классы в него и связующие для всех компонентов. Имеющие атрибут systemWorks (работает система),c типом данных Boolean истина или лож. В методах notifyWork.

*Лента новостей –* предоставляется с собой класс, хранивший в себе атрибуты *Пользователя* и *Тред.*

Классы Категория, Тред, Ветвь сообщений, Профиль и Настойки профиля имеются схожие атрибуты и методы.

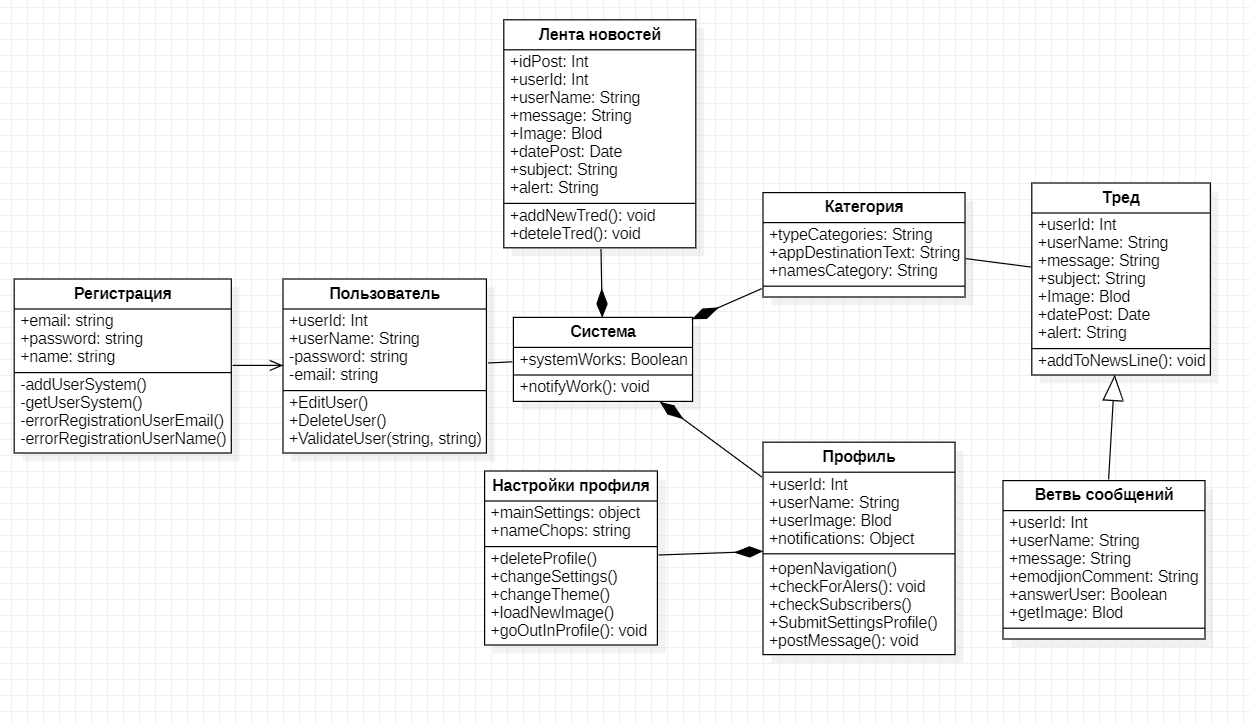


Рисунок 2 Схема диаграмма классов

**Разработка диаграммы последовательности –** UML диаграмма, которая представляет взаимодействие между линиями жизни как упорядоченную последовательность событий. На рис.3 представлена диаграмма последовательности, описывающая процесс регистрации в приложение и создание треда. В данном процессе происходит регистрация с ответом от мобильном системы(базы данных), что прошла регистрация и при альтернативном варианте, где появляется ошибка и система просит произвести повторную регистрацию с вводом данных. После процесса регистрации, пользователь выбирает категорию и нажимает создать тред, после заполнение необходимой информации происходит создание треда с оповещением пользователя дисплеем.

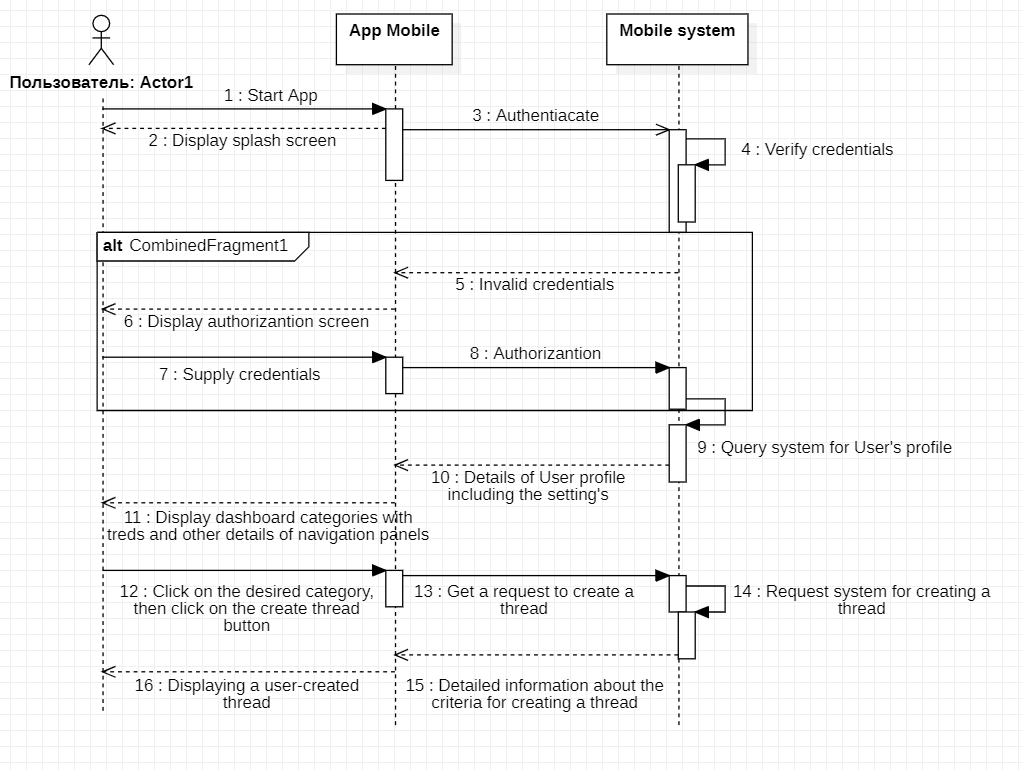


Рисунок 3 Схема диаграммы последовательности

**Разработка диаграммы деятельности -** На рис.4 представлена диаграмма деятельности мобильного приложения пошаговый процесс регистрации, переход в раздел категории, выбор интересующей категории и создания треда, с циклами ИЛИ/ИЛИ.

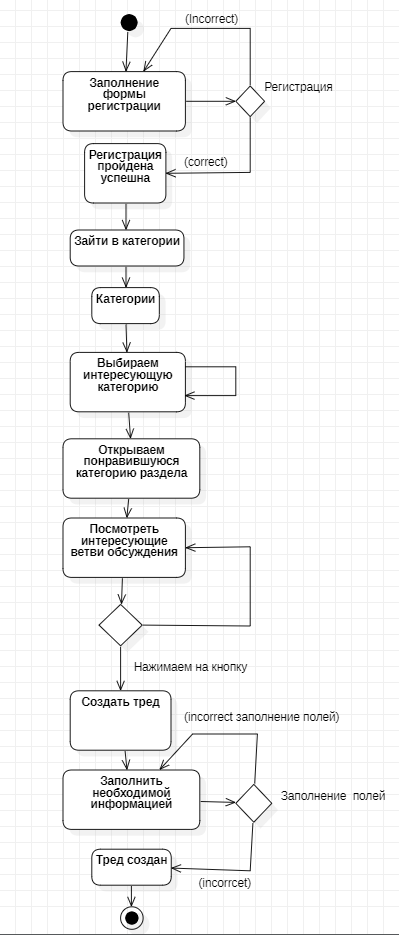


Рисунок 4 Схема диаграммы деятельности регистрации и создания треда

1*. Начальным состоянием системы* является запуск мобильного приложения. При запуске мобильного приложения, пользователю необходим произвести регистрацию, для дальнейшего пользованием приложения.

2*. Во время процесса регистрации* и для дальнейшей последовательности, пользователь переходит на цикл ИЛИ, где при корректном выполнение регистрации, пользователь создает свой профиль и переход дальше, при некорректной регистрации, система просит повторно произвести регистрацию.

3*. Пользователь изучает мобильное приложение и переходит в навигационную панель Категорию,* пользователь выбирает интересующую категорию для обсуждения или прочтения, процесс является повторным, где пользователь может остановится на долгое время и просмотреть все категории, их наименование и описание.

4. Пользователь открывает понравившуюся раздел категорию, при дальнейшем переходе пользователь может длительное время просматривать все существующие на данный момент треды(ветви обсуждения), процесс является повторяющемся и стоит на месте, до решения пользователя принять решение. Далее появляется цикл ИЛИ, при котором можно создать свой собственный тред или продолжить просматривать ветви обсуждения.

5. *Пользователь нажимает на кнопку Создать тред* , далее происходит перенаправление на окно заполнение необходимой информации ветви сообщения для постинга. Заполнив необходимые поля , осуществляется цикл ИЛИ, при корректном выполнение заполнение полей, создается тред в категории, при некорректном заполнение полей, система показывает ошибки и просит исправить их, оставляя пользователя на этом же этапе.

1. Разработка дизайна и требований ПО
   1. Требования к навигационной панели

Панель навигации должна состоять из следующих разделов:

1. Лента. В данном разделе выводятся все добавленные пользователем треды, при нажатии на кнопку “В ленту”.
2. Название темы треда.
3. Имя автора пользователя треда, с датой постинга формата года и времени xx/xx/xxxx, наим. xx/xx.
4. Главное изображение треда .
5. Текстовое сообщение.
6. Информационная панель, сколько пропущенный прочтение сообщений в треде.
7. Кнопка “В тред”, при нажатие пользователя перенаправляет в ветвь обсуждения.
8. Категории. Данный раздел отображает все существующие категории в мобильные приложения.
9. Текстовое наименование категорий.
10. Текстовое сообщение описывающая функциональность и предназначения приложения.
11. Блок с изображением
12. Создать. Данный раздел осуществляет быстрое перенаправление на экран создания треда.
13. Чекбокс с выпадающий списком, для выбора категорий.
14. Наименование категории сообщения.
15. Поля ввода информации для создания треда.
16. Кнопка “Закрыть форму постинга”, возвращает на предыдущий этап до нажатия кнопки Создать.
17. Панель дополнительных возможности редактирования текста
18. Кнопка “Создать тред”, при заполнение всех необходимых полей информацией.
19. Чат. Данная страница содержит панель социальной сети, для ведения личного диалога с пользователями приложения.
20. Имя пользователя
21. Идентификатор пользователя
22. Текст сообщения
23. Кнопки взаимодействия
24. Аватар пользователя
25. Входящие. Данный раздел включает открытие профиля и перенаправление на страницу с навигацией Посты, Комментарии, Обо мне.
26. Комментарии к тредам
27. Созданные треды пользователем
28. Информация об пользователе, созданная отдельными постами.
    1. Общие рекомендации по безопасности
29. При работе с базой данных Сервера, в области обработки данных, полученных извне, следует использовать безопасные методы обработки NO-SQL запросов.
30. Пароль пользователей должен храниться в базе данных Сервера в хешированном виде. При хеширование следует использовать salt.
31. Деавторизация Пользователей не должна происходит автоматически.

Сервер предполагается к реализации на языке программирования Python, с использованием фреймворка Pytest и Kivy. Верстка мобильного приложения и осуществление функционала будет осуществляться Swift и Kotlin.

* 1. Технические требования к ПО

ПО предполагается к разработке с помощью следующих технических средств:

Python, NoSql ( или иная БД), Swift или Object-c для разработки на мобильной платформе Apple, Kotlin для разработки на мобильной платформе Android. ПО должен корректно работать в следующих версиях ОС.

* iOS 14.2 IPhone Mobile
* Android 11.3 version.

1. Тестирование
   1. Функциональное тестирование

Функциональное тестирование – это тестирование программного продукта обеспечения в целях проверки реализуемости функциональных требований, то есть способности ПО в определённых условиях решать задачи , нужные пользователям. Функциональные требования определяют, что именно делает программное обеспечение, какие задачи оно решает. Осуществление тестирования мобильного приложения занимается отдел QA инженеров.

* 1. Интеграционное тестирование

Интеграционное тестирование – полная проверка программного обеспечения продукта после его сборки с целью выявления ошибок, возникающих в процессе интеграции программного модуля или компонентов.

1. Этапы и сроки Внедрения

Работы по данному ТЗ выполняются в сроки и этапы, указанные в управление проектами Jira , согласно условиям Договора xx-xx-xx от xx.xx.xxx.

Работа над Социальной сети будет вестись по Каскадной модели жизненного

Начало работы проекта состояние с первого этапа цикла разработка ПО анализа и произойдет в день начала работы над Проектом. Постепенно после выполнения каждого заложенного этапа в документации, будет осуществляться переход к дальнейшим этапам разработка ПО. Таким образом, берет значительно общее время на проектирование всего проекта и разбивается на 6 этапов, где примерно берется образно время на выполненного того и иного этапа, по итогу суммируется время разработки. При выполнение всех этапов, при имеющемся свободным времени по заложенному договору, будет происходить перепроверка всех этапов разработки. При нехватке времени на выполненного того или иного этапа, больше сил будет накладываться на следующий, для укрощения потери во времени. Данный период называется “Узкое место производства” (Bottleneck). В данный период не включается время, необходимое Клиенту на рассмотрение и утверждение Дизайна или иных промежуточных результатов

Приложение А.

Концептуальные макеты Мобильного Приложения

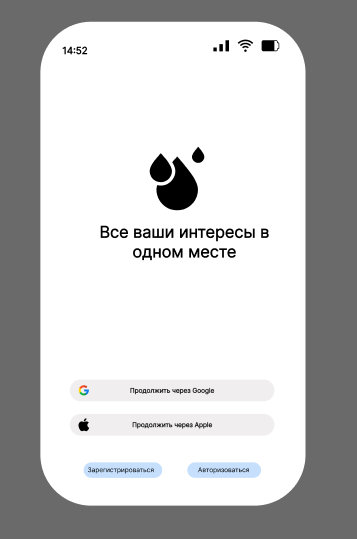


Рисунок 1-A Окно авторизации

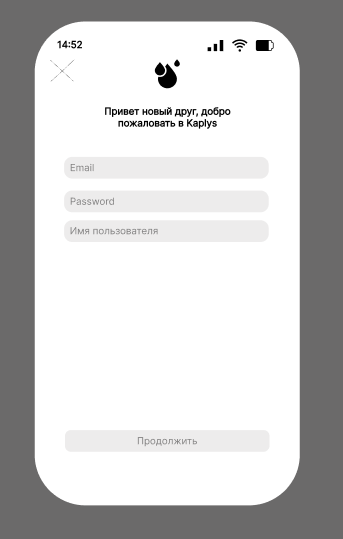


Рисунок 2-A Окно регистрация пользователя

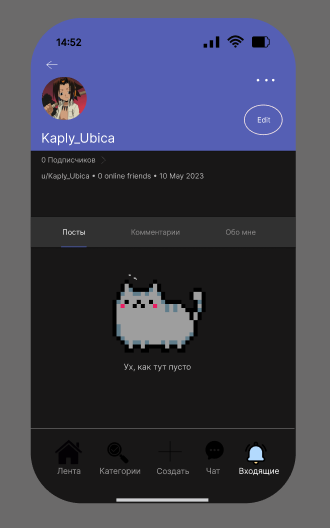


Рисунок 3-A Окно профиль пользователя

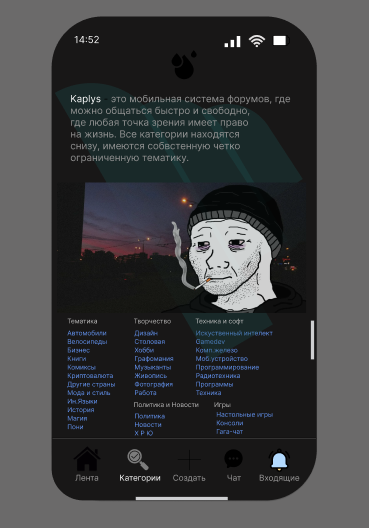


Рисунок 4-A Окно категории обсуждения

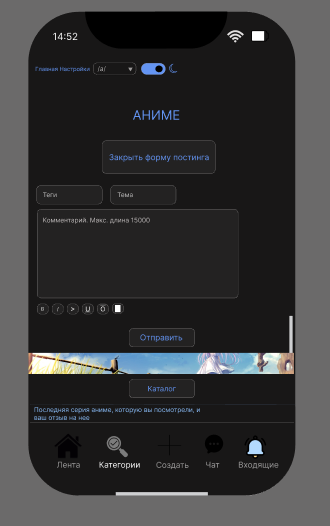


Рисунок 5-A Окно создания треда

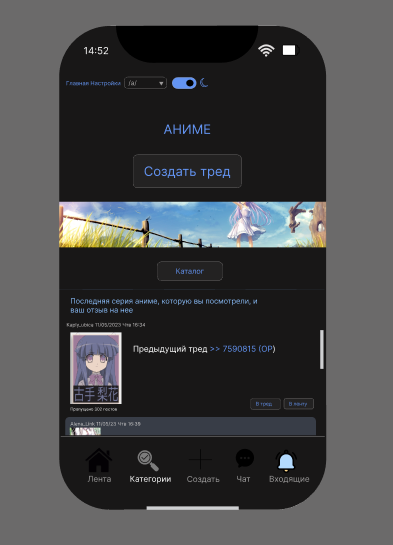


Рисунок 6-A Открытие категории и прочтения треда