**Введение**

На учебную практику по разработке и сопровождению программного обеспечения была поставлена задача разработать мобильное приложение “DIX”.

Цель учебной практики заключается в разработке мобильного мессенджера «DIX», которое позволит общаться с людьми на расстоянии через всемирную сеть Интеренет.

Создаваемая программа будет рассчитана на любого рода пользователей.

Далее приведем краткое описание разделов пояснительной записки.

Первый раздел носит название “Анализ задачи”. В нем вы сможете ознакомиться с постановкой задачи, которая включает в себя: исследование предметной области поставленной задачи, определение ее организационно-экономической сущности. Также в этом разделе вы сможете узнать о том, как данная задача решается в настоящее время. Все входные и выходные данные тоже будут описаны в первом разделе. В подразделе “Инструменты разработки” будет рассмотрена среда, в которой создается данный курсовой проект. Здесь также будут установлены минимальные и оптимальные требования к аппаратным характеристикам, обеспечивающим правильное функционирование поставленной задачей.

В разделе “Проектирование задачи” будут рассмотрены основные аспекты разработки программного продукта. Здесь можно будет узнать об организации данных в контексте среды разработки. В данном разделе будет четко описан пользовательский интерфейс, составлены алгоритмы процесса обработки информации, описана разработка системы справочной информации.

“Реализация задачи” – это третий раздел пояснительной записки, в котором описываются все элементы и объекты, которые будут использованы при реализации данного приложения. В этом разделе будут четко описаны функции пользователя и их структура. Здесь можно будет найти таблицу, в которой будет представлена полная аннотация файлов используемых в данном проекте.

Четвертый раздел – “Тестирование”. В нем будет описано полное и функциональное тестирование данной программы, т.е. будет оттестирован каждый пункт меню, каждая операция, которая выполняется приложением. Будут смоделированы все возможные действия пользователя при работе с программой, начиная от запуска до выхода.

В разделе “Применение” будет описано назначение, область применения, среда функционирования курсовой программы. Также в нем будет описано использование справочной системы.

“Заключение” будет содержать краткую формулировку задачи, результаты проделанной работы, описание использованных методов и средств, описание степени автоматизации процессов на различных этапах разработки.

В “Списке использованных источников” будет приведен список используемых при разработке источников.

В приложениях к пояснительной записке будет приведен листинг программы с необходимыми комментариями.

1. **Анализ задачи**

**1.1. Разработка постановки задачи**

**1.1.1. Организационно-экономическая сущность задачи:**

Темой данного курсового проекта является «Разработка мобильного приложения «DIX».

Первый мессенджер запустили в 1988 году. Он назывался Internet Relay Chat. Его разработал программист из Финляндии Яркко Ойкаринен. Сервис был популярным, в 2009 году им пользовалось более 500 тыс. человек. Но намного более известным в то время стал мессенджер ICQ. К началу 2010 года в нем были зарегистрированы почти 48 млн пользователей по всему миру. Со временем ICQ не выдержал конкуренции социальных сетей. Уже в 2012 году, с ростом популярности MySpace, Facebook\* и «Вконтакте», его аудитория снизилась на 31%.

Можно привести такие аналоги приложения как: Telegram,WhatsApp,Viber;

Периодичность использования данного программного продукта неограниченна.

Цель данной учебной практики – разработать программный продукт,

который позволит вести чаты с людьми в разных точках мира.

За основу программы следует взять из соответствующих источников

(мессенджеров, сайтов). Приложение должно располагать своим интерфейсом и функционалом к пользователю, чтобы ему было комфортно. Данный проект должен стать общедоступным для всех пользователей. В поставленной задаче необходимо реализовать максимально простой пользовательский интерфейс, позволяющий использовать проект пользователю, не обладающему дополнительными знаниями в интерфейсах мобильных приложений.

Программный продукт предоставляет функционал для следующего ряда пользователей: зарегистрированный пользователь.

**1.1.2. Функциональные требования:**

Разрабатываемый программный продукт должен позволять выполнять

следующие действия:

-регистрация и вход в аккаунт пользователя;

- обмен сообщениями текстового формата;

-поддержка отправки смайлов;

**1.1.3. Описание исходной (входной) информации:**

Перечень исходной информации для мессенджера включает в себя:

1. Текстовые сообщения
2. Смайлы

Формы представления документов по каждой позиции перечня могут быть следующими:

1. Текстовые сообщения - диалоговое окно чата

Примеры заполнения документов:

1. Текстовые сообщения - "Привет, как дела?"

Перечень пользователей исходной информации для мессенджера включает в себя всех зарегистрированных пользователей мессенджера, которые могут обмениваться сообщениями, файлами и проводить аудио и видео вызовы.

**1.1.4. Описание результатной (выходной) информации:**

Результатная (входная) информация для мессенджера может включать в себя следующее:

- Сообщения от других пользователей мессенджера

- Уведомления о новых сообщениях и других событиях

Формы представления результатной информации для мессенджера могут быть различными, включая текстовые сообщения, изображения, видео и звуковые уведомления.

Периодичность и сроки представления результатной информации зависят от активности других пользователей мессенджера и могут быть непредсказуемыми.

Пользователями результатной информации для мессенджера являются другие пользователи этого мессенджера, как индивидуальные лица, так и группы пользователей.

**1.1.5. Описание используемой условно-постоянной информации:**

- Классификаторы контактов: список контактов пользователей мессенджера

- Таблица истории сообщений: показывает все отправленные и полученные сообщения

- Список уведомлений: список всех уведомлений о новых сообщениях

Формы представления для мессенджера могут включать интерфейс чата, список контактов, список групповых чатов, историю сообщений и уведомления.

**1.1.6.** **Нефункциональные (эксплуатационные) требования:**

- Требования к применению включают создание интуитивно понятного пользовательского интерфейса, доступной документации и обучающих материалов.

- Требования к производительности могут включать ограничения на время загрузки приложения, скорость отправки и получения сообщений, использование памяти и другие ресурсы.

- Требования к реализации могут предписывать использование определенных технологий, языков программирования, архитектурных решений и стандартов безопасности.

- Требования к надежности могут определять допустимое время простоя системы, частоту сбоев и возможности восстановления данных.

- Требования к интерфейсу могут включать поддержку различных устройств (компьютеры, мобильные устройства), способы взаимодействия (клавиатура, сенсорный экран) и доступность для людей с ограниченными возможностями.

**1.1.7. Составление плана и графика работы над проектом (диаграмма Ганта)**

Диаграмма Ганта — это популярный тип столбчатых диаграмм, который используется для иллюстрации плана, графика работ по какому-либо проекту. Является одним из методов планирования проектов. Используется в приложениях по управлению проектами. Первый формат диаграммы был разработан Генри Л. Гантом в 1910 году.

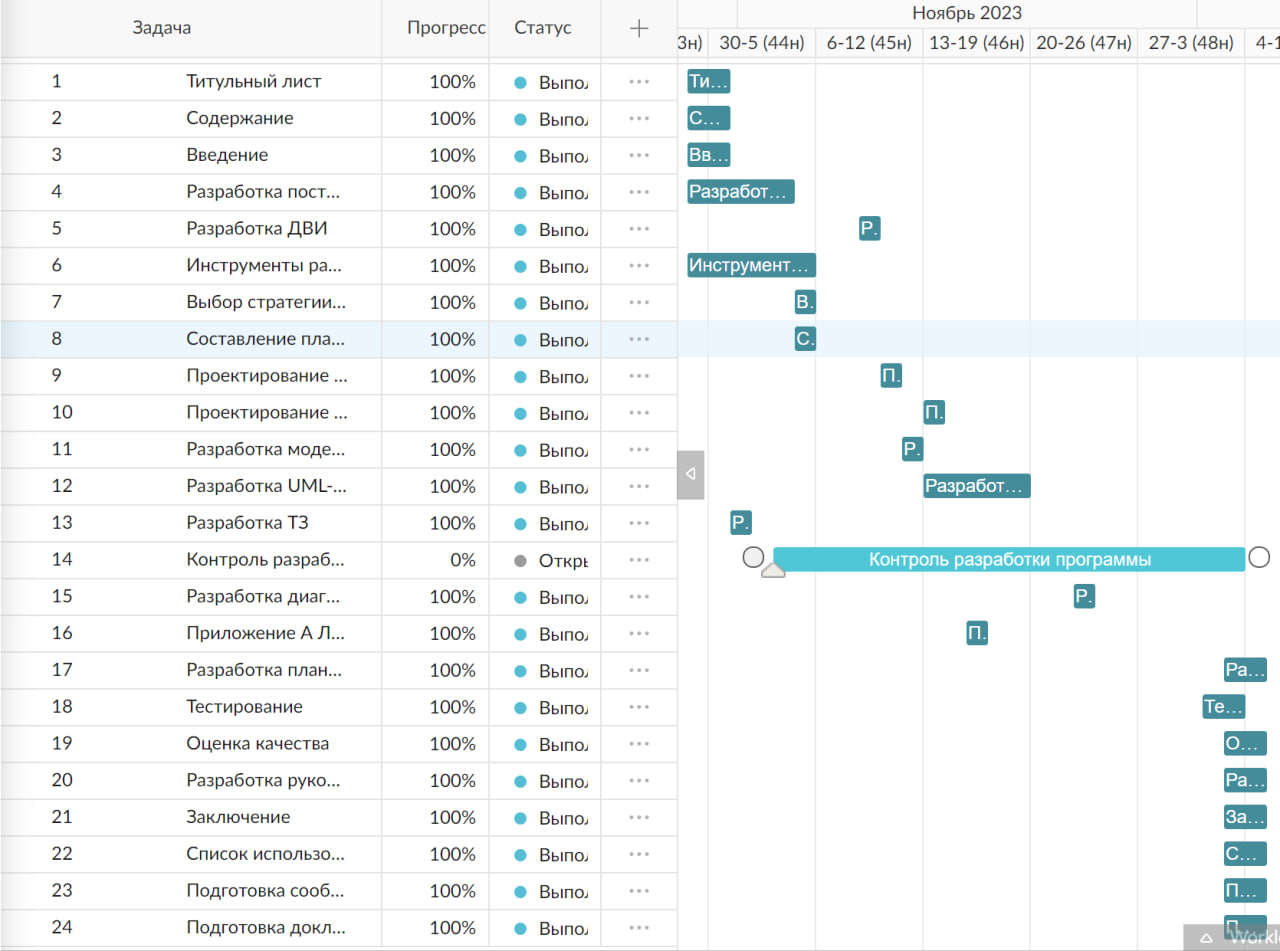


Рисунок 1- Диаграмма Ганта

**1.1.8. Эксплуатационные требования**

**Требования к применению:** Иметь доступ к интернету.

**Требования к реализации:** Для реализации статических страниц и шаблонов должен использоваться среда разработки Android Studio.

**Требования к надежности:** Система может быть недоступна не более чем 24 часа в год. У администратора приложения должна быть возможность выгрузить и загрузить копию приложения, доступ ко всем данным зарегистрированных пользователей.

**Требования к интерфейсу:** При разработке приложения должны быть использованы преимущественно простые цвета. Основные разделы приложения должны быть доступны с первой страницы. Грамотный и удобный пользовательский интерфейс. Приложение должно адаптироваться под компьютер, телефон и планшет.

**1.2. Выбор стратегии разработки и модели жизненного цикла**

Для разработки чат-приложения «DIX» следует выбрать стратегию разработки и модель жизненного цикла. Осуществляем выбор посредством составления таблиц:

Таблица 1 **–** Выбор модели жизненного цикла на основе характеристик требований

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| №  крит ерия | Критерии категории требований | Каска дная | Vобра зная | RAD | Инкрем  ентная | Быстрого прототип ирования | Эволюци  онная |
| 1. | Являются ли требования к проекту легко определимыми и реализуемыми? | Да | Да | Да | Да | Нет | Нет |
| 2. | Могут ли требования быть сформулированы в начале ЖЦ? | Да | Да | Да | Да | Нет | Нет |
| 3. | Часто ли будут изменяться требования на протяжении ЖЦ? | Нет | Нет | Нет | Нет | Да | Да |

Продолжение (Таблица 1):

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 4. | Нужно ли демонстрировать требования с целью их определения? | Нет | Нет | Да | Да | Да | Да |
| 5. | Требуется ли проверка концепции программного средства или системы? | Нет | Нет | Да | Да | Да | Да |
| 6. | Будут ли требования изменяться или уточняться с ростом сложности системы (программного средства) в ЖЦ? | Нет | Нет | Нет | Да | Да | Да |
| 7. | Нужно ли реализовать основные | Нет | Нет | Да | Да | Да | Да |
|  | требования на ранних этапах разработки? |  |  |  |  |  |  |
|  | Итого | 2 | 2 | 3 | 6 | 5 | 5 |

Итог: На основе результатов заполнения табл. 1 подходящей является модель быстрого прототипирования и инкрементная.

Таблица 2**–** Выбор модели жизненного цикла на основе характеристик команды разработчиков

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| №  критерия | Критерии категории команды разработчиков проекта | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |

Продолжение (Таблица 2):

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1. | Являются ли проблемы предметной области проекта новыми для большинства разработчиков? | Нет | Нет | Нет | Нет | Да | Да |
| 2. | Являются ли инструментальны е средства, используемые в проекте, новыми для большинства разработчиков? | Да | Да | Нет | Да | Нет | Да |
| 3. | Изменяются ли роли участников проекта на протяжении ЖЦ? | Нет | Нет | Нет | Да | Да | Да |
| 4. | Является ли структура процесса разработки более значимой для разработчиков, чем гибкость? | Да | Да | Нет | Да | Нет | Нет |
| 5. | Важна ли легкость распределения человеческих ресурсов проекта? | Да | Да | Да | Да | Нет | Нет |
| 6. | Приемлет ли команда разработчиков оценки, проверки, стадии разработки? | Да | Да | Нет | Да | Да | Да |
|  | Итого | 4 | 3 | 5 | 6 | 3 | 3 |

Итог: На основе результатов заполнения табл. 2 подходящими являются RAD и инкрементная модели.

Таблица 3 **–** Выбор модели жизненного цикла на основе характеристик коллектива пользователей

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| №  крите рия | Критерии категории коллектива пользователей | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| 1. | Будет ли присутствие пользователей  ограничено в  ЖЦ  разработки? | Да | Да | Нет | Да | Нет | Да |
| 2. | Будут ли пользователи оценивать текущее состояние программного продукта (системы) в процессе разработки? | Нет | Нет | Нет | Да | Нет | Да |
| 3. | Будут ли пользователи вовлечены во все фазы ЖЦ разработки? | Нет | Нет | Да | Нет | Нет | Нет |
| 4. | Будет ли заказчик отслеживать ход выполнения проекта? | Нет | Нет | Нет | Да | Нет | Да |
|  | Итого | 0 | 0 | 2 | 3 | 0 | 2 |

Итог: На основе результатов заполнения табл. 3 подходящей является инкрементная модель.

Таблица 4 **–** Выбор модели жизненного цикла на основе характеристик типа проектов и рисков

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| №  критерия | Критерии категории типов проекта и рисков | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| 1. | Разрабатывается ли в проекте продукт нового для организации направления? | Нет | Нет | Нет | Да | Да | Да |
| 2. | Будет ли проект являться расширением существующей системы? | Да | Да | Да | Да | Нет | Нет |
| 3. | Будет ли проект крупно- или среднемасштабны м? | Нет | Нет | Нет | Да | Да | Да |
| 4. | Ожидается ли длительная эксплуатация продукта? | Да | Да | Нет | Да | Нет | Да |
| 5. | Необходим ли высокий уровень надежности продукта проекта? | Нет | Да | Нет | Да | Нет | Да |
| 6. | Предполагается ли эволюция продукта проекта в течение ЖЦ? | Нет | Нет | Нет | Да | Да | Да |
| 7. | Велика ли вероятность изменения системы (продукта) на этапе сопровождения? | Нет | Нет | Нет | Да | Да | Да |

Продолжение (Таблица 4):

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 8. | Является ли график сжатым? | Нет | Нет | Да | Да | Да | Да |
| 9. | Предполагается ли повторное | Нет | Нет | Да | Да | Да | Да |
|  | использование компонентов? |  |  |  |  |  |  |
| 10. | Являются ли достаточными ресурсы (время, деньги, инструменты, персонал)? | Нет | Нет | Нет | Нет | Да | Да |
|  | Итого | 4 | 5 | 3 | 9 | 4 | 6 |

Итог: На основе результатов заполнения табл. 4 подходящей является Инкрементная и эволюционная модели.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Каска дная | Vобразная | RAD | Инкремент  ная | Быстрого прототипирова ния | Эволюционная |
| Таблица 1 | 2 | 2 | 3 | 6 | 5 | 5 |
| Таблица 2 | 4 | 3 | 5 | 6 | 3 | 3 |
| Таблица  3 | 0 | 0 | 2 | 3 | 0 | 2 |
| Таблица  4 | 4 | 5 | 3 | 9 | 4 | 6 |
| Общие итоги | 10 | 10 | 13 | 24 | 12 | 16 |

Общий итог: в итоге заполнения табл. 1 – 4 наиболее подходящей является инкрементная модель.

Инкрементная модель нам подходит, потому что задача достаточно сложная и требует изменения в процессе проектирования. Эта модель предполагает разбиение проекта на небольшие итерации, в ходе которых происходит анализ, дизайн, разработка, тестирование и интеграция продукта. Таким образом, мы можем получать работоспособные версии приложения на каждой итерации и постепенно улучшать его функциональность и качества.

* 1. **Инструменты разработки**

Для разработки данного проекта будет выбрана среда разработки Android Studio, которая является наиболее актуальной средой для создания приложений данного типа.

Разработка будет производится на таких языках программирования, как:

* Kotlin– статически типизированный, объектно-ориентированный

язык программирования, работающий поверх Java Virtual Machineсайта;

* WEB-ресурс DRAW.IO – будет использоваться для создания

графической части и разработки UML-диаграмм;

* Microsoft Office Word – для написания документации к

программному продукту;

* Firebase – будет использоваться в качестве

инструмента для администрирования Real Time Data Base;

Разработка проекта будет происходить на компьютерах со следующими параметрами:

Процессор: Intel® Core I5-11300h 4,3Gh, AMD Ryzen 5 5800U 4,3Ggh;

Объем оперативной памяти: 8-16 GB;

Объем места на жестком диске: 256-1024 GB;

Видеокарта: NVIDIA MX 450, NVIDIA RTX 3050, Intel Iris XE Graphics;

ОС: Windows 10-11.

**2. Проектирование задачи**

* 1. **Разработка структуры приложения, системы меню, навигации**



Рисунок 2- Система меню

* 1. **Разработка UML-диаграмм**

**2.2.1. Разработка ДВИ ( диаграммы вариантов использования)**

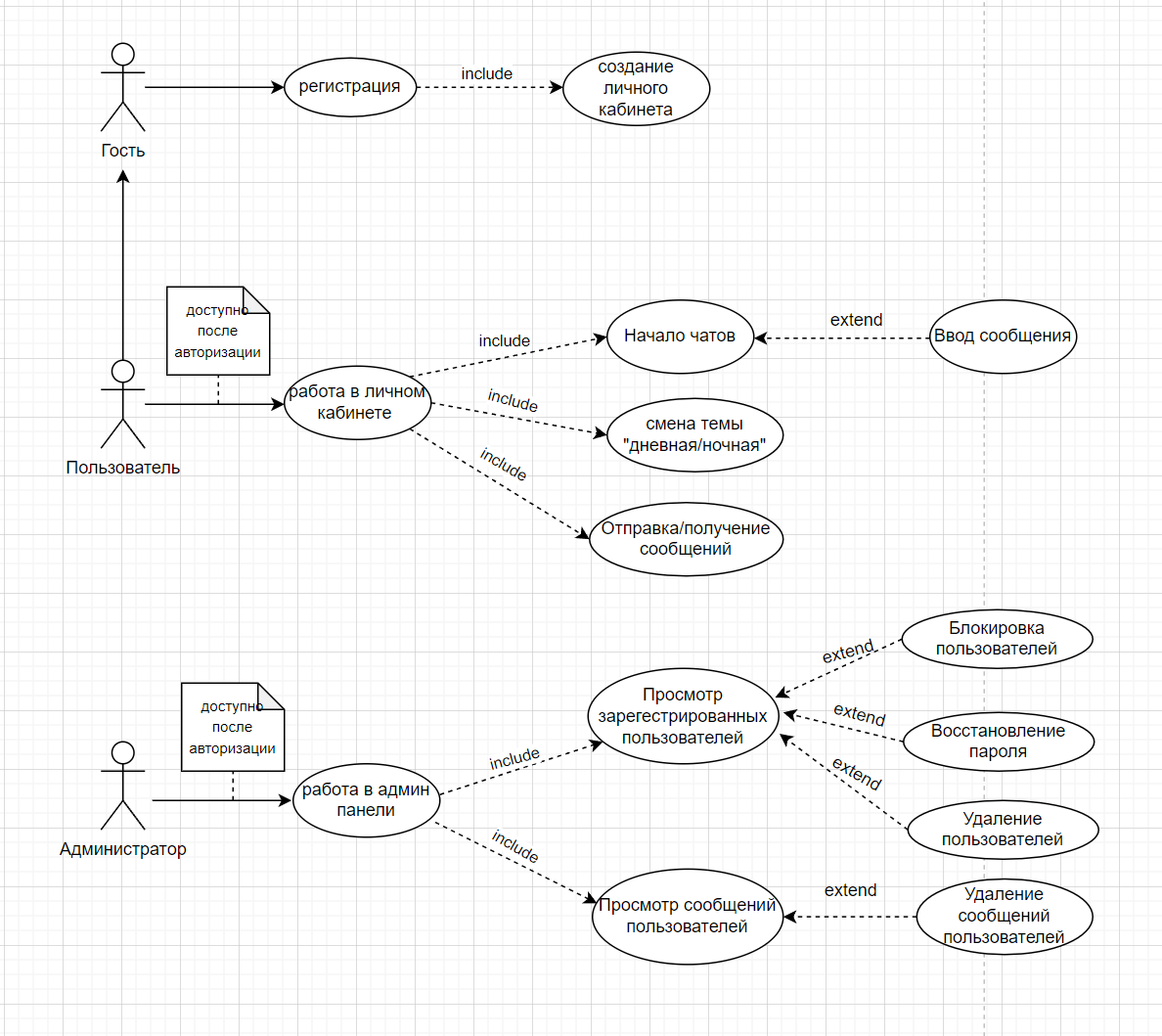


Рисунок 3- Диаграмма вариантов использования

* + 1. **Диаграмма деятельности**

Диаграмма деятельности — UML-диаграмма, на которой показаны действия, состояния которых описано на диаграмме состояний (рисунок 4).

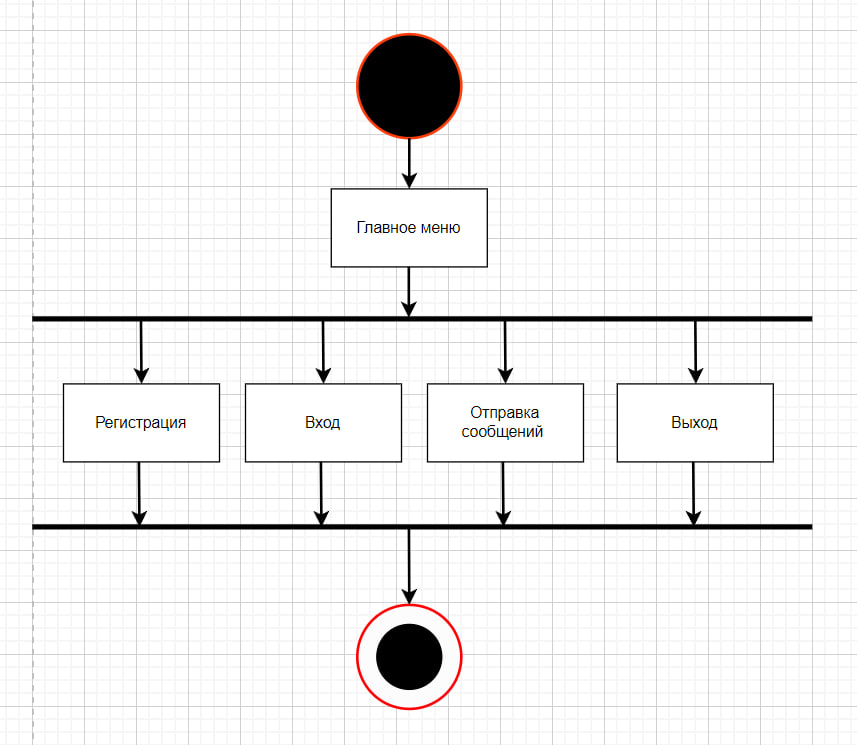


Рисунок 4- Диаграмма деятельности

* + 1. **Диаграмма классов**

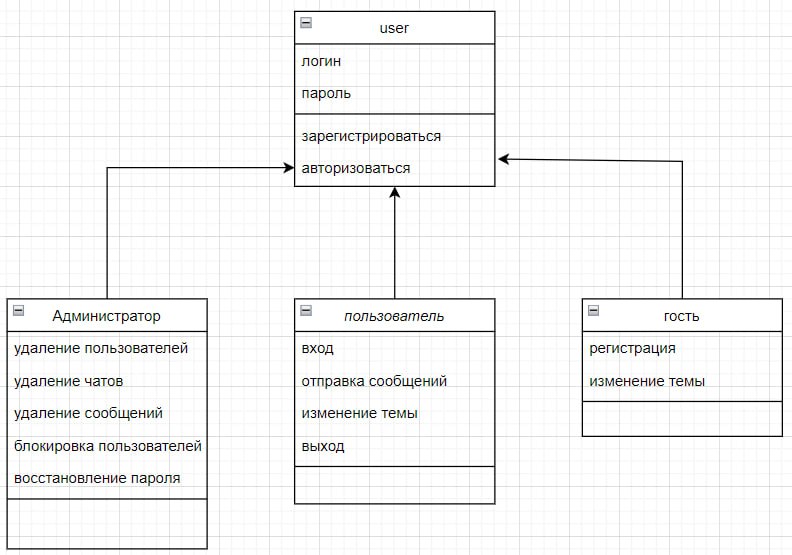


Рисунок 5- Диаграмма классов

* 1. **Разработка пользовательского интерфейса**

Важным элементом проектирования данного программного продукта является описание внешнего интерфейса разрабатываемого интернет-ресурса.

Для разработки визуального дизайн использовались сдержанные, мягкие цвета для удобства использования программного продукта.

В ходе разработки был спроектировано мобильное приложение “DIX”.

Для организации эффективной работы пользователя нужно создать целостный программный продукт данной предметной области, в котором все компоненты будут сгруппированы по функциональному назначению. При этом необходимо обеспечить удобный графический интерфейс пользователя. Интернет-ресурс должен позволить пользователю решать задачи, затрачивая значительно меньше усилий, чем при работе с разрозненными объектами. Все исходные данные будут разделены на несколько групп.

Прототип – это наглядная модель пользовательского интерфейса. В сущности, это «черновик» созданный на основе представления разработчика о потребностях пользователя. Итоговое отображение программы может отличаться от прототипа. C прототипами UX/UI можно ознакомиться в приложении А.

**3. Реализация**

Данный программный продукт был разработан с помощью среды разработки мобильных приложений–Android Studio. Удобный и понятный в использовании, с большим функциональном и готовых шаблонов. В данной базе хранится вся информация приложения.

Функциональность Android Studio включает в себя:

1. Редактор кода: Android Studio предоставляет мощный редактор кода с поддержкой автодополнения, проверки ошибок, рефакторинга и других инструментов для удобной работы с кодом.

2. Дизайнер пользовательского интерфейса: Инструменты для создания пользовательского интерфейса приложения с помощью графического интерфейса и возможности предварительного просмотра интерфейса на различных устройствах.

3. Отладка: Встроенные инструменты для отладки приложений, включая возможность установки точек останова, отслеживание значений переменных, анализ стека вызовов и другие инструменты отладки.

4. Эмуляторы и устройства: Возможность запуска и тестирования приложений на встроенных эмуляторах или подключенных устройствах.

5. Интеграция с системами контроля версий: Возможность интеграции с системами контроля версий, такими как Git, для управления исходным кодом проекта.

6. Поддержка языков программирования: Android Studio поддерживает различные языки программирования, включая Java, Kotlin и C++.

Для создания примеров работ использовалась такие программы Adobe Illustrator 2023, Adobe XD, Adobe Photoshop 2024.

Также в программном продукте ”DIX” для выбора чата, отправки и получения сообщений была использована база данных Firebase.

Преимущества Firebase включают в себя:

1. Расширенная функциональность: Firebase предоставляет широкий спектр инструментов и сервисов, таких как аутентификация, база данных в реальном времени, облачное хранилище, облачные функции, уведомления и многое другое, что позволяет разработчикам создавать полнофункциональные приложения.

2. Простота использования: Firebase предлагает простой и интуитивно понятный интерфейс, что делает его доступным для разработчиков всех уровней.

3. Масштабируемость: Firebase обеспечивает масштабируемость приложений, позволяя им расти по мере необходимости без необходимости переписывать код или изменять архитектуру.

4. Высокая доступность и надежность: Firebase предлагает высокую доступность и надежность своих сервисов благодаря облачной инфраструктуре Google.

5. Интеграция с другими сервисами Google: Firebase интегрируется с другими сервисами Google, такими как Google Analytics, AdMob, Google Cloud Platform и другими, что позволяет разработчикам получить доступ к широкому спектру инструментов для улучшения своих приложений.

6. Бесплатный уровень использования: Firebase предоставляет бесплатный уровень использования для многих своих сервисов, что позволяет начать использовать их без затрат на начальном этапе разработки.

**4.** [**Тестирование**](#_Достоинства_и_недостатки)

* 1. **Тесты на использование**

При разработке мобильного чат-приложения «DIX» многие возникающие ошибки и недоработки были исправлены на этапе реализации программного продукта. После завершения испытания реализации интернет-ресурса было проведено тщательное функциональное тестирование. Функциональное тестирование должно гарантировать работу всех элементов программного продукта в автономном режиме.

Таблица 5 – Тесты на использование

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Название теста** | **Действие** | **Ожидаемый результат** | **Физический результат** | **Результат тестирования** |
| Кнопка «Log in» | Переход на блок «Список чатов» | Переход на блок «Список чатов» | Переход на блок «Список чатов» | Выполнено |
| Кнопка «Sign Up» | Переход на блок «Регистрация» | Переход на блок «Регистрация» | Переход на блок «Регистрация» | Выполнено |
| Кнопка «Выход» | Выход из аккаунта | Выход из аккаунта | Выход из аккаунта | Выполнено |
| Кнопка «Отправить сообщение» | Отправка сообщения | Отправка сообщения | Отправка сообщения | Выполнено |

* 1. **Отчет о результатах тестирования**

В результате проведения тестирования выяснилось, что все ранее оговоренные функции и требования, были разработаны, а также протестированы. Тесты показали, что все функции работают правильно, следовательно, разработанный сайт можно выпускать.

В ходе тестирования программного обеспечения продукта на устройстве не было выявлено каких-либо ошибок, так как адаптивность интернет-ресурса была проведена на всех стадиях разработки.

**5. Руководство пользователя**

На блоке «Список чатов» можно увидеть перечень контактов(зарегистрированных пользователей) с которыми можно вести общение. (рисунок 8).



Рисунок 6- Список чатов

Чтобы начать чат с зарегистрированным пользователем вам необходимо нажать на имя пользователя (Кнопка выбора чатов), далее вас отправит на блок «Чат».

В случае если вам необходима выйти из вашего аккаунта необходимо нажать на кнопку «Выйти», далее вас отправит на блок «Входа в аккаунт».

На блоке «Чат» можно увидеть сообщения пользователей, имя пользователя. (рисунок 9).

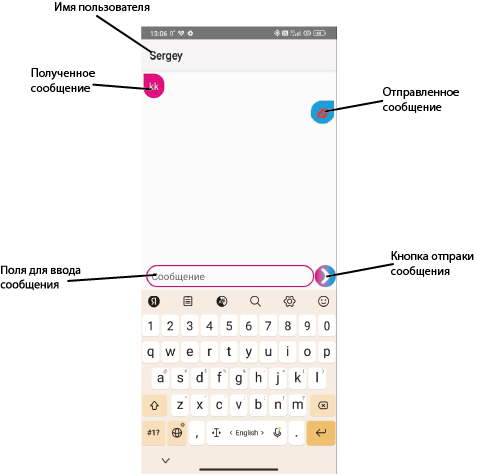
****

Рисунок 7- Чат

Для ввода сообщения, которое вы хотите написать, вам необходимо нажать на поле для ввода сообщения, ввести само сообщение или выбрать смайлик, который хотите отправить. Далее вам необходимо нажать на кнопку «Отправить сообщение», после чего ваше сообщение будет отправлено вашему собеседнику.

«Имя пользователя» - это имя человека, с которым вы ведёте общение в данный момент.

«Полученное сообщение» - это сообщение, отправленное пользователем вам.

«Отправленное сообщение» - это сообщение, отправленное вами пользователю.