

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	6
1 АНАЛИЗ ИСХОДНЫХ ДАННЫХ И ПОСТАНОВКА ЗАДАЧИ	7
1.1 Описание предметной области	7
1.2 Сравнительный анализ аналогов и прототипов	9
1.3 Постановка задачи проектирования	11
1.4 Выбор и обоснование средств и методов решения задач	12
1.5 Разработка технического задания	13
2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	14
2.1 Функциональная структура приложения	14
2.2 Проектирование диаграммы вариантов использования	15
2.3 Информационное обеспечение проекта	15
2.4 Разработка структуры конфигурации	16
2.5 Разработка концепции пользовательского интерфейса	22
2.6 Безопасность и защита данных	23
3 РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ	24
3.1 Результаты реализации функциональных частей	24
3.2 Разработка программной документации	29
3.3 Тестирование	29
4 ЭКОНОМИЧЕСКАЯ ЧАСТЬ	31
4.1 Обоснование необходимости вывода продукта на рынок	31
4.2 Структура (этапы) работ по созданию программного обеспечения	31
4.3 Составление сметы затрат на разработку программного обеспечения	33
4.4 Расчет экономического эффекта разработчика программного обеспечения	40
4.5 Выводы по экономической части	43
ЗАКЛЮЧЕНИЕ	44
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	45
ПРИЛОЖЕНИЕ А (обязательное). Техническое задание	46
ПРИЛОЖЕНИЕ Б (справочное). Диаграмма вариантов использования	50
ПРИЛОЖЕНИЕ В (справочное). База данных	51
ПРИЛОЖЕНИЕ Г (обязательное). Описание программы	52
ПРИЛОЖЕНИЕ Д (обязательное). Программа и методика испытаний	54
ПРИЛОЖЕНИЕ Е (обязательное). Руководство пользователя	58

					ФАА.502300.074 ПЗ		
Изм	Лист	№ докум.	Подпись	Дата			
Разраб.		Филиппов А.А.			Мессенджер «Fly» с шифрованием типа «end-to-end»	Лит.	Лист
Провер.		Розулев В.С.					Листов
Реценз.		Богущ Р.П.					5
Н. Контр.		Михнович О.В.					62
Утверд.		Чертков В.М.				Учреждение образования «Полоцкий государственный университет», гр.18-ИТ-2	

ВВЕДЕНИЕ

Общение всегда являлось неотъемлемой частью жизни человечества. Каждый день люди общаются с коллегами, друзьями и родственниками. С появлением мобильной связи стало доступно общение не только вживую, но и на расстоянии, посредством звонков и SMS-сообщений. А с появлением интернета многим открылась возможность делать это быстрее, используя при этом мессенджеры.

Мессенджер – это специальное приложение или программа, которую скачивают и устанавливают на смартфон или компьютер. Его основная цель – это мгновенный обмен текстовыми сообщениями с друзьями, родственниками, знакомыми, коллегами по работе или по учебе. Также можно совершать звонки при помощи аудио или видеосвязи.

Слово «мессенджер» произошло от английского «messenger», что означает курьер. Обмен сообщениями идет мгновенно, в режиме реального времени. Сообщения отправляются собеседнику сразу после того, как отправитель закончит ввод, редактирование и нажмет на кнопку отправки. Но при этом получатель сообщения должен быть на связи, иначе сообщение вынуждено будет ждать, пока пользователь тоже запустит свой мессенджер и обратит на него внимание [11].

Когда люди общаются в интернете, казалось бы, только собеседники участвуют в общении. Однако, как показывают примеры прошлого, существует немало лиц, которое пытаются шпионить за общением людей в интернете. Это может быть корпоративное наблюдение или правительственные организации, которые следят не за конкретным пользователем, а за всеми подряд.

Данный мессенджер создан именно для того, чтобы защитить от шпионажа абсолютно каждого пользователя. В нем можно отправлять сообщения и не бояться, что кто-то узнает содержимое написанного.

Мессенджер «Fly» с шифрованием данных – это удобный, и в то же время, надежный ресурс для пользователей. Данный мессенджер – это площадка, где люди могут общаться, не переживая за конфиденциальность своих данных и свободно выражая свои мысли.

1 АНАЛИЗ ИСХОДНЫХ ДАННЫХ И ПОСТАНОВКА ЗАДАЧИ

1.1 Описание предметной области

Всё чаще для общения используются мессенджеры. Данные программы просты в использовании, понятны людям всех возрастных категорий, а также имеют высокий уровень защиты данных.

Все мессенджеры делятся на два типа:

- универсальные;
- корпоративные.

Уровень защиты данных зависит от алгоритма шифрования, который используется разработчиками при написании программы. Выделяют следующие алгоритмы шифрования:

- симметричный – алгоритм, при котором для шифрования и дешифрования используется один и тот же ключ. При этом важно учесть, что ключ должен быть установлен на обоих устройствах до начала обмена данными.

- асимметричный. Такое шифрование предпочитает двухключевой подход. Открытый ключ используется для шифрования, а закрытый – для дешифрования. При этом стоит отметить, что последний доступен только личному устройству пользователя, а потому не стоит переживать по поводу сохранности первого, ибо никто не сможет получить доступ к личной информации без закрытого ключа.

- AES (Advanced Encryption Standard). Один из самых безопасных алгоритмов симметричного шифрования, который существует сегодня. Он использует шифр размером блока 128 бит и длиной ключа 128, 192 и 256 бит. Алгоритм AES используют многие организации, включая правительство США.

- Triple DES. Ещё один симметричный алгоритм шифрования. Когда исходный стандарт шифрования данных DES стал уязвим для атак, его заменили на Triple DES, название которого происходит от использования трёх 56-битных ключей. Однако в последние годы этот алгоритм вышел из моды, так как появились более безопасные варианты, такие как AES.

- Blowfish. Очередной симметричный алгоритм шифрования, который был изобретён экспертом по безопасности Брюсом Шнайером (Bruce Schneier). Как и Triple DES, он появился на замену устаревшему DES. Это – блочный шифр с симметричным ключом, который входит в число самых безопасных алгоритмов. Более того, он является общественным достоянием и может свободно использоваться кем угодно.

- Twofish. Симметричный алгоритм шифрования, который использует ключи длиной до 256 бит. Данное изобретение также было создано Брюсом Шнайером. По сути, Twofish – это продвинутый вариант Blowfish. Он не только быстрый и безопасный, но ещё и находится в открытом доступе, как и его предшественник.

					ФАА.502300.074 ПЗ	Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

– RSA. Алгоритм асимметричного шифрования, который является одним из старейших и наиболее широко используемых стандартов для шифрования данных в интернете. Этот алгоритм не отличается скоростью, но считается очень безопасным.

– E2EE. Так называют сквозное шифрование. Этот способ передачи данных гарантирует, что доступ к исходному тексту сообщения имеется только у отправителя и получателя. То есть, даже интернет-провайдеры или другие поставщики услуг не смогут расшифровать пользовательскую информацию. E2EE идеально подходит для тех, кто заботится о конфиденциальности и безопасности. Говорить о полной безопасности при использовании сквозного шифрования нельзя, но её уровень достаточно велик.

– облачное шифрование. Облачное хранилище – одно из самых распространённых мест для хранения фотографий, видеоматериалов и документов. А потому эта служба нуждается в шифровании для защиты конфиденциальных файлов. Многие поставщики облачных хранилищ предлагают ту или иную его форму.

– шифрование диска. Инструмент защиты данных, который шифрует содержимое жёсткого диска. Новый уровень безопасности. Есть несколько способов шифрования диска: с помощью программного или аппаратного обеспечения. В последнем случае используется диск с самошифрованием. Он автоматически шифрует и дешифрует данные, что делает его одним из самых простых способов обезопасить личные файлы. К тому же на рынке представлен большой ассортимент самошифруемых дисков. Единственный минус – высокая стоимость.

С каждым годом растёт потребность в защите данных, и пользователи всё чаще предпочитают мессенджеры, использующие сквозное шифрование. Например, Telegram гарантирует, что вся информация поступает на сервер уже зашифрованной.

Почему шифрование настолько важно? Тут стоит отметить несколько пунктов:

– конфиденциальность. Шифрование является важным инструментом для тех, кто хочет сохранить личные данные только для себя и предполагаемых получателей. Ведь некоторые разговоры, фотографии, видеофайлы и документы лучше скрывать от посторонних глаз.

– безопасность. В период пандемии многие пользователи были вынуждены вводить конфиденциальную информацию в удалённом режиме. Шифрование убережёт данные кредитной карты или личную информацию от кражи.

Сейчас мессенджеры выполняют кучу ролей и давно ушли от базовой отправки сообщений. С помощью мессенджеров можно писать текстовые сообщения, отправлять фотографии, аудио, видео, документы, делиться геолокацией, создавать публичные каналы и обсуждения, а бизнесу – общаться с клиентами [10].

1.2 Сравнительный анализ аналогов и прототипов

Прежде, чем приступить к реализации дипломного проекта, следует проанализировать существующие на данный момент программные продукты подобного характера и схожей тематикой. Данный анализ поможет сделать верные выводы о том, как необходимо создавать собственный программный продукт на основе достоинств и недостатков рассмотренных аналогов.

«WhatsApp Messenger» – приложение, которое пользуется популярностью у пользователей, так как работает на всех доступных мобильных платформах. Внешний вид представлен на рисунке 1.1. Данное приложение позволяет обмениваться всеми видами сообщений, анализирует список контактов пользователя и помечает пользователей, которые тоже пользуются приложением «WhatsApp». Приложение полностью синхронизируется на разных типах устройств. Можно создавать индивидуальные и групповые чаты (правда с ограничением до 256 пользователей). Рассмотрим положительные аспекты данного мессенджера:

- большая часть контактов из телефонной книги будет в «WhatsApp», так как этот мессенджер появился достаточно давно;
- интуитивно понятный и простой интерфейс;
- приложение при установке сразу отмечает пользователей из списка контактов, у которых установлен мессенджер;
- достаточно стабильно работает и предоставляет весь основной набор функций.

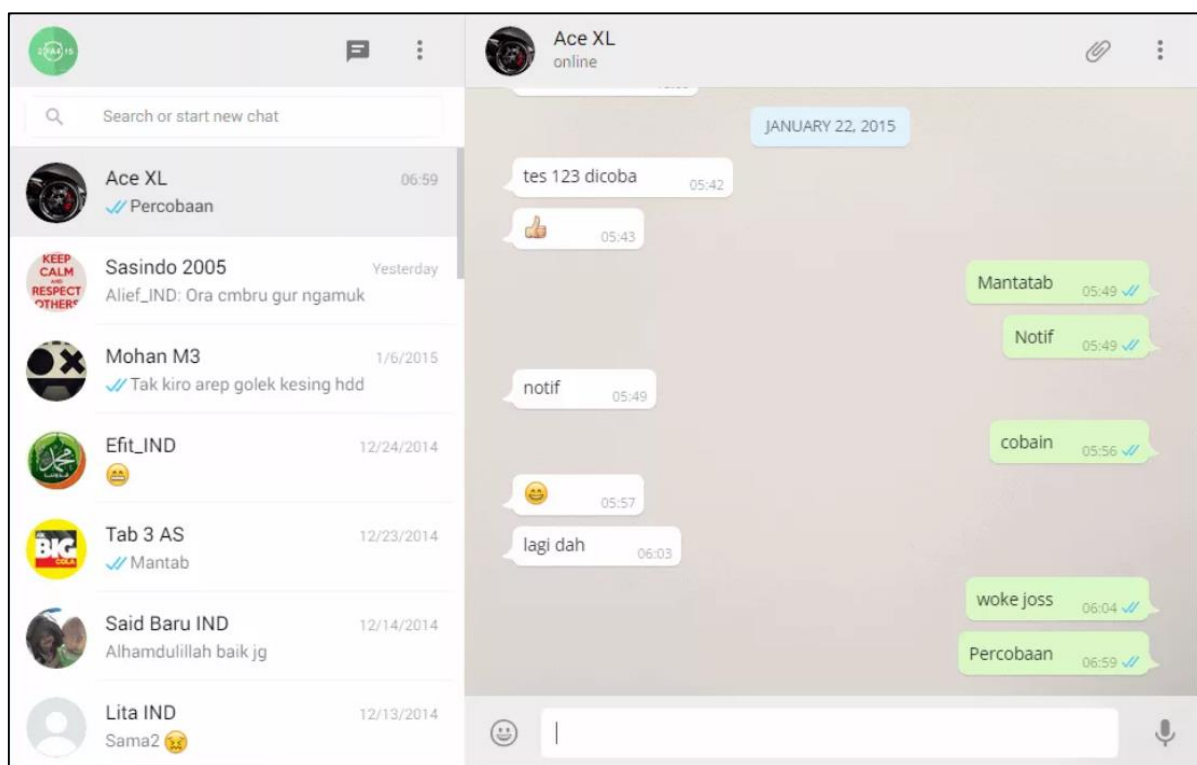


Рисунок 1.1 – Интерфейс «WhatsApp Web»

Изм.	Лист	№ докум.	Подпись	Дата

ФАА.502300.074 ПЗ

Лист

9

Далее рассмотрим недостатки:

- данный мессенджер выкупил «Facebook», поэтому теперь все данные пользователей собираются и используются с рекламными целями;
- если у смартфона села батарея, то десктопная версия также перестанет работать.

«Viber» – это мессенджер, который уже долгое время существует на рынке коммуникационных продуктов. Главные его преимущества такие:

- с помощью функции «Viber Out» можно звонить на телефоны, на которых не установлено это приложение;
- видеоконференции с ограничением до 20 участников;
- мессенджер установлен у достаточно большого числа пользователей.

Из недостатков нужно выделить:

- не очень высокий уровень безопасности данных: пользователям постоянно приходит спам и нежелательная реклама;
- не сохраняется история переписок, если войти в профиль с другого устройства или установить «Viber» на новом смартфоне.



Рисунок 1.2 – Внешний вид «Viber Desktop»

Перейдем к рассмотрению мессенджера «Fly». Из достоинств можно выделить:

- комплексная защита данных шифрованием «end-to-end»;
- простота использования (мессенджер не перегружен ненужными функциями);
- стабильное функционирование;
- сохранение истории переписок при входе с другого устройства.

Выявлены следующие минусы:

- наличие только веб-версии;
- отсутствие возможности записи и передачи видео.

По результатам анализа аналогов были выявлены плюсы и минусы, которые позволили сформулировать функциональные требования к проектируемому мессенджеру, представленные в разделе 3.

Сравнение характеристик аналогов представлено в таблице 1.1.

Таблица 1.1 – Сравнение аналогов и разрабатываемого продукта

Наименование программного продукта	UI/UX	Возможность повторного скачивания вложения	Сквозное шифрование	Наличие рекламы	Сохранение истории переписок
«WhatsApp»	–	+	+	+	+
«Viber»	+	–	+	+	–
«Fly»	+	+	+	–	+

По результатам анализа аналогов были выявлены плюсы и минусы, которые позволили сформулировать функциональные требования к проектируемому мессенджеру, представленные в техническом задании приложения А.

1.3 Постановка задачи проектирования

В рамках преддипломной практики осуществлялась разработка мессенджера с шифрованием типа «end-to-end».

На сегодняшний день пользователи во всем мире сталкиваются с такой проблемой как незащищенность данных. Так как мессенджер «Fly» оснащен достаточно хорошим шифрованием, пользователи могут спокойно отправлять важную информацию без боязни, что эти данные окажутся в руках у мошенника или любого другого нежелательного лица.

Для грамотной реализации мессенджера требуется изучить: в каком формате пользователям будет удобнее общаться, как осуществить шифрование данных и каким образом следует хранить зашифрованную информацию.

Функциональное назначение разрабатываемого мессенджера состоит в защите данных от кражи, а также защите пользователей от шпионажа.

К основным функциям мессенджера относятся:

- обмен текстовыми сообщениями;
- создание группы пользователей для общения по интересам, по работе, по учебе – коллективное общение и даже дистанционное обучение;
- возможность обмена фото, файлов различного типа;

- отображение статуса собеседника;
- расходование трафика интернета.

Для удобства использования и конкурентоспособности мессенджера необходимо выделить требования, которые послужат ориентирами при написании:

- неперегружаемость – мессенджер не должен быть перегружен большим объемом информации, иначе пользователь будет теряться и не понимать, какие действия он может совершать;
- удобство использования – пользователь должен производить минимальное количество действий, при этом мессенджер должен содержать в себе оптимальное число настроек;
- переносимость – легкость в переносе приложения или данных с одного компьютера на другой без потерь функциональности;
- стабильность работы – при возникновении критических ошибок, мессенджер должен оповещать пользователя о сбоях и их причинах с возможностью корректного завершения работы;
- устойчивость к нагрузкам – при выполнении обработки большого объема данных мессенджер должен стабильно функционировать.

1.4 Выбор и обоснование средств и методов решения задач

Для реализации задачи по проектированию мессенджера «Fly» выбор пал на СУБД PostgreSQL 14 [3]. PostgreSQL – свободная объектно-реляционная система управления базами данных, являющаяся «open-source» проектом. На сегодняшний день именно эта СУБД является одной из востребованных и популярных систем управления и обслуживания баз данных. В ней применяются новейшие разработки и последние достижения в области проектирования, построения и обслуживания баз данных.

Для разработки мессенджера используется PyCharm [4] и Django [5].

PyCharm помогает писать красивый код, который легко поддерживать. IDE контролирует качество кода с помощью проверок соответствия требованиям PEP8, умных рефакторингов и множества инспекций, а также оказывает помощь при тестировании. PyCharm создается программистами для программистов, поэтому в нем есть все необходимое для продуктивной разработки на Python.

Django – свободный фреймворк для веб-приложений на языке Python, использующий шаблон проектирования MVC. Проект поддерживается организацией Django Software Foundation.

Django проектировался для работы под управлением Apache с модулем mod python и с использованием PostgreSQL в качестве базы данных. С включением поддержки WSGI, Django может работать под управлением FastCGI, mod wsgi, или SCGI на Apache и других серверах (lighttpd, nginx),

					ФАА.502300.074 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		12

сервера uWSGI. В составе Django присутствует собственный веб-сервер для разработки. Сервер автоматически определяет изменения в файлах исходного кода проекта и перезапускается, что ускоряет процесс разработки на Python.

Языком написания приложения был выбран Python. Данный язык и технологии являются мощным средством для написания мессенджера.

Python – высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ. Язык является полностью объектно-ориентированным в том плане, что всё является объектами. Синтаксис ядра языка минималистичен, за счёт чего на практике редко возникает необходимость обращаться к документации. Сам же язык известен как интерпретируемый и используется в том числе для написания скриптов [6].

1.5 Разработка технического задания

На основании анализа сущности проблемы и исходных данных были прописаны следующие требования к функциональности разрабатываемого мессенджера:

- регистрация пользователя в мессенджере с указанием необходимой информации в аккаунте;
- редактирование и удаление аккаунта из мессенджера;
- индивидуальная настройка мессенджера;
- общение с другими пользователями;
- добавление файлов к сообщению;
- просмотр статуса собеседника;
- защита всех данных пользователей посредством сквозного шифрования.

Расширенное техническое задание для проектируемого мессенджера представлено в приложении А.

2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

2.1 Функциональная структура приложения

Мессенджер «Fly» предназначен в большей степени для пользователей, которые хотят защитить от утечки свои данные и себя от шпионажа другими лицами.

Помимо защиты от стороннего вмешательства и предоставления конфиденциальности для пользователей мессенджер должен обладать интуитивно понятным интерфейсом, оптимальными настройками и бесперебойной работой.

Таким образом, исходя из приведенных характеристик, учитывая при этом требования технического задания (приложение А), была спроектирована функциональная структура разрабатываемого мессенджера, представленная на рисунке 2.1.

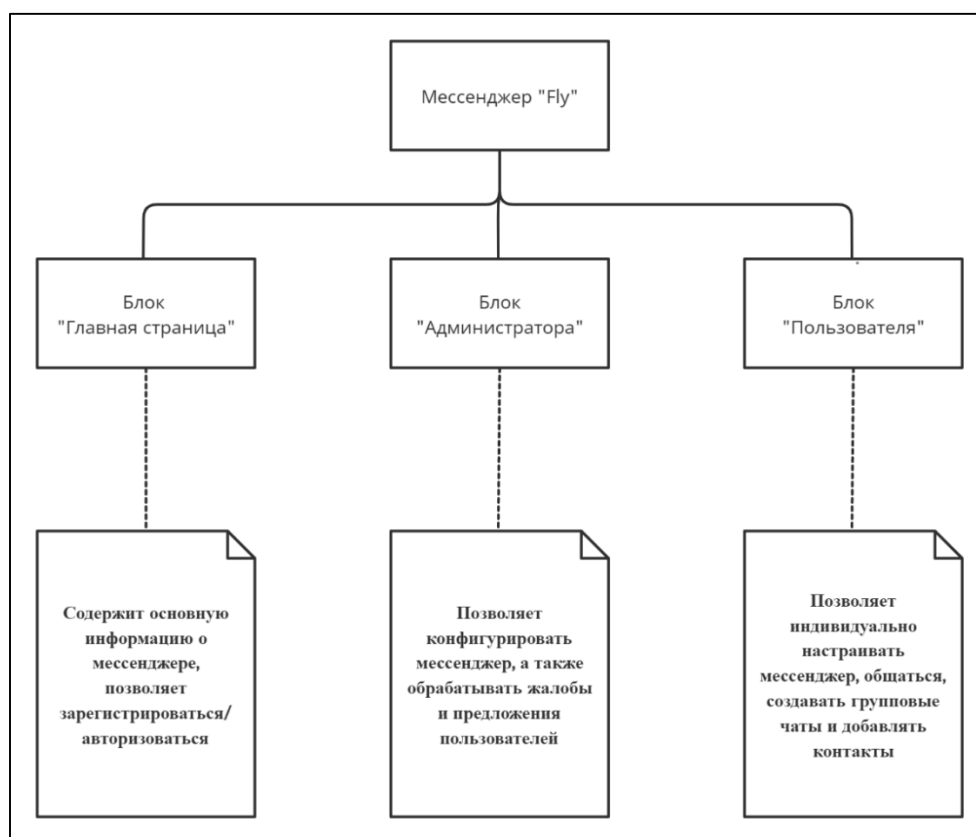


Рисунок 2.1 – Функциональная структура мессенджера

Блок Главная страница содержит основную информацию о мессенджере, а также именно в этом блоке расположены кнопки регистрация/авторизация.

Блок Администратора позволяет конфигурировать мессенджер, а также обрабатывать жалобы и предложения пользователей.

Блок Пользователя позволяет индивидуально настраивать мессенджер, общаться, создавать групповые чаты и добавлять контакты.

Изм.	Лист	№ докум.	Подпись	Дата

ФАА.502300.074 ПЗ

Лист

14

2.2 Проектирование диаграммы вариантов использования

Визуальным моделированием (visual modeling) называется способ представления идей и проблем реального мира с помощью моделей. Модель помогает понять проблему всем участникам, задействованным в реализации проекта на различных этапах: заказчику, эксперту, аналитику, проектировщику, автору документации, программисту и др. Моделирование обеспечивает более точную оценку необходимых ресурсов, четкую проработку планов и эффективное функционирование создаваемых систем [1].

Диаграмма вариантов использования (сценариев поведения, прецедентов) является исходным концептуальным представлением системы в процессе ее проектирования и разработки. Данная диаграмма состоит из актеров, вариантов использования и отношений между ними. При построении диаграммы могут использоваться также общие элементы нотации: примечания и механизмы расширения.

Суть данной диаграммы состоит в следующем: проектируемая система представляется в виде множества актеров, взаимодействующих с системой с помощью так называемых вариантов использования. При этом актером (действующим лицом, актантом, актором) называется любой объект, субъект или система, взаимодействующая с моделируемой системой извне. В свою очередь вариант использования – это спецификация сервисов (функций), которые система предоставляет актеру. Другими словами, каждый вариант использования определяет некоторый набор действий, совершаемых системой при взаимодействии с актером. При этом в модели никак не отражается то, каким образом будет реализован этот набор действий [9].

Для мессенджера «Fly» можно выделить двух актеров – это администратор и пользователь (приложение Б).

Администратор конфигурирует работу мессенджера, а также обрабатывает жалобы и предложения пользователей.

Пользователь индивидуально настраивает мессенджер, может общаться с другими пользователями, создавать групповые чаты и добавлять контакты.

2.3 Информационное обеспечение проекта

В состав информационного обеспечения проекта на различных этапах разработки входят семейство стандартов UML, PostgreSQL, Django и Python. Методология стандартов IDEF на этапе проектирования применялась для моделирования информационной системы при помощи CASE-средства ERwin Data Modeler r7.2. СУБД PostgreSQL 14, язык программирования Python и фреймворк Django применяются для создания приложения.

					ФАА.502300.074 ПЗ	Лист
						15
Изм.	Лист	№ докум.	Подпись	Дата		

2.4 Разработка структуры конфигурации

При проектировании и разработке базы данных требуется выбрать оптимальное количество таблиц для хранения данных, полей, входящих в ту или иную таблицу, а также планирование отношений между таблицами. Для решения данного рода задач и используется нормализация.

Нормализация баз данных заключается в приведении структуры хранения данных к нормальным формам (NF). Всего таких форм существует 8, но часто достаточным является соблюдение первых трех. Рассмотрим их более подробно на примере учебной базы данных. Примеры будут строиться по принципу «что было бы, если было иначе, чем сейчас».

Первые три нормальные формы можно описать следующим образом:

- основным правилом 1NF является необходимость неделимости значения в каждом поле (столбце) строки – атомарность значений. Строки таблиц не должны зависеть друг от друга, т.е. первая запись не должна влиять на вторую и наоборот, вторая на третью и т.д. Размещение записей в таблице не имеет никакого значения. Аналогичная ситуация со столбцами записей. Их порядок не должен влиять на понимание информации. Каждая строка должна быть уникальна, поэтому для нее определяется первичный ключ, состоящий из одного либо нескольких полей (составной ключ). Первичный ключ не может повторяться в пределах таблицы и служит идентификатором записи;

- отношение находится во 2NF, если оно находится в 1NF и каждый не ключевой атрибут неприводимо зависит от первичного ключа. Неприводимость означает, что в составе потенциального ключа отсутствует меньшее подмножество атрибутов;

- 3NF схожа по логике с 2NF, но с некоторым отличием. Если 2 форма ликвидирует зависимости не ключевых полей от части ключа, то третья нормальная форма исключает зависимость не ключевых полей от других не ключевых полей.

Реляционная база данных – это совокупность взаимосвязанных таблиц, каждая из которых содержит информацию об объектах определенного типа. Строка таблицы содержит данные об одном объекте (например, товаре, клиенте), а столбцы таблицы описывают различные характеристики этих объектов – атрибутов (например, наименование, код товара, сведения о клиенте). Записи, то есть строки таблицы, имеют одинаковую структуру – они состоят из полей, хранящих атрибуты объекта. Каждое поле, то есть столбец, описывает только одну характеристику объекта и имеет строго определенный тип данных. Все записи имеют одни и те же поля, только в них отображаются различные информационные свойства объекта [7].

Схема базы данных мессенджера «Fly» представлена в приложении В.

В ходе проектирования базы данных были выявлены следующие основные сущности:

					ФАА.502300.074 ПЗ	Лист
						16
Изм.	Лист	№ докум.	Подпись	Дата		

Сущность Пользователи представляет собой список пользователей мессенджером. Характеризуется номером телефона, E-mail, паролем, ФИО, статусом, настройками, а также датой создания и редактирования профиля.

Сущность Верификация представляет собой проверку на вход. Характеризуется идентификационным номером пользователя, верификационным кодом и временем верификации.

Сущность Доступ представляет собой доступ пользователя к аккаунту. Характеризуется идентификационным номером, идентификационным номером пользователя, токеном, датой получения доступа и датой отключения доступа.

Сущность Чёрный список представляет собой список всех заблокированных пользователей. Характеризуется идентификационным номером, идентификатором пользователя, идентификатором собеседника и датой занесения в чёрный список.

Сущность Участники представляет собой список участников диалога. Характеризуется идентификационным номером, идентификационным номером диалога, идентификационным номером пользователя, типом участника, датой добавления и датой изменения участника.

Сущность Диалог представляет собой информацию о диалоге. Характеризуется идентификационным номером, названием, идентификационным номером создателя диалога, идентификационным номером канала, датой создания, датой изменения и датой удаления диалога.

Сущность Удаленный диалог представляет собой список удаленных диалогов. Характеризуется идентификационным номером, идентификационным номером диалога, идентификационным номером пользователя, датой создания диалога.

Сущность Жалобы представляет собой список отчетов о нарушениях. Характеризуется идентификационным номером, идентификационным номером пользователя, идентификационным номером участника диалога, типом жалобы, примечанием, статусом и датой подачи жалобы.

Сущность Сообщения пользователей представляет собой список сообщений пользователей. Характеризуется идентификационным номером, 128-битным идентификационным номером, идентификационным номером диалога, идентификационным номером отправителя, типом сообщения, содержимым, датой отправления и датой удаления сообщения.

Сущность Удалённые сообщения представляет собой список удалённых сообщений. Характеризуется идентификационным номером, идентификационным номером сообщения, идентификационным номером пользователя, датой отправления и датой изменения сообщения.

Сущность Вложения представляет собой список отправленных файлов пользователем в сообщениях. Характеризуется идентификационным номером, идентификационным номером сообщения, ссылкой на файл, датой отправления и датой изменения вложения.

Сущность Контактные данные представляет собой информацию о контакте. Характеризуется идентификационным номером, ФИО, номером телефона, E-mail, датой создания аккаунта.

Сущность Контакт представляет собой пользователя, который добавлен в записную книжку. Характеризуется идентификационным номером пользователя, идентификационным номером контактных данных, ФИ, датой добавления и датой изменения контакта.

Далее будут описаны все таблицы, созданные в базе данных мессенджера «Fly».

Таблица Users содержит информацию о пользователе. Ее структура приведена в таблице 3.1.

Таблица 3.1 – Характеристика атрибутов таблицы Users

Имя атрибута	Тип	Описание
Id	int	Идентификатор пользователя. Ключевой атрибут
Phone	varchar	Номер телефона
Email	varchar	Электронный адрес
Password	varchar	Пароль
First_name	varchar	Имя
Last_name	varchar	Фамилия
Middle_name	varchar	Отчество
Is_active	bool	Статус «Активен»
Is_reported	bool	Статус «С жалобами»
Is_blocked	bool	Статус «Заблокирован»
Preferences	text	Настройки
Created_at	datetime	Дата создания аккаунта
Updated_at	datetime	Дата изменения аккаунта

Таблица User_verification содержит информацию о верификации пользователя. Ее структура приведена в таблице 3.2.

Таблица 3.2 – Характеристика атрибутов таблицы User_verification

Имя атрибута	Тип	Описание
Users_id	int	Идентификатор пользователя
Verification_code	varchar	Верификационный код
Created_at	varchar	Дата выдачи кода

Таблица Access содержит информацию о получении доступа пользователем. Ее структура приведена в таблице 3.3.

Таблица 3.3 – Характеристика атрибутов таблицы Access

Имя атрибута	Тип	Описание
Id	int	Идентификатор доступа. Ключевой атрибут
Users_id	int	Идентификатор пользователя
Token	varchar	Токен
Created_at	datetime	Дата получения доступа
Deleted_at	datetime	Дата закрытия доступа

Таблица Block_list содержит список заблокированных пользователей. Ее структура приведена в таблице 3.4.

Таблица 3.4 – Характеристика атрибутов таблицы Block_list

Имя атрибута	Тип	Описание
Id	int	Идентификатор блокировки. Ключевой атрибут
Users_id	int	Идентификатор пользователя
Participants_id	int	Идентификатор собеседника
Created_at	datetime	Дата блокировки

Таблица Participants содержит информацию об участниках диалога. Ее структура приведена в таблице 3.5.

Таблица 3.5 – Характеристика атрибутов таблицы Participants

Имя атрибута	Тип	Описание
Id	int	Идентификатор участника. Ключевой атрибут
Conversation_id	int	Идентификатор диалога
Users_id	int	Идентификатор пользователя
Type	enum	Роль в диалоге
Created_at	datetime	Дата добавления
Updated_at	datetime	Дата изменения

Таблица Conversation содержит информацию о диалогах, создаваемых пользователями в мессенджере. Также здесь содержится информация о создателе беседы, дате удаления и дате изменения диалога. Ее структура приведена в таблице 3.6.

Таблица 3.6 – Характеристика атрибутов таблицы Conversation

Имя атрибута	Тип	Описание
Id	int	Идентификатор диалога. Ключевой атрибут
Title	nvarchar	Название диалога
Creator_id	int	Идентификатор создателя диалога
Channel_id	varchar	Идентификатор канала
Created_at	datetime	Дата создания диалога
Updated_at	datetime	Дата изменения диалога
Deleted_at	datetime	Дата удаления диалога

Таблица Reports содержит список жалоб на пользователей. Ее структура приведена в таблице 3.7.

Таблица 3.7 – Характеристика атрибутов таблицы Reports

Имя атрибута	Тип	Описание
Id	int	Идентификатор жалобы. Ключевой атрибут
Users_id	int	Идентификатор пользователя
Participants_id	int	Идентификатор участника
Report_type	varchar	Тип жалобы
Notes	text	Примечание
Status	enum	Статус жалобы
Created_at	datetime	Дата отправления жалобы

Таблица Deleted_conversations содержит информацию об удаленных диалогах. Ее структура приведена в таблице 3.8.

Таблица 3.8 – Характеристика атрибутов таблицы Deleted_conversations

Имя атрибута	Тип	Описание
Id	int	Идентификатор удаленного диалога. Ключевой атрибут
Conversation_id	int	Идентификатор беседы
Users_id	int	Идентификатор пользователя
Created_at	datetime	Дата создания диалога

Таблица Messages содержит список сообщений, отправленных пользователем. Ее структура приведена в таблице 3.9.

Таблица 3.9 – Характеристика атрибутов таблицы Messages

Имя атрибута	Тип	Описание
Id	int	Идентификатор сообщения. Ключевой атрибут
Guid	varchar	128-битный идентификатор
Conversation_id	int	Идентификатор диалога
Sender_id	int	Идентификатор отправителя
Message_type	enum	Тип сообщения
Message	varchar	Текст сообщения
Created_at	datetime	Дата отправления
Deleted_at	datetime	Дата удаления

Таблица Deleted_messages содержит информацию об удаленных сообщениях. Ее структура приведена в таблице 3.10.

Таблица 3.10 – Характеристика атрибутов таблицы Deleted_messages

Имя атрибута	Тип	Описание
Id	int	Идентификатор удаленного сообщения. Ключевой атрибут
Messages_id	int	Идентификатор сообщения
Users_id	int	Идентификатор пользователя
Created_at	datetime	Дата добавления сообщения
Updated_at	datetime	Дата изменения сообщения

Таблица Attachments содержит информацию о файлах, которые пользователь может прикрепить к своим сообщениям. Ее структура приведена в таблице 3.11.

Таблица 3.11 – Характеристика атрибутов таблицы Attachments

Имя атрибута	Тип	Описание
Id	int	Идентификатор вложения. Ключевой атрибут
Messages_id	int	Идентификатор сообщения
Thumb_url	varchar	Ссылка на миниатюру файла
File_url	varchar	Ссылка на файл
Created_at	datetime	Дата добавления файла
Updated_at	datetime	Дата изменения файла

Таблица Contacts содержит контактные данные пользователя. Ее структура приведена в таблице 3.12.

Таблица 3.12 – Характеристика атрибутов таблицы Contacts

Имя атрибута	Тип	Описание
Id	int	Идентификатор контактных данных. Ключевой атрибут
First_name	varchar	Имя
Middle_name	varchar	Отчество
Last_name	varchar	Фамилия
Phone	varchar	Номер телефона
Email	varchar	Электронный адрес
Created_at	datetime	Дата создания аккаунта

Таблица User_contact содержит информацию о контакте, который добавил пользователь. Ее структура приведена в таблице 3.13.

Таблица 3.13 – Характеристика атрибутов таблицы User_contact

Имя атрибута	Тип	Описание
User_id	int	Идентификатор пользователя
Contact_id	int	Идентификатор контактных данных
First_name	varchar	Имя
Last_name	varchar	Фамилия
Created_at	datetime	Дата добавления контакта
Updated_at	datetime	Дата изменения контакта

Таким образом, при разработке базы данных мессенджера «Fly» было выявлено 13 сущностей, а также были спроектированы 13 таблиц, которые связаны между собой связями типа «один ко многим».

2.5 Разработка концепции пользовательского интерфейса

При создании мессенджера значимую роль играет правильный пользовательский интерфейс, который должен быть удобным в использовании, понятным, лаконичным и не перегруженным лишними информацией и функциями.

После анализа аналогов данной тематики были вынесены следующие пункты: главная страница, форма регистрации/авторизации, страница «Профиль».

Интерфейсом проектируемого мессенджера является веб-приложение, написанное на языке Python.

На главной странице размещена информация о назначении мессенджера, его идеях. Кроме этого, главная страница содержит кнопки регистрации/авторизации для пользователей.

Форма регистрации содержит поля, где требуется в обязательном порядке указать почту, логин, имя, фамилию, задать пароль создаваемому аккаунту.

Страница «Профиль» содержит основную информацию о пользователе, а также здесь происходят все основные действия пользователя.

2.6 Безопасность и защита данных

Для защиты данных зачастую используется протокол передачи данных HTTPS. Обеспечение безопасной передачи данных необходимо на сайтах, где вводится и передается конфиденциальная информация – на любых сайтах с авторизацией, взаимодействием с платежными системами, почтовыми сервисами. Шифрование таких данных позволит предотвратить их получение и использование третьими лицами.

Для реализации передачи данных посредством HTTPS на веб-сервере, обрабатывающем запросы от клиентов, должен быть установлен специальный SSL-сертификат. Есть сертификаты, защищающие только один домен. А есть сертификаты, которые обеспечивают защиту информации на всех поддоменах, и это Wildcard SSL.

Наличие SSL-сертификата является одним из факторов ранжирования Google, поэтому переход на протокол HTTPS повышает позиции в поисковой выдаче Google. Хотя на текущий момент этот фактор не имеет основополагающего значения, его влияние на поисковой результат может увеличиваться в будущем.

Обеспечение безопасности данных строится в основном на принципе, что целостность данных подразумевает недопустимость модификации информации пользователями или ее порчи вследствие сбоев или вирусных атак. Поэтому у пользователей есть доступ только к тем таблицам, которые им разрешено редактировать, в зависимости от их статуса.

При возникновении каких-либо ошибок в базе данных предусмотрен откат транзакций. В случае несоответствия вводимых данных в базе данных предусматривается отказ.

3 РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ

3.1 Результаты реализации функциональных частей

Django – это высокоуровневый Python веб-фреймворк, который позволяет быстро создавать безопасные и поддерживаемые веб-сайты. Созданный опытными разработчиками, Django берёт на себя большую часть хлопот веб-разработки, поэтому можно сосредоточиться на написании веб-приложения без необходимости изобретать велосипед. Он бесплатный и с открытым исходным кодом, имеет растущее и активное сообщество, отличную документацию и множество вариантов как бесплатной, так и платной поддержки.

Django помогает писать программное обеспечение, которое будет:

- полным. Django следует философии «Всё включено» и предоставляет почти всё, что разработчики могут захотеть сделать «из коробки». Поскольку всё, что нужно, является частью единого «продукта», всё это безупречно работает вместе, соответствует последовательным принципам проектирования и имеет обширную и актуальную документацию;

- разносторонним. Django может быть (и был) использован для создания практически любого типа веб-сайтов – от систем управления контентом и wiki до социальных сетей и новостных сайтов. Он может работать с любой клиентской средой и может доставлять контент практически в любом формате (включая HTML, RSS-каналы, JSON, XML и т. д.). Хотя Django предоставляет решения практически для любой функциональности, которая может понадобиться (например, для нескольких популярных баз данных, шаблонизаторов и так далее), внутренне он также может быть расширен сторонними компонентами, если это необходимо;

- безопасным. Django помогает разработчикам избежать многих распространённых ошибок безопасности, предоставляя фреймворк, разработанный чтобы «делать правильные вещи» для автоматической защиты сайта. Например, Django предоставляет безопасный способ управления учётными записями пользователей и паролями, избегая распространённых ошибок, таких как размещение информации о сессии в файлы cookie, где она уязвима или непосредственное хранение паролей вместо хэша пароля. Хэш пароля – это значение фиксированной длины, созданное путём обработки пароля через криптографическую хэш-функцию. Django может проверить правильность введённого пароля, пропустив его через хэш-функцию и сравнив вывод с сохранённым значением хэша. Благодаря «одностороннему» характеру функции, даже если сохранённое хэш-значение скомпрометировано, злоумышленнику будет сложно определить исходный пароль. Django, по умолчанию, обеспечивает защиту от многих уязвимостей, включая SQL-инъекцию, межсайтовый скриптинг, подделку межсайтовых запросов и клиджекинг (см. Website security для получения дополнительной информации об этих атаках);

– масштабируемым. Django использует компонентную «shared-nothing» архитектуру (каждая её часть независима от других и, следовательно, может быть заменена или изменена, если это необходимо). Чёткое разделение частей означает, что Django может масштабироваться при увеличении трафика, путём добавления оборудования на любом уровне: серверы кеширования, серверы баз данных или серверы приложений. Одни из самых загруженных сайтов успешно масштабировали Django (например, Instagram и Disqus, если назвать только два из них);

– удобным в сопровождении. Код Django написан с использованием принципов и шаблонов проектирования, которые поощряют создание поддерживаемого и повторно используемого кода. В частности, в нём используется принцип «Don't Repeat Yourself» (DRY, «не повторяйся»), поэтому нет ненужного дублирования, что сокращает объём кода. Django также способствует группированию связанных функциональных возможностей в повторно используемые «приложения» и, на более низком уровне, группирует связанный код в модули (в соответствии с шаблоном Model View Controller (MVC));

– переносным. Django написан на Python, который работает на многих платформах. Это означает, что вы не привязаны к какой-либо конкретной серверной платформе и можете запускать приложения на многих версиях Linux, Windows и Mac OS X. Кроме того, Django хорошо поддерживается многими веб-хостингами, которые часто предоставляют определённую инфраструктуру и документацию для размещения сайтов Django.

Bootstrap (также известен как Twitter Bootstrap) – свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML- и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения.

Bootstrap использует современные наработки в области CSS и HTML, поэтому необходимо быть внимательным при поддержке старых браузеров.

Преимущества Bootstrap при его использовании для frontend-разработки сайтов, веб-порталов и интерфейсов админок:

– высокая скорость создания качественной адаптивной вёрстки даже начинающими веб-разработчиками (достигается это благодаря использованию готовых классов и компонентов, созданных профессионалами);

– кроссбраузерность и кроссплатформенность (корректное отображение и работа сайта во всех поддерживаемых этим фреймворком браузерах и операционных системах);

– наличие большого количество готовых хорошо продуманных компонентов, протестированных огромным сообществом веб-разработчиков на различных устройствах;

– возможность настройки под свой проект, достигается это посредством изменения SCSS переменных и использования миксинов (можно изменить

количество колонок, цвета, радиус скруглений, отступы между колонками и т.д.);

- низкий порог вхождения; для работы с фреймворком не обязательно иметь «глубокие» знания по HTML, CSS, JavaScript (достаточно знать только основы этих технологий);

- однородность дизайна и его согласованность между различными компонентами (в Bootstrap все компоненты выполнены в едином стиле);

- наличие огромного количества сообществ и обучающих материалов; при желании это поможет не только хорошо разобраться в фреймворке, но и найти ответы практически на любые возникающие у вас вопросы.

Фреймворк Bootstrap – это проект с открытым исходным кодом, доступным на Github. Он имеет лицензию MIT. Это означает, что его можно бесплатно использовать как в открытых, так и в коммерческих проектах [8].

Класс обработки отправления сообщений на стороне backend представлен в листинге 3.1.

Листинг 3.1 – Обработка отправления сообщений

```
1. import json
2. from channels.generic.websocket import AsyncWebsocketConsumer
3. from channels.auth import login
4. from .models import Room, Message
5. from django.contrib.auth.models import User
6. class ChatConsumer(AsyncWebsocketConsumer):
7.     def fetch_messages(self, data):
8.         print('fetch')
9.         messages = Message.last_30_messages()
10.        content = {
11.            'messages': self.messages_to_json(messages)
12.        }
13.        self.send_message(content)
14.        def new_message(self, data):
15.            print('new_message')
16.            author = data['from']
17.            author_user = User.objects.filter(username=author)[0]
18.            message = Message.objects.create(
19.                author=author_user,
20.                content=data['message']
21.            )
22.            content = {
23.                'command': 'new_message',
24.                'message': self.message_to_json(message)
25.            }
26.            return self.send_chat_message(content)
27.        def messages_to_json(self, messages):
28.            result = []
29.            for message in messages:
30.                result.append(self.message_to_json(message))
31.            return result
```

```

32. def message_to_json(self, message):
33.     return {
34.         'author': message.author.username,
35.         'content': message.content,
36.         'created_at': str(message.created_at)
37.     }
38.     commands = {
39.         'fetch_messages': fetch_messages,
40.         'new_message': new_message,
41.     }
42.     async def connect(self):
43.         self.user = self.scope['user']
44.         print(self.scope.keys())
45.         self.room_name
self.scope['url_route']['kwargs']['room_name']
46.         self.room_group_name = 'chat_%s' % self.room_name
47.         await self.channel_layer.group_add(
48.             self.room_group_name,
49.             self.channel_name
50.         )
51.         await self.accept()
52.         async def disconnect(self, close_code):
53.             await self.channel_layer.group_discard(
54.                 self.room_group_name,
55.                 self.channel_name
56.             )
57.         async def receive(self, text_data):
58.             data = json.loads(text_data)
59.             self.commands[data['command']](self, data)
60.         async def send_chat_message(self, message):
61.             await self.channel_layer.group_send(
62.                 self.room_group_name,
63.                 {
64.                     'type': 'chat_message',
65.                     'message': message
66.                 })
67.         async def send_message(self, message):
68.             await self.send(text_data=json.dumps(message))
69.         async def chat_message(self, event):
70.             message = event['message']
71.             await self.send(text_data=json.dumps(message))

```

Пример формы регистрации, получаемой из модели при выполнении пользовательского запроса на стороне frontend представлен в листинге 3.2.

Листинг 3.2 – Форма регистрации пользователя

```

1. class UserCreationForm(forms.ModelForm):
2.     error_messages = {
3.         "password_mismatch": _("The two password fields didn't
match."),

```

```

4.     }
5.     password1 = forms.CharField(
6.         label=_("Password"),
7.         strip=False,
8.         widget=forms.PasswordInput(attrs={"autocomplete":          "new-
password"})),
9.     help_text=password_validation.password_validators_help_text_html(),
10. )
11. password2 = forms.CharField(
12.     label=_("Password confirmation"),
13.     widget=forms.PasswordInput(attrs={"autocomplete":          "new-
password"}),
14.     strip=False,
15.     help_text=_("Enter the same password as before, for
verification."),)
16. class Meta:
17.     model = User
18.     fields = ("username",)
19.     field_classes = {"username": UsernameField}
20.     def init(self, *args, **kwargs):
21.         super().init(*args, **kwargs)
22.         if self._meta.model.USERNAME_FIELD in self.fields:
23.             self.fields[self._meta.model.USERNAME_FIELD].widget.attrs[
24.                 "autofocus"
25.             ] = True
26.     def clean_password2(self):
27.         password1 = self.cleaned_data.get("password1")
28.         password2 = self.cleaned_data.get("password2")
29.         if password1 and password2 and password1 != password2:
30.             raise ValidationError(
31.                 self.error_messages["password_mismatch"],
32.                 code="password_mismatch",
33.             )
34.         return password2
35.     def _post_clean(self):
36.         super()._post_clean()
37.         password = self.cleaned_data.get("password2")
38.         if password:
39.             try:
40.                 password_validation.validate_password(password, self.instance)
41.             except ValidationError as error:
42.                 self.add_error("password2", error)
43.     def save(self, commit=True):
44.         user = super().save(commit=False)
45.         user.set_password(self.cleaned_data["password1"])
46.         if commit:
47.             user.save()
48.         return user

```


В ходе реализации программного продукта был разработан мессенджер «Fly» с шифрованием типа «end-to-end», отвечающий всем требованиям технического задания. Представление результатов реализации приведено в документе «Описание программы», показанном в приложении Г.

3.2 Разработка программной документации

В результате создания программного продукта была разработана следующая документация:

- техническая документация (приложение А);
- описание программы (приложение Г);
- программа и методика испытаний (приложение Д);
- руководство пользователя (приложение Е) – для помощи пользователю в ознакомлении и освоении программного продукта.

3.3 Тестирование

Тестирование программного обеспечения – это один из самых важных этапов в процессе создания продукта. Не уделив ему должного внимания, нельзя добиться высокого качества на выходе. В тестовых сценариях проверяются требования и характеристики конкретной функциональной возможности (функционала). Они могут предоставлять определённые детали или шаги, чтобы их можно было воспроизвести.

Тестирование программного обеспечения является неотъемлемой частью создания программного продукта. От того, насколько досконально проведены тесты, зависит то, как скоро проект будет сдан окончательно, и будет ли необходимость впоследствии устранять ошибки. Тестирование программного продукта на разных стадиях создания – залог качественного выполнения заказа.

Отсутствие должного тестирования программного продукта может вызвать волну отрицательных отзывов и впоследствии сформировать к программе негативное отношение [2].

Все виды тестирования программного обеспечения, в зависимости от преследуемых целей, можно условно разделить на следующие группы:

- функциональные;
- нефункциональные;
- связанные с изменениями.

Нефункциональные виды тестирования. Нефункциональное тестирование описывает тесты, необходимые для определения характеристик программного обеспечения, которые могут быть измерены различными величинами. В целом,

это тестирование того, «Как» система работает. Далее перечислены основные виды нефункциональных тестов:

- все виды тестирования производительности:
 - 1) нагрузочное тестирование (Performance and Load Testing);
 - 2) стрессовое тестирование (Stress Testing);
 - 3) тестирование стабильности или надежности (Stability / Reliability Testing);
 - 4) объемное тестирование (Volume Testing);
- тестирование установки (Installation testing);
- тестирование удобства пользования (Usability Testing);
- тестирование на отказ и восстановление (Failover and Recovery Testing);
- конфигурационное тестирование (Configuration Testing).

Связанные с изменениями виды тестирования. После проведения необходимых изменений, таких как исправление бага/дефекта, программное обеспечение должно быть пере тестировано для подтверждения того факта, что проблема была действительно решена. Ниже перечислены виды тестирования, которые необходимо проводить после установки программного обеспечения, для подтверждения работоспособности приложения или правильности осуществленного исправления дефекта:

- дымовое тестирование (Smoke Testing);
- регрессионное тестирование (Regression Testing);
- тестирование сборки (Build Verification Test);
- санитарное тестирование или проверка согласованности/исправности (Sanity Testing).

Функциональные тесты базируются на функциях и особенностях, а также взаимодействии с другими системами, и могут быть представлены на всех уровнях тестирования: компонентном или модульном (Component/Unit testing), интеграционном (Integration testing), системном (System testing) и приемочном (Acceptance testing). Функциональные виды тестирования рассматривают внешнее поведение системы [2].

В качестве метода тестирования было выбрано функциональное тестирование.

Функциональные тесты основываются на функциях, выполняемых системой, и могут проводиться на всех уровнях тестирования (компонентном, интеграционном, системном, приемочном). Как правило, эти функции описываются в требованиях, функциональных спецификациях или в виде случаев использования системы (use cases).

Результаты тестирования показаны в таблице Д.1 (приложение Д).

В результате тестирования система показала себя, как готовый к эксплуатации программный продукт, стабильным в использовании и с интуитивно понятным интерфейсом.

					ФАА.502300.074 ПЗ	Лист
						30
Изм.	Лист	№ докум.	Подпись	Дата		

4 ЭКОНОМИЧЕСКАЯ ЧАСТЬ

4.1 Обоснование необходимости вывода продукта на рынок

Цель экономического раздела – рассчитать затраты на разработку программного обеспечения и определить экономическую эффективность от его внедрения.

После проведенного в главе 1.2 сравнительного анализа мессенджеров-аналогов был сделан вывод о имеющемся ряде недостатков данных платформ. Данный анализ позволяет разрабатываемому программному продукту найти своих потребителей.

Реализованный продукт имеет широкое применение среди различных социальных групп, так как не имеет жестких ограничений в пользовании.

Мессенджер применим для всевозможных слоев населения, которые общаются посредством сети Интернет и которые боятся за свою конфиденциальность. Еще одним способом применения является привлечение клиентов для последующих заказов на различные услуги.

Таким образом, разрабатываемый продукт можно назвать актуальным для рынка мессенджеров, так как данное программное обеспечение не только не теряет актуальности, но и набирает обороты на сегодняшний день, а вместе с этим набирает обороты прослушивание и отслеживание пользователей не только государственными органами, но и мошенниками.

4.2 Структура (этапы) работ по созданию программного обеспечения

Жизненный цикл программного обеспечения (ПО) – период времени, который начинается с момента принятия решения о необходимости создания программного продукта и заканчивается в момент его полного изъятия из эксплуатации.

Для разработки программного продукта требуется время и структура этапов разработки, в которой будет выражаться количество недель, затраченных на его реализацию. Структура этапов и времени, выделенного для каждого этапа, необходимого для реализации «Мессенджера «Fly» с шифрованием типа «end-to-end»» представлена в таблице 4.1.

Таблица 4.1 – Таблица суммарного времени разработки программного обеспечения

Этапы разработки	Затраты времени, недели	Процент от общего времени, процент
Проектирование	1 неделя	8

Этапы разработки	Затраты времени, недели	Процент от общего времени, процент
Реализация	7 недель	58
Тестирование	2 недели	17
Доработка	2 недели	17
Итого	12 недель	100

Процент от общего времени находим по следующей формуле:

$$T_p = \frac{W_i}{W_s} \cdot 100\%, \quad (4.1)$$

где T_p – процент от общего времени;

W_i – число недель одного этапа;

W_s – количество недель разработки.

$$\frac{1}{12} \cdot 100\% = 8\%,$$

$$\frac{7}{12} \cdot 100\% = 58\%,$$

$$\frac{2}{12} \cdot 100\% = 17\%,$$

$$\frac{2}{12} \cdot 100\% = 17\%.$$

Диаграмма выполнения работ по разработке программного обеспечения (рисунок 4.1).

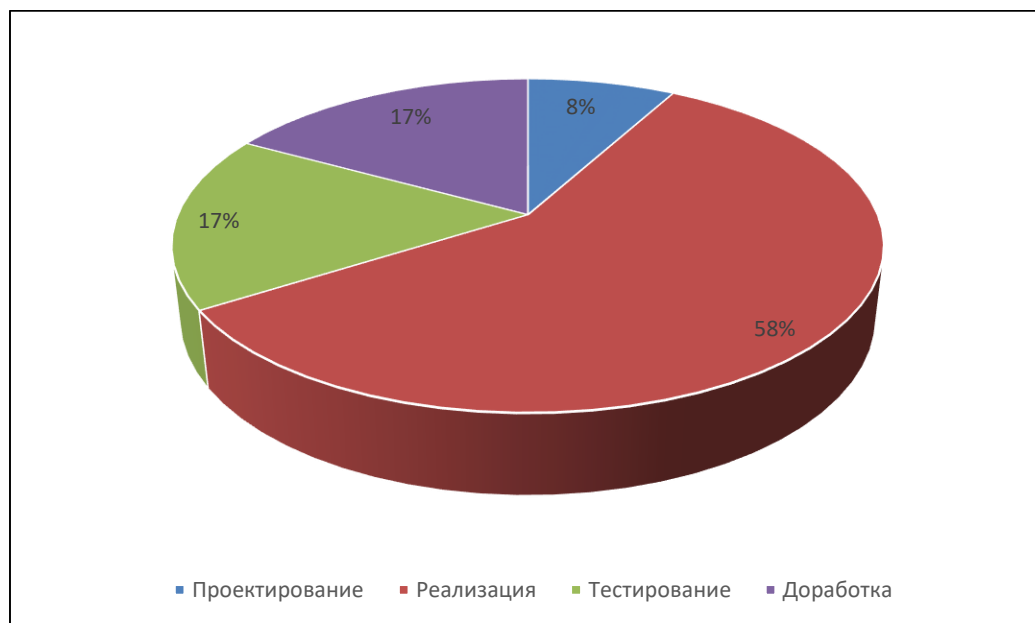


Рисунок 4.1 – Диаграмма распределения времени работы при разработке программного обеспечения

Кроме деления жизненного цикла программного обеспечения на стадии и фазы, можно выделить восемь видов работ, которые выполняются в процессе создания программного продукта:

- анализ требований и планирование – предусматривает разработку спецификаций, анализ и модификацию функциональных, технических, интерфейсных требований;

- проектирование изделий – включает в себя определение, спецификацию, анализ и модификации аппаратно-программной архитектуры проекта программы и базы данных;

- программирование – детальное проектирование, кодирование, автономная отладка и комплексирование отдельных компонентов программы, а также планирование работ программистов, разработка БД, документирование отдельных компонентов и организация программирования;

- планирование отладки;

- верификация – процесс проверки корректности требований, отладки изделий и приемные испытания;

- управление проектом – планирование и контроль проекта, контроль и регулирование договоров, связь с пользователями;

- контроль качества – разработка и контроль, стандартные и технические проверки программных средств и процессов разработки;

- документирование – разработка и корректировка руководства для пользователя и операторов.

Деление жизненного цикла программных средств на стадии, фазы и этапы, а также более подробное описание работ, выполняемых в процессе их разработки, производства и эксплуатации, необходимо для детального определения затрат, требуемых на эти цели.

4.3 Составление сметы затрат на разработку программного обеспечения

Расчет полной стоимости разработки и внедрения программного продукта учитывает следующие основные статьи затрат:

- материалы и комплектующие (М);
- электроэнергия (Э);
- основная заработная плата разработчиков (З_о);
- дополнительная заработная плата разработчиков (З_д);
- отчисления на социальные нужды (О_{сн});
- амортизация основных средств и нематериальных активов (А);
- расходы на спецоборудование (Р_с);
- прочие прямые расходы (П_з).

					ФАА.502300.074 ПЗ	Лист
						33
Изм.	Лист	№ докум.	Подпись	Дата		

1. Расходы по статье «Материалы и комплектующие» (М) отражают расходы на магнитные носители, бумагу и другие материалы, необходимые для разработки ПО.

Таблица 4.2 – Стоимость материалов, необходимых для разработки программного обеспечения

Наименование	Ед. измерения	Количество	Цена(руб./шт.)	Сумма(руб.)
Бумага формата А4	шт.	500	0,03	16,61
Компакт-диск	шт.	1	2	2
Картридж матричный ERC-30/34/38 (SC)	шт.	1	2,71	2,71
Всего				21,32

Общая стоимость основных и вспомогательных материалов М = 21,32 белорусских рубля.

2. Статья «Электроэнергия». Стоимость электроэнергии, необходимой для разработки программного обеспечения, представлена в таблице 4.4. Для расчета данного показателя необходимо иметь в наличии следующие данные:

- продолжительность недель разработки;
- кол-во часов потребления электроэнергии за весь период разработки;
- стоимость одного кВт/ч электроэнергии (тариф).
- значение курса белорусского рубля по отношению к доллару США,

установленного Национальным банком РБ, на дату установления тарифа на электрическую энергию.

Таблица 4.3 – Стоимость электроэнергии необходимой для разработки программного обеспечения

Статья расходов	Длительность периода, недели	Количество кВт/ч	Тариф за 1 кВт/ч.	Расход (бел. руб.)
Электроэнергия	12	480	0,3974	190,8

Затраты электроэнергии формируются из затрат электроэнергии за один рабочий день, количество дней в неделю, количество недель разработки. Максимальная потребляемая энергия компьютера составляет 1 кВт/ч. При 8-часовом рабочем дне и 5-дневной неделе получаем в неделю 40 часов, за 12 недель разработки 480 часов.

$$H_w = h \cdot w_d \cdot w_s, \quad (4.2)$$

где H_w – количество рабочего времени, затраченного на разработку;

h – продолжительность рабочего дня (часы);

W_d – количество рабочих дней;

W_s – сумма недель разработки.

$$8 \cdot 5 \cdot 12 = 480$$

Базовый тариф для прочих потребителей с 01.01.2022 г. при соотношении курса белорусского рубля к доллару США (1\$ = 2.5481 руб. РБ) составляет 0,40220 руб. за 1 кВт/ч (согласно Приказа Министерства антимонопольного регулирования и торговли Республики Беларусь от 18.01.2018 №11). При изменении курса доллара США тариф индексируется по формуле:

$$T_H = T_6 \cdot \left(0,31 + 0,69 \cdot \frac{K_H}{K_6} \right), \quad (4.3)$$

где T_H – тариф на электрическую энергию, подлежащий применению на день оформления платежных документов и день оплаты потребителем за потребленную электрическую энергию;

T_6 – тариф на электрическую энергию, установленный декларацией;

K_H – значение курса белорусского рубля по отношению к доллару США, установленного Национальным банком РБ, на день оформления платежных документов и день оплаты за потребленную электрическую энергию;

K_6 – значение курса белорусского рубля по отношению к доллару США, установленного Национальным банком РБ, на дату установления тарифа на электрическую энергию.

На 20.05.2022 курс белорусского рубля к доллару США 2,5043:1 установлен Национальным банком РБ. На основе данного курса рассчитаем тариф на электрическую энергию.

$$T_H = 0,4022 \cdot \left(0,31 + 0,69 \cdot \frac{2,5043}{2,5481} \right) = 0,3974$$

3. Статья «Основная заработная плата». Основной статьей расходов на создание ПО является заработная плата разработчиков (исполнителей) проекта, в число которых принято включать.

Таблица 4.4 – Расходы на специалистов

Специалисты	Количество (чел.)	Количество месяце в разработки	Основная зарплата (с учетом премии) за мес. (бел. руб.)	Основная с учетом дополнительной за месяц (бел. руб.)	Общая сумма заработной платы (бел. руб.)	Общая дополнительная (бел. руб.)
Инженеры программы	1	3	692,37	775,45	2077,11	2326,36
Итого:						2326,36

С 1 января 2020 года в Беларуси появится новая система начисления зарплаты работникам бюджетной сферы. Правовое основание для этого – Указ Президента от 18 января 2019 г. № 27 «Об оплате труда работников бюджетных организаций». Для оплаты труда бюджетников применяется базовая ставка. С 01.05.2022 базовая ставка составляет 210 рублей.

Рассчитаем зарплату месячную тарифную ставку (оклад). Для программистов, работающих в бюджетной организации представлены коэффициенты в таблице 4.5.

Таблица 4.5– Тарифные ставки

Категории работников	Разряд	Коэффициент
Специалисты, занимающие должности, квалификационными требованиями которых предусматривается наличие высшего, среднего специального образования	4 - 8	1.21 – 1.57

Месячная тарифная ставка каждого исполнителя (T_M) определяется путем умножения действующей базовой ставки (T_{M1}) на тарифный коэффициент (T_K), соответствующий установленному тарифному разряду:

$$T_M = T_{M1} \cdot T_K, \quad (4.4)$$

$$T_M = 210 \cdot 1,57 = 329,7.$$

Основная заработная плата исполнителей рассчитывается по формуле:

$$Z_o = T_M \cdot K \cdot n \cdot t, \quad (4.5)$$

где n – количество исполнителей, занятых разработкой конкретного ПО;

T_M – месячная тарифная ставка (бел. руб.);

K – коэффициент премирования;

t – период времени, затраченный на разработку ПО.

Рассчитаем основную заработную плату исполнителя:

$$Z_o = 329,7 \cdot 2,1 \cdot 1 \cdot 3 = 2077,11$$

Основная заработная плата исполнителя составила 2077,11 белорусских рубля.

4. Статья «Дополнительная заработная плата» на конкретное ПО (Z_d) включает выплаты, предусмотренные законодательством о труде (оплата отпусков, льготных часов, времени выполнения государственных обязанностей и других выплат, не связанных с основной деятельностью исполнителей), и определяется по нормативу в процентах к основной заработной плате:

$$Z_d = Z_{oi} \cdot H_d / 100\%, \quad (4.6)$$

где Z_{di} – дополнительная заработная плата исполнителей на конкретное ПО (руб.);

H_o – норматив дополнительной заработной платы (принимается в размере 7 - 15 %).

Дополнительная заработная плата на конкретное ПО составит:

$$Z_o = 2077,11 \cdot \frac{12}{100} = 249,25$$

5. Статья «Отчисления на социальные нужды». К данным затратам относят отчисления в фонд социальной защиты и отчисления на обязательное страхование от несчастных случаев ($H_{cmp} - 0,1 - 0,6\%$).

Отчисления на социальные нужды определяются в соответствии с действующими законодательными актами по нормативу в процентном отношении к фонду основной и дополнительной зарплаты исполнителей, определенной по нормативу, установленному в целом по организации:

$$Z_{csi} = (Z_{di} + Z_{oi}) \cdot (H_{cz} + H_{cmp}) / 100, \quad (4.7)$$

где H_{cz} – норматив отчислений в фонд социальной защиты населения (%);

H_{cmp} – норматив отчислений на обязательное страхование от несчастных случаев (%).

Тариф отчислений в фонд социальной защиты населения установлен в размере 34%: 28% – пенсионное страхование и 6% – социальное страхование.

Страховой тариф составляет: для страхователей бюджетных организаций - 0,1 процента; для иных страхователей – 0,6 процента.

Исходя из выше перечисленного, отчисления на социальные нужды составят:

$$Z_{csi} = (249,25 + 2077,11) \cdot (34 + 0.1) / 100 = 793,29$$

6. Статья «Амортизация основных средств и нематериальных активов». Определим стоимость основных средств (ноутбук) и нематериальных активов (программное обеспечение), на которые распространяются амортизационные отчисления (таблица 4.6).

Таблица 4.6 – Данные по амортизационной стоимости.

Активы	Стоимость актива, бел.р.	Период амортизации (лет)	За 1 год, бел. р.	За период разработки (3 мес.) бел.р.
Asus VivoBook 15 X513EA-BQ1916	2039	4	509,75	127,44
Программное обеспечение	968	8	121	30,25
Итого:	157,69			

Расчет амортизационных отчислений проведен на основе линейного способа. Линейный способ заключается в том, что в течение всего срока полезного использования основное средство или нематериальный актив списываются равными долями. Амортизацию начисляют ежемесячно, начиная со следующего месяца после введения имущества в эксплуатацию, и до тех пор, пока не амортизируют полностью первоначальную стоимость основного средства или нематериального актива.

Для того чтобы рассчитать сумму амортизации линейным способом, можно использовать один из двух вариантов:

– исходя из годовой или месячной нормы амортизации в процентах (предусматривает умножение амортизируемой стоимости объекта ОС на норму амортизации);

– без использования нормы амортизации (предусматривает распределение амортизируемой стоимости на срок амортизации ОС).

7. Статья «Расходы на спецоборудование» (P_{ci}) включает затраты средств на приобретение вспомогательных технических и программных средств специального назначения, необходимых для разработки конкретного программного обеспечения, включая расходы на их проектирование, изготовление, отладку, установку и эксплуатацию. Затраты по этой статье определяются в соответствии со сметой расходов, которая составляется перед разработкой ПО.

Данная статья включается в смету расходов на разработку программного обеспечения в том случае, когда приобретаются специальное оборудование или специальные программы, предназначенные для разработки и создания только данного ПО.

Сумма начисленной амортизации за 1-й год составит:

$$P_{ci} = \sum_{i=1}^n C_{ci}, \quad (4.8)$$

где C_{ci} – годовая норма амортизации;

n – остаточная или первоначальная стоимость актива.

Расходы на спецоборудование не понадобились.

8. Статья «Прочие прямые расходы» (Π_z) на конкретное ПО включает затраты: на оплату услуг связи, Интернета, транспортные расходы, канцтовары, приобретение и подготовку специальной научно-технической информации и специальной литературы. Расходы определяются студентом самостоятельно по действующим на момент написания работы ценам и тарифам.

Тарифный план 4G не лимитируемого доступа в сеть Интернет («Безлимитище+») (скорость прием/передача до 112 Мбит/с)), в месяц 26,96 белорусских рублей.

$$3 \text{ месяца} \cdot 26,96 = 80,84$$

					ФАА.502300.074 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		38

Общая сумма расходов по смете (плановая себестоимость) (C) на ПО рассчитывается по формуле:

$$C = M + \mathcal{E} + 3_o + 3_d + O_{\text{сн}} + A + P_c + \Pi_3 \quad (4.9)$$

Результаты расчетов представлены в таблице 4.7.

Таблица 4.7 – Расчет плановой себестоимости программного обеспечения

Статья затрат	Затраты, бел. руб.
1. Материалы и комплектующие (M);	21,32
2. Электроэнергия;	190,8
3. Основная заработная плата исполнителей (3_o)	2077,11
4. Дополнительная заработная плата исполнителей (3_d);	249,25
5. Отчисления на социальные нужды ($O_{\text{сн}}$);	793,29
6. Амортизация (A);	157,69
7. Расходы на спецоборудование (P_c);	-
8. Прочие прямые расходы (Π_3);	80,88
Общая сумма расходов по смете (C)	3570,34

Представим затраты из таблицы 4.7 в графическом виде (рисунок 4.2).

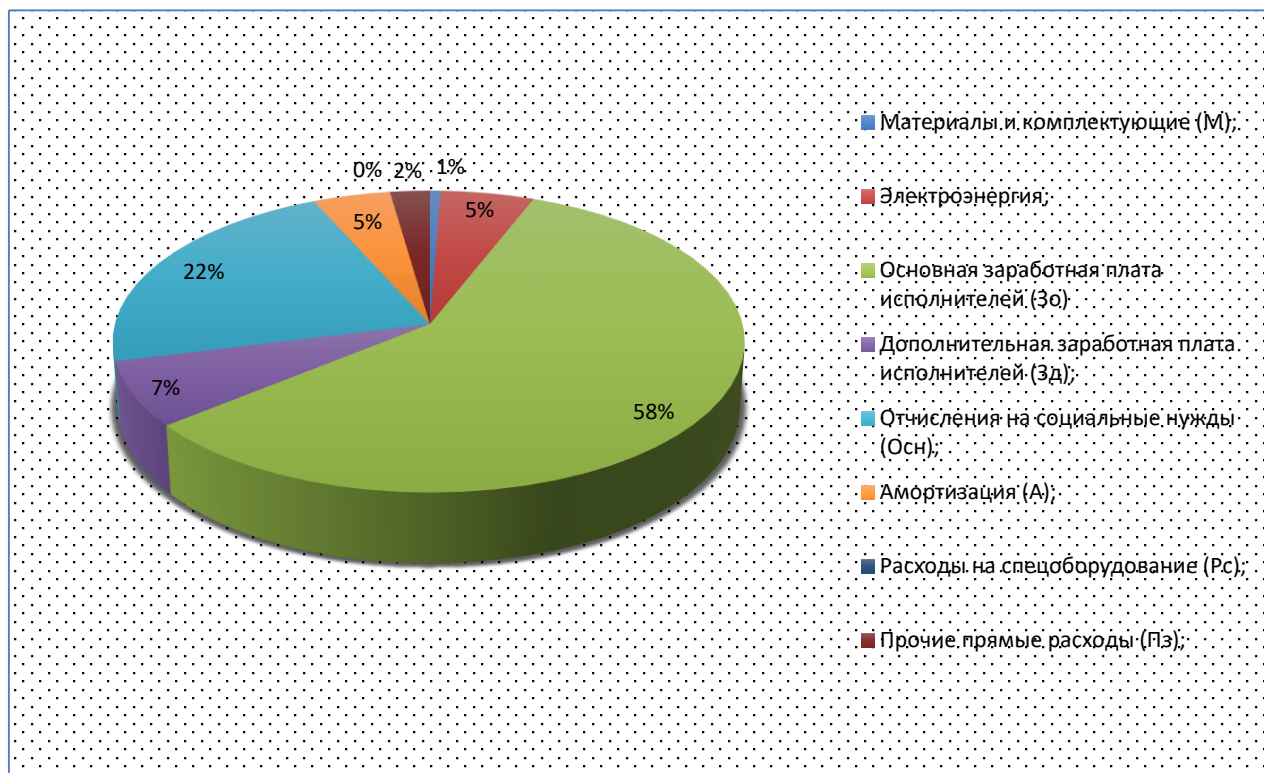


Рисунок 4.2 – Структура затрат на разработку программного обеспечения

4.4 Расчет экономического эффекта разработчика программного обеспечения

4.4.1 Экономический эффект у разработчика программного обеспечения

Рассчитаем отпускную цену программного обеспечения.

С учетом действующих в республике нормативных документов отпускная цена на продукцию предприятия рассчитывается по формуле:

$$ОЦ = С + П, \quad (4.10)$$

где $ОЦ$ – отпускная цена изготовителя, руб.;

$С$ – плановая себестоимость, руб.;

$П$ – прибыль руб.

$$3570,34 + 714,07 = 4284,41$$

Прибыль закладывается в цену по нормативу рентабельности (устанавливается самостоятельно), расчет производится по следующей формуле:

$$П = R \cdot С, \quad (4.11)$$

где R – норматив рентабельности (например, если рентабельность 20 %, то при расчете переводим в коэффициент и получаем 0,2).

$С$ – плановая себестоимость, руб.;

$$3570,34 \cdot 0,2 = 714,07$$

Отпускная цена изготовителя с налогом на добавленную стоимость:

Ставка налога на добавленную стоимость установлена в размере 20 %.

$$НДС = (С + П) \cdot \text{ставка НДС}(\%) / 100 \%, \quad (4.12)$$

где $С$ – плановая себестоимость, руб.;

$П$ – прибыль, руб.; ставка НДС = 20 %;

$$(3570,34 + 714,07) \cdot 20 / 100 = 856,88$$

Прогнозируемая отпускная цена на программное обеспечение с НДС:

$$ОЦ \text{ с НДС} = С + П + НДС, \quad (4.13)$$

$$4810,75 + 962,15 + 1154,58 = 6927,48$$

4.4.2 Экономический эффект от использования программного обеспечения у пользователя (заказчика)

Под риском информационной безопасности понимается возможность того, что данная угроза будет использовать уязвимости информационного актива (группы активов) и тем самым нанесет вред организации. Он измеряется комбинацией вероятности нежелательного события и его последствий (возможного ущерба). В последние годы проблема рисков информационной безопасности стала как нельзя актуальной, причиной тому послужил серьезный ущерб, влияющий не только на финансовую составляющую.

В связи с этим во время разработки мессенджера были применены одни из самых популярных средств по защите данных. У многих конкурентов имеются большие проблемы с информационной безопасностью, поэтому разработанный сервис будет привлекать внимание большего количества пользователей.

Основываясь на современных подходах к защите данных, экономический эффект у пользователя программного обеспечения можно выразить в виде дополнительной монетизации за счет наличия внутренней подписки. Стоимость минимальной подписки составляет 10 рублей в месяц и 20 рублей в месяц за максимальную. Затраты на покупку специализированного приложения, стоимость которого равна 5141,29 рублей без учета маркетинговых затрат, сравниваются с затратами с затратами ежегодной покупки через 1 год использования, при покупке хотя бы 258-ми пользователями максимальной подписки.

4.4.3 Расчет экономической эффективности разработки ПО для заказчика.

При сравнении базового и нового варианта программного обеспечения в качестве экономического эффекта будет выступать общая сумма экономии всех видов ресурсов относительно базового варианта.

Приведем расчет экономии некоторых видов ресурсов в связи с использованием нового программного обеспечения.

Экономия затрат на заработную плату (\mathcal{E}_3) при использовании нового программного обеспечения в расчете на объем выполненных работ определяется по формуле:

$$\mathcal{E}_3 = \mathcal{E}_{39} \cdot Q, \quad (4.14)$$

где \mathcal{E}_{39} – экономия затрат на заработную плату при решении задач с использованием нового программного обеспечения в расчете на 1 задачу, руб.;

					ФАА.502300.074 ПЗ	Лист
						41
Изм.	Лист	№ докум.	Подпись	Дата		

Q – количество задач, решаемых за год с использованием нового программного обеспечения.

Экономия затрат на заработную плату в расчете на 1 задачу (\mathcal{E}_z) рассчитывается по формуле:

$$\mathcal{E}_z = (Z_{cm} \cdot (T_{c1} - T_{c2}) / T_{\text{ч}}) / D_p \quad (4.15)$$

где Z_{cm} – среднемесячная заработная плата одного программиста, руб.;
 T_{c1}, T_{c2} – снижение трудоемкости на одну задачу в базовом и новом варианте соответственно, чел.-ч.;

$T_{\text{ч}}$ – количество часов работы в день, ч.;

D_p – среднемесячное количество рабочих дней.

Экономия с учетом начислений на заработную плату (\mathcal{E}_n):

$$\mathcal{E}_n = \mathcal{E}_z \cdot K_{nz}, \quad (4.16)$$

где K_{nz} – коэффициент начислений на заработную плату. Рассчитывается по данным заказчика.

Экономия за счет сокращения простоя сервиса (\mathcal{E}_c):

$$\mathcal{E}_c = \frac{D_{pz} \cdot (P_1 - P_2) \cdot C_{\text{п}}}{60}, \quad (4.17)$$

где P_1, P_2 – время простоя сервиса, обусловленное программным обеспечением в день;

D_{pz} – плановый фонд работы сервиса (дней);

C_n – стоимость одного часа, руб.

Общая экономия от применения нового программного обеспечения рассчитывается по формуле:

$$\mathcal{E}_o = \mathcal{E}_n \cdot \mathcal{E}_c, \quad (4.18)$$

где K_{nz} – коэффициент начислений на заработную плату. Рассчитывается по данным заказчика.

Плановый объем выполняемых работ в год составляет 800 задач. Среднемесячная заработная плата программиста в бюджетной организации 775,45 руб. Средняя трудоемкость работ в расчете на 1 задачу 0,06 и 0,017 человеко-часов в базовом и новом варианте программного обеспечения соответственно. Количество часов работы в день 8 часов. Среднемесячное количество рабочих дней 22 дня.

Экономия затрат на заработную плату в расчете на 1 задачу составит:

$$\mathcal{E}_{39} = \frac{775,45 \cdot (0,06 - 0,017) / 8}{22} = 0,19 \text{ (бел.руб.)}$$

Экономия затрат на заработную плату на весь объем запланированных задач составит:

$$\mathcal{E}_3 = 0,19 \cdot 800 = 152 \text{ (бел.руб.)}$$

Экономия с учетом начислений на заработную плату:

$$\mathcal{E}_n = 152 \cdot 1,2 = 182,4 \text{ (бел.руб.)}$$

Для того, чтобы мессенджер функционировал в общем доступе, необходимо оплачивать хостинг, ObjectStorage и домен. Стоимость домена в год составляет 33 бел. рубля, хостинг стоит 219 рублей за год, а ObjectStorage – 15,4 рублей. В общем цена составляет 267,4 рублей.

$$\mathcal{E}_c = \frac{225 \cdot (20 - 10) \cdot 0,6}{60} = 22,5 \text{ (бел.руб.)}$$

Сумма общей экономии:

$$\mathcal{E}_o = 182,4 + 22,5 = 204,9 \text{ (бел.руб.)}$$

Исходя из выше подсчитанных цифр, общая экономия от применения разработанного программного обеспечения составит 204,9 рублей в год.

4.5 Выводы по экономической части

В данной экономической части проекта представлено обоснование о необходимости вывода программного продукта на рынок с приведением следующих расчетов по разработке ПО: затраты времени на этапы разработки, затраты на материалы, комплектующие, затраты на электроэнергию, заработную плату персоналу, амортизацию материальных и нематериальных активов, приобретение оборудования для разработки, расчет прибыли, отчисления на социальные нужды и налоговые отчисления. Представлен расчет экономической эффективности разработки ПО для (пользователя) заказчика.

Капитальные вложения заказчика на приобретение и внедрение ПО составляют 5141,29 бел. руб. Экономия денежных средств составила для заказчика 204,9 бел. руб. Исходя из показателей можно сделать вывод, что приобретение и внедрение ПО заказчиком экономически целесообразно.

					ФАА.502300.074 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		43

ЗАКЛЮЧЕНИЕ

В ходе реализации дипломного проекта итогом является спроектированный мессенджер «Fly» с шифрованием типа «end-to-end».

В последствии реализации дипломного проекта были решены следующие задачи:

- изучение предметной области;
- проведение анализа аналогов и прототипов с описанием преимуществ и недостатков рассматриваемых систем;
- выявление требований к разрабатываемому программному продукту;
- разработка технического задания;
- проектирование программного продукта;
- реализация программного продукта;
- тестирование готового программного продукта;
- разработка сопроводительной документации («Техническое задание», «Описание программы», «Программа и методика испытаний», «Руководство пользователя»);
- проведение расчетов цены и себестоимости программного продукта.

При разработке мессенджера использовались следующие технологии:

1. Серверная часть:

- СУБД PostgreSQL;
- серверный язык программирования – Python;
- Django.

2. Клиентская часть:

- HTML – язык гипертекстовой разметки;
- CSS – язык таблиц стилей;
- свободный набор инструментов Bootstrap;
- JavaScript – язык программирования, который добавляет интерактивность на ваш веб-сайт.

В качестве сред разработки использовались:

- интегрированная среда разработки – PyCharm 2021 3.3;
- редактор исходного кода – Visual Studio Code.

Реализованное программное средство полностью соответствует требованиям технического задания.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Визуальное моделирование [Электронный ресурс]: Википедия. – Режим доступа: https://en.wikipedia.org/wiki/Visual_modeling – Дата доступа: 25.05.2022;
- 2 Куликов С. Тестирование программного обеспечения. Базовый курс;
- 3 PostgreSQL: The World's Most Advanced Open Source Relational Database [Электронный ресурс] – Режим доступа: <https://www.postgresql.org> – Дата доступа: 21.05.2022.;
- 4 PyCharm: IDE для профессиональной разработки на Python [Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/ru-ru/pycharm/> – Дата доступа: 21.05.2022;
- 5 The web framework for perfectionists with deadlines | Django [Электронный ресурс] – Режим доступа: <https://www.djangoproject.com> – Дата доступа: 21.05.2022;
- 6 The official home of the Python Programming Language [Электронный ресурс] – Режим доступа: <https://www.python.org> – Дата доступа: 21.05.2022;
- 7 Реляционная база данных [Электронный ресурс]: Википедия. – Режим доступа: https://ru.wikipedia.org/wiki/Реляционная_база_данных – Дата доступа: 01.04.2022.SEO Wiki / Web-портал [Электронный ресурс] – Режим доступа: <https://wiki.rookee.ru/web-portal/> – Дата доступа: 23.05.2022.
- 8 Bootstrap. Документация на русском [Электронный ресурс] – Режим доступа: <https://bootstrap-4.ru/> – Дата доступа: 22.05.2022.
- 9 Диаграмма вариантов использования [Электронный ресурс] / Портал зн. – Режим доступа: http://www.znannya.org/?view=uml_use_case_diagram_4. – Дата доступа: 15.05.2022.
- 10 Современное шифрование изнутри [Электронный ресурс]. – Режим доступа: <https://trashbox.ru/link/what-is-encryption> – Дата доступа: 17.05.2022.
- 11 Что такое мессенджеры, какие у них возможности [Электронный ресурс]. – Режим доступа: <https://www.inetgramotnost.ru/online-servisy/chto-takoe-messendzhery-ih-vozmozhnosti-10-messendzherov.html> – Дата доступа: 17.05.2022.
- 12 Процесс разработки программного обеспечения [Электронный ресурс]: Википедия. – Режим доступа: <https://clck.ru/DapYo> – Дата доступа: 10.05.2022.

ПРИЛОЖЕНИЕ А
(обязательное)
Техническое задание

Введение

Наименование продукта: мессенджер «Fly» с шифрованием типа «end-to-end».

Данный продукт будет использоваться жителями Республики Беларусь для общения, а также для защиты своих данных от мошенников. Мессенджер «Fly» хранит в себе информацию о зарегистрированных пользователях, а также шифрует данные пользователей, обеспечивая конфиденциальность.

А.1 Основания для разработки

Основанием для разработки является приказ №___ от 05.2022. «О закреплении тем дипломных проектов студентов факультета информационных технологий специальности 1-40-01-01 «Программное обеспечение информационных технологий» по кафедре технологий программирования УО «Полоцкий государственный университет»

Тема разработки – мессенджер «Fly» с шифрованием типа «end-to-end», выданная студенту группы 18ИТ-2 Филиппову Александру Андреевичу, руководителем назначен старший преподаватель кафедры технологий программирования Рогулев Владимир Сергеевич.

А.2 Назначение разработки

Основной целью разработки мессенджера является обеспечение конфиденциальности для всех его пользователей. Мессенджер предоставляет следующие возможности:

- регистрация пользователя в мессенджере с указанием необходимой информации в аккаунте;
- редактирование и удаление аккаунта из мессенджера;
- индивидуальная настройка мессенджера;
- общение с другими пользователями;
- добавление файлов к сообщению;
- просмотр статуса собеседника;

					ФАА.502300.074 ПЗ	Лист
						46
Изм.	Лист	№ докум.	Подпись	Дата		

— защита всех данных пользователей посредством сквозного шифрования.

А.3 Требование к программе или программному изделию

А.3.1 Требования к функциональным характеристикам

К функциональным характеристикам выдвигаются следующие требования:

— пользователь может просматривать аккаунты других пользователей (с разрешения этого пользователя), добавлять фотографии формата JPG/JPEG, добавлять информацию для дополнения профиля, редактировать профиль и удалять лишнюю информацию в профиле;

— при некорректном вводе данных пользователем программа должна отменить добавление или редактирование и отобразить ошибки, не сбрасывая введенные данные;

— в аккаунте пользователя должны отображаться его контактные данные: ФИ, номер телефона и почта (могут быть скрыты с помощью функции приватности);

— при просмотре диалога (с одним и более людьми) должна быть доступна возможность просмотра всех ранее прикрепленных другими пользователями файлов;

— при отсутствии данных в таблицах они не должны отображаться на странице.

А.3.2 Требования к надёжности

Программное средство должно обеспечивать контроль корректности входных данных. В случае обнаружения ошибок во входных данных пользователю должны выводиться соответствующие сообщения без сброса введенных данных.

А.3.3 Условия эксплуатации

Данное программное средство не требует предварительной подготовки.

					ФАА.502300.074 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		47

А.3.4 Требования к составу и параметрам технических средств

Минимальными и достаточными требованиями по конфигурации оборудования клиента для функционирования системы являются:

- процессоры с частотой не менее 233 МГц;
- наличие браузера;
- объем оперативной памяти не менее 128 Мб;
- свободное дисковое пространство не менее 50 Мб.

А.3.5 Требования к информационной и программной совместимости

Для программного обеспечения клиента требуется:

- Операционная система: Windows 7, 8, 10.
- Любой браузер.

Для программного обеспечения сервера требуется:

- PostgreSQL 14.

А.3.6 Требования к маркировке и упаковке

Требования к маркировке и упаковке отсутствуют.

А.3.7 Требования к транспортированию и хранению

Требования к транспортировке и хранению отсутствуют.

А.4 Требования к программной документации

Программная документация должна состоять из следующих документов:

- техническое задание в соответствии с ГОСТ 19.201-78 ЕСПД;
- описание программы в соответствии с ГОСТ 19.402-2000 ЕСПД;
- руководство оператора в соответствии с ГОСТ 19.505-79 ЕСПД;
- программа и методика испытаний в соответствии с ГОСТ 19.301-2000

ЕСПД.

					ФАА.502300.074 ПЗ	Лист
						48
Изм.	Лист	№ докум.	Подпись	Дата		

Содержание и структура программной документации соответствует требованиям ЕСПД.

А.5 Технико-экономические показатели

Эффективность данного программного продукта в основном обуславливается отсутствием полных аналогов данного приложения.

А.6 Стадии и этапы разработки

Этапы разработки:

- постановка задачи;
- анализ исходных данных;
- разработка программного средства;
- тестирование программного средства;
- разработка технической документации к данному программному средству.

А.7 Порядок контроля и приёмки

Для контроля выполнения требования технического задания необходимо провести испытания. Порядок и состав испытания определяются программой и методикой испытания.

Приемка готового программного средства осуществляется комиссией по защите дипломного проекта в соответствии с программой и методикой испытаний.

					ФАА.502300.074 ПЗ	Лист
						49
Изм.	Лист	№ докум.	Подпись	Дата		

ПРИЛОЖЕНИЕ Б
(справочное)
Диаграмма вариантов использования

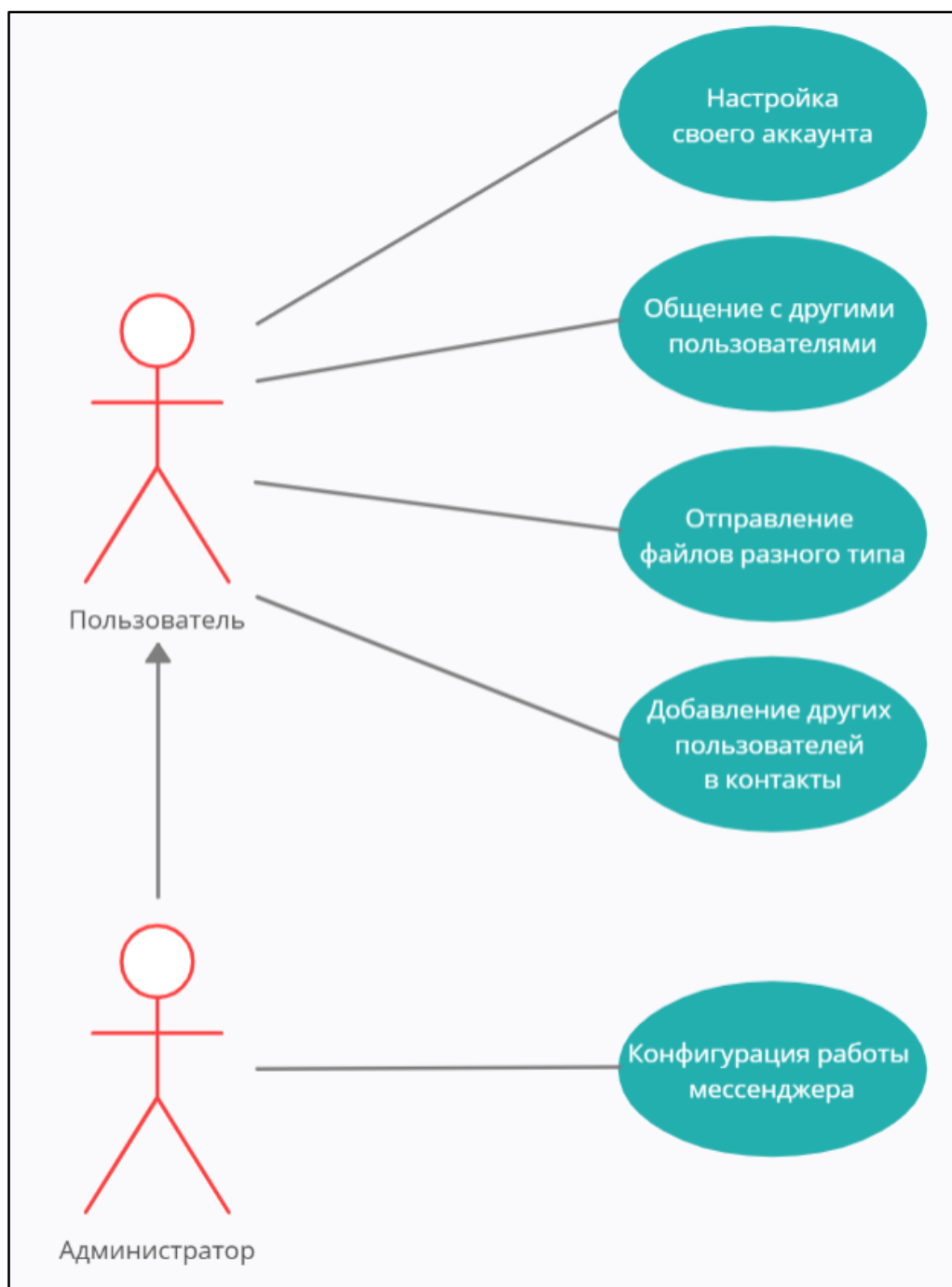


Рисунок Б.1 – Диаграмма вариантов использования

ПРИЛОЖЕНИЕ В (справочное) База данных

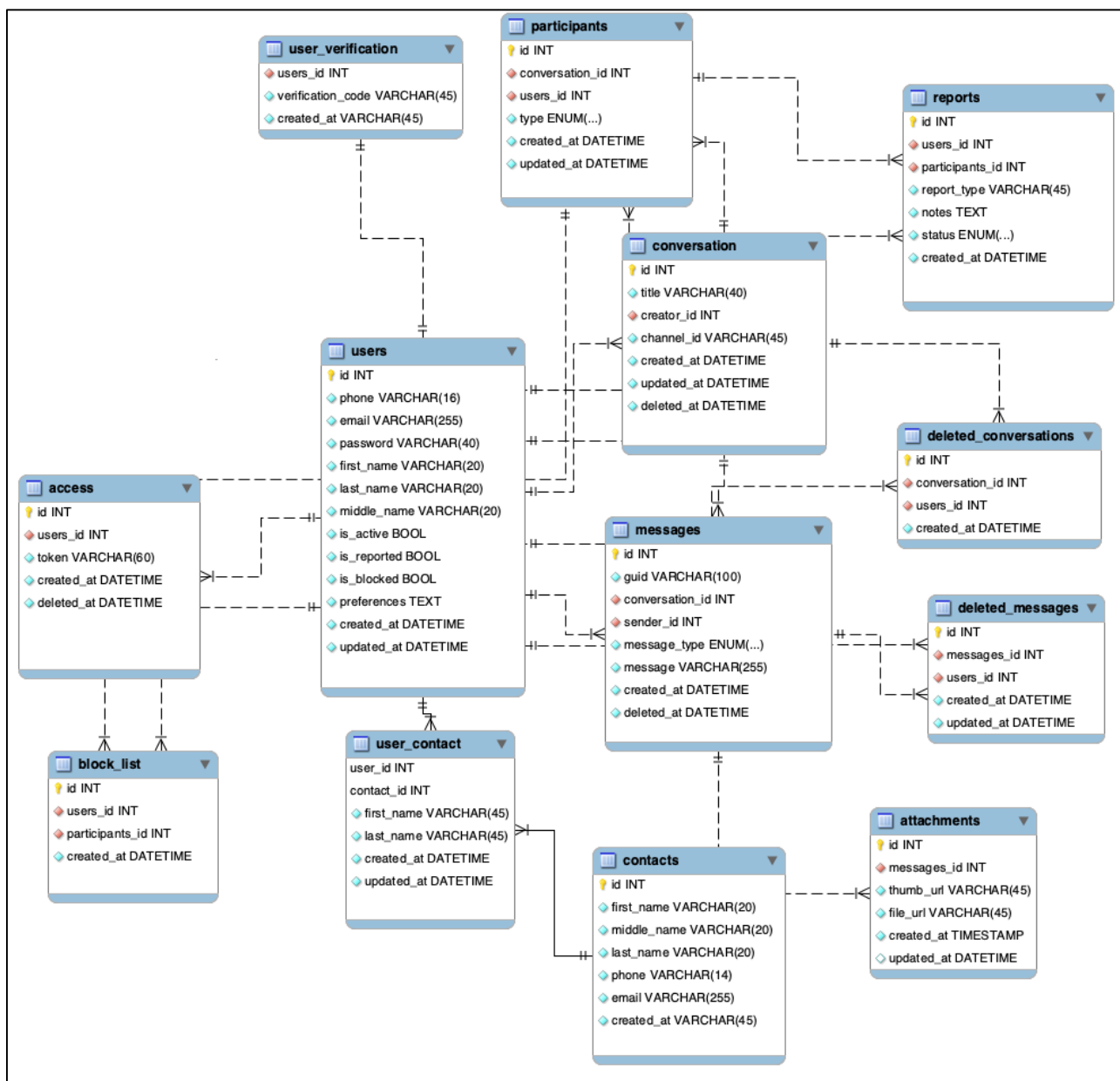


Рисунок В.1 – База данных

ПРИЛОЖЕНИЕ Г

(обязательное)

Описание программы

Г.1 Общие сведения

Наименование программного средства – мессенджер «Fly» с шифрованием типа «end-to-end».

Данный программный продукт предоставляет возможность пользователям общаться между собой в сети Интернет, обеспечивая защиту от слежки и кражи данных злоумышленниками.

Для успешного функционирования программного продукта необходимо:

- подключение к сети Интернет;
- установленный один из популярных веб-браузеров последней версии.

Интерфейс разрабатываемого мессенджера реализован при помощи языка гипертекстовой разметки HTML5, языка таблиц стилей CSS, фреймворка Bootstrap и набора функций JavaScript.

Г.2 Функциональное назначение

Данное программное средство предназначено для решения следующих задач:

- предоставление возможности пользователям переписываться друг с другом, обеспечивая защиту от слежки и кражи данных;
- предоставление возможности передачи файлов между пользователями;
- предоставление возможности создания собственного канала с целью продвижения своих товаров или услуг.

Г.3 Описание логической структуры

Данный программный продукт работает по следующему алгоритму:

- для запуска портала требуется открыть веб-браузер;
- в адресной строке веб-браузера нужно ввести адрес мессенджера, после чего откроется главная страница мессенджера «Fly»;
- для входа в профиль нужно нажать на кнопку «Авторизация», после чего откроется форма авторизации, ввести личные данные и войти в систему.

					ФАА.502300.074 ПЗ	Лист
						52
Изм.	Лист	№ докум.	Подпись	Дата		

Г.4 Используемые технические средства

Пользовательский компьютер, на котором будет осуществляться взаимодействие с веб-приложением должен соответствовать минимальным требованиям, которые перечислены ниже:

- процессор с тактовой частотой 1350 МГц и выше;
- оперативная память 2 Гб и выше;
- монитор, клавиатура, мышь;
- подключение к сети Интернет (со скоростью соединения 1 МБит/с);
- установленный один из популярных веб-браузеров последней версии;
- наличие свободного места на жестком диске не менее 50 МБ.

Сервер обработки запросов должен соответствовать следующей конфигурации:

- процессор с тактовой частотой не менее 1350 МГц;
- не менее 2 Гб оперативной памяти;
- наличие свободного места на жестком диске не менее 1 Гб.

Г.5 Вызов и загрузка

Загрузка ПО осуществляется при запуске веб-сервера. Вызов страницы мессенджера осуществляется при открытии страницы в браузере.

Г.6 Входные данные

Для разрабатываемого программного продукта входной будет служить следующая информация:

- информация о пользователях;
- информация о местоположении.

Г.7 Выходные данные

Выходными данными являются веб-страницы, которые сгенерированы сервером и отправлены пользователю.

					ФАА.502300.074 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		53

ПРИЛОЖЕНИЕ Д
(обязательное)
Программа и методика испытаний

Д.1 Объект испытаний

Объект испытаний: «Мессенджер «Fly» с шифрованием типа «end-to-end»».

Д.2 Цель испытаний

Проверка программного продукта «Мессенджер «Fly» с шифрованием типа «end-to-end»» с целью установления его работоспособности, соответствия техническому заданию и выявления несоответствий в работе программы и в программной документации.

Д.3 Требования к программе

Разработанный программный продукт должен включать в себя функциональный модуль пользователя, который отвечает за:

- настройку профиля пользователя;
- поиск собеседников в самом мессенджере;
- общение с другими пользователями;
- пересылку файлов;
- создание общих бесед или же личного канала;
- сквозное шифрование данных пользователя.

Д.4 Требования к программной документации

Проект должен содержать следующие программные документы:

- описание программы;
- техническое задание;
- руководство пользователя.

					ФАА.502300.074 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		54

Д.5 Средства и порядок испытания

Д.5.1 Средства для испытания

Для проведения испытаний над веб-сайтом необходимо наличие персонального компьютера, с возможностью выхода в сеть Интернет. Для проверки корректности функционирования разработанного программного обеспечения должны применяться следующие программные средства:

- процессор с тактовой частотой 1350 МГц и выше;
- оперативная память 2 Гб и выше;
- монитор;
- клавиатура;
- мышь;
- подключение к сети Интернет (со скоростью соединения 1 МБит/с);
- установленный один из популярных веб-браузеров последней версии (Google Chrome, Mozilla Firefox, Opera и другие);
- наличие свободного места на жестком диске не менее 50 МБ.

Сервер обработки запросов должен соответствовать следующей конфигурации:

- процессор с тактовой частотой не менее 1350 МГц;
- не менее 2 Гб оперативной памяти;
- наличие свободного места на жестком диске не менее 1 Гб.

Д.5.1 Порядок испытаний

Проверка выполняется в соответствии с тест-планом, представленном в таблице Д.1.

Таблица Д.1 – Способы проверок с указанием ожидаемых результатов испытаний

Тестовый вариант	Входные данные	Ожидаемый результат	Результат тестирования
Запуск клиентского модуля	Вход в мессенджер	Открытие главной страницы	Тест пройден успешно
Переход на форму «Регистрация»	Нажатие на пункт главной страницы «Регистрация»	Открытие формы «Регистрация»	Тест пройден успешно

Продолжение таблицы Д.1

Тестовый вариант	Входные данные	Ожидаемый результат	Результат тестирования
Переход на форму «Авторизация»	Нажатие на пункт главной страницы «Авторизация»	Открытие формы «Авторизация»	Тест пройден успешно
Получение письма на почту с целью подтверждения регистрации	Нахождение на странице с почтой, нажатие на ссылку в полученном письме уведомлении	Открытие формы «Регистрация прошла успешно»	Тест пройден успешно
Авторизация в мессенджере	Заполненные поля для авторизации пользователя	Открытие непосредственно й страницы пользователя	Тест пройден успешно
Поиск пользователей	Имя требуемого пользователя в поле поиска	Найдены совпадения, соответствующие запросу пользователя	Тест пройден успешно
Отправка сообщений	Нажатие на кнопку «Отправить сообщение» с каким-либо текстом	Сообщение, которое написал пользователь, доставлено собеседнику	Тест пройден успешно
Переход на форму диалога с другим пользователем	Нажатие на кнопку диалога с пользователем	Открытие формы с отправленными и полученными сообщениями	Тест пройден успешно
Создание беседы	Нахождение в форме «Создать беседу» с добавленными двумя и более собеседниками и нажатие на кнопку «Создать»	Создание нового чата с собеседниками, добавленными в форму	Тест пройден успешно
Переход на страницу «Профиль»	Нахождение на главной странице и нажатие на кнопку «Профиль»	Открывается страница с информацией о профиле пользователя	Тест пройден успешно

Продолжение таблицы Д.1

Тестовый вариант	Входные данные	Ожидаемый результат	Результат тестирования
Переход на страницу «Настройки»	Нахождение на странице «Профиль» и нажатие на кнопку «Настройки»	Открывается страница с настройками профиля	Тест пройден успешно
Создание контакта	Нажатие ПКМ на профиле пользователя, а затем нажатие кнопки «Добавить в контакты»	Появление сообщения «Запрос на добавление в контакты отправлен»	Тест пройден успешно
Отправка жалобы на пользователя	В форме «Пожаловаться» заполнены все поля, а после нажатие кнопки «Отправить»	Появление сообщения «Жалоба отправлена»	Тест пройден успешно

Д.6 Методы испытаний

Проверка данного программного средства должна осуществляться методом «Черного ящика», который предназначен для проверки функционального назначения программы.

Корректность выполнения определяется сравнением фактического и ожидаемого результатов. Если результаты совпали, то тестирование пройдено успешно.

Если же результаты не равны или произошел сбой системы, и пользователю вывелось системное сообщение об ошибке, то это свидетельствует о том, что тестирование пройдено неуспешно.

ПРИЛОЖЕНИЕ Е
(обязательное)
Руководство пользователя

Е.1 Назначение программы

«Мессенджер «Fly» с шифрованием типа «end-to-end»» предоставляет пользователям возможность общаться с другими пользователями, обмениваться файлами в сети Интернет, заниматься продвижением своих товаров и услуг, обеспечивая при этом конфиденциальность и полную приватность данных.

Благодаря мессенджеру «Fly» не стоит задумываться о том, что кто-то извне сможет прочитать ваши сообщения и украсть личные данные, так как в нём используется самый современный алгоритм шифрования – сквозной (или же «end-to-end»).

Е.2 Условия выполнения программы

Для того чтобы работа с программным средством на компьютере пользователя была удобной, компьютер должен иметь минимальную аппаратную комплектацию: монитор, системный блок, клавиатура, мышь, доступ к сети Интернет.

Минимальная аппаратная конфигурация системы пользователя, при которой обеспечивается стабильная работа программного средства, должна быть не ниже следующей:

- процессор с тактовой частотой 1350 МГц и выше;
- оперативная память 2 Гб и выше;
- соединение с сетью интернет (со скоростью соединения 1 МБит/с).

Сервер обработки запросов должен соответствовать следующей конфигурации:

- процессор с тактовой частотой не менее 1350 МГц;
- не менее 2 Гб оперативной памяти; – наличие свободного места на жестком диске не менее 1 Гб.

Если данные требования будут соблюдены, то программное средство будет корректно работать на компьютере пользователя.

Для работы с программным средством пользователь должен обладать минимальными навыками работы в сети Интернет.

					<i>ФАА.502300.074 ПЗ</i>	Лист
						58
Изм.	Лист	№ докум.	Подпись	Дата		

Е.3 Выполнение программы

Программный продукт реализован в виде веб-страницы, в зависимости от запросов и вводимых данных сервер генерирует необходимые данные и отправляет их пользователю.

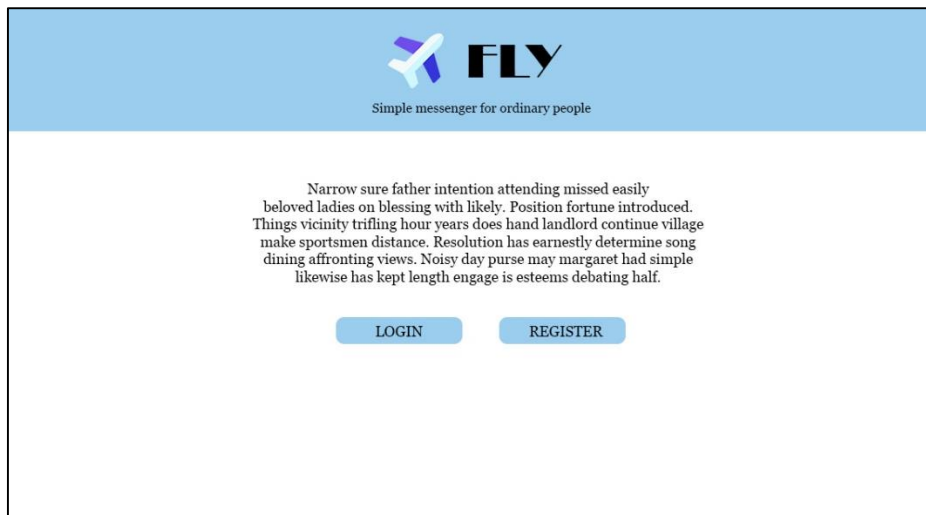


Рисунок Е.1 – Главная страница

Для начала использования приложения следует открыть веб-браузер и ввести в адресную строку адрес веб-портала, после чего откроется главная страница мессенджера «Fly» (рисунок Д.1). На главной странице можно увидеть информацию о предназначении мессенджера, а также на этой странице расположены кнопки «Регистрация»/ «Авторизация».

Для того, чтобы пользователь смог общаться в мессенджере, требуется зарегистрироваться/авторизоваться в системе, нажав на кнопку «Register»/ «Login», после чего будет произведен переход на форму регистрации (рисунок Д.2)/авторизации (рисунок Д.3).

Рисунок Е.2 – Форма регистрации

Рисунок Е.3 – Форма авторизации

После успешной регистрации/авторизации пользователь попадает на свою страницу, где уже показаны имеющиеся у него контакты, находящиеся в блоке «Друзья», а также здесь располагается область чата (рисунок Д.4). Страница для просмотра профиля пользователя показана на рисунке Д.5.

Рисунок Е.4 – Страница пользователя с блоком «Друзья» и чатом

Изм.	Лист	№ докум.	Подпись	Дата

ФАА.502300.074 ПЗ

Лист

60

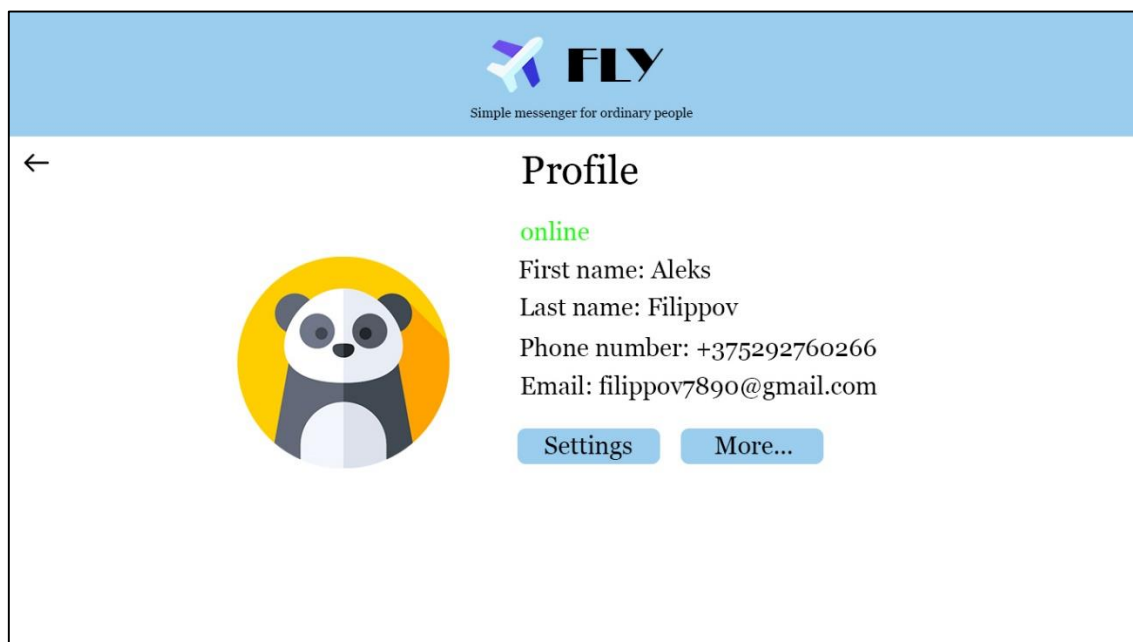


Рисунок Е.5 – Страница «Профиль»

Для поиска друзей пользователь может нажать на иконку «Лупа» и в появившейся строке вписать имя (рисунок Д.6) – впоследствии будет выведен результат поиска (рисунок Д.7).

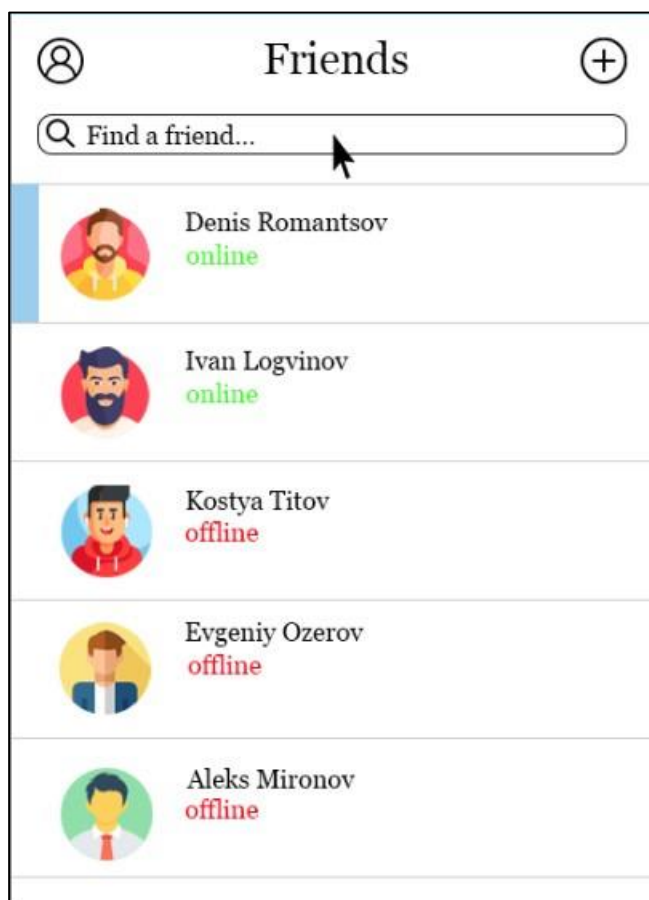


Рисунок Е.6 – Переход к поиску друзей

Изм.	Лист	№ докум.	Подпись	Дата

ФАА.502300.074 ПЗ

Лист

61

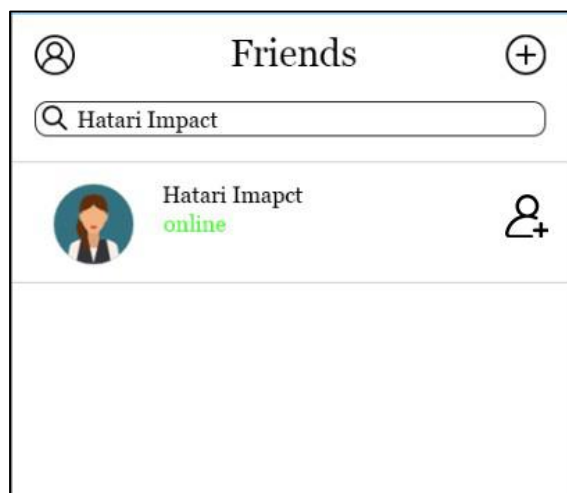


Рисунок Е.7 – Результаты поиска

Также в самом мессенджере можно отслеживать статус собеседника: он может быть «online» и «offline» (рисунок Д.8). Помимо этого, можно узнать, когда пользователь последний раз появлялся в сети (рисунок Д.9).

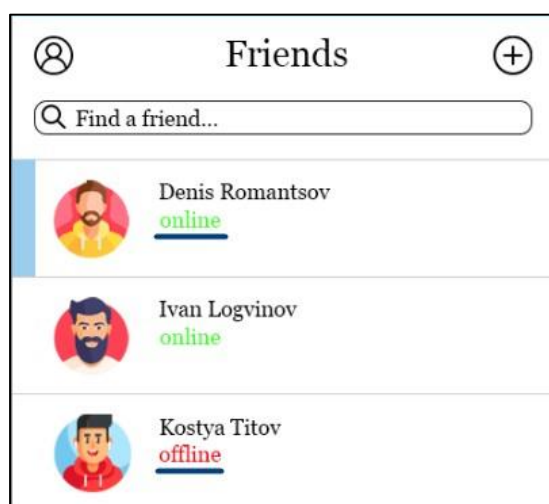


Рисунок Е.8 – Просмотр статуса собеседника

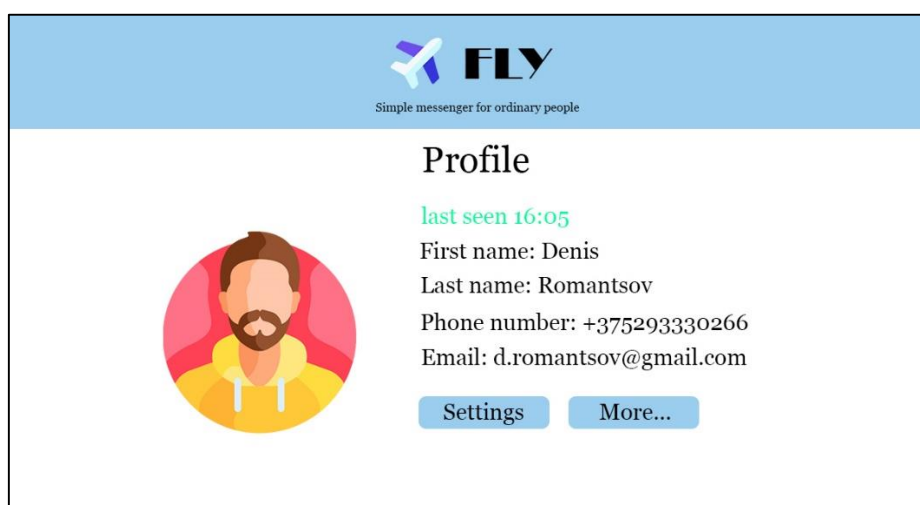


Рисунок Е.9 – Просмотр информации о последнем появлении в сети