

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО»
ВШ программной инженерии



ПОЛИТЕХ
Санкт-Петербургский
политехнический университет
Петра Великого

КУРСОВАЯ РАБОТА
по дисциплине «Программная инженерия»

Студентка, 3530202/90002

Касимова К.М.

Руководитель

Самочадин А.В.

Санкт-Петербург
2021 г.

Оглавление

Введение	4
1. Анализ технических требований и уточнение спецификаций	6
1.1. Выбор и описание методологии разработки	6
1.2. Выбор и описание технологического стека	7
1.3. Разработка вариантов использования	10
1.4. Пользовательские требования к ПО, составленные на основе сценариев вариантов использования	15
1.5. Детальные требования (SRS или фрагмент технического задания)	17
1.6. Оценка трудозатрат на выполнение проекта по UCP	20
1.7. План проекта	23
1.8. Перечень работ, выполненных автором курсовой работы	26
2. Проектирование структуры и компонентов программного продукта ..	27
2.1. Архитектура разрабатываемого ПО	27
2.2. Разработка концептуальной модели предметной области	29
2.3. Разработка интерфейса пользователя	31
2.4. Проектирование БД	36
2.5. Проектирование компонентов	37
2.6. Реализация компонентов	39
3. Выбор стратегии тестирования и разработка тестов	41
3.1. Выбор стратегии тестирования	41
3.2. Тест план	42
Заключение	43
Литература	44

Приложение А. Руководство пользователя	45
Приложение Б. Coding standard	46
Приложение В. Фрагменты текстов программ.....	47
Приложение Г. Отчет о code review	49
Приложение Д. Описание тестирования модуля по методу базового пути	50

Введение

Все чаще в своей жизни я стала замечать, как многие из моих знакомых и друзей не просыпаются вовремя. Для некоторых это вовсе невозможно без сторонней помощи. Чаще всего кто-то может просто забыть о том во сколько у него занятие, либо перепутать расписание. Иногда и я забывала поставить будильник, чтобы встать на учебу, но мне помогали проснуться мои друзья. Именно поэтому эта тема задела меня, и мы с ребятами решили создать приложение.

Проект заключается в создании будильника на платформе iOS. В основу приложения входит умный будильник, который можно просто завести на определённое время, а также не заводить вовсе, так как он будет подстраиваться под расписание. Пользователь должен будет ввести номер группы, а также за время, за которое ему нужно проснуться до занятия. Если человек не реагирует на сигнал будильника в течение 15 минут, то его близким, которых он укажет в приложении будет сделана рассылка.

№	Требование
1	Требования к приложению: <ul style="list-style-type: none">• Приложение должно предоставлять пользователю возможность регистрации• Приложение должно обладать понятным интерфейсом и удобной системой навигации
2	Требования к разделам приложения: Основные разделы: <ul style="list-style-type: none">• Страница пользователя• Страница будильников• Страница настроек Дополнительные разделы: <ul style="list-style-type: none">• Страница верификации• Страница регистрации
3	Требования к разделу с будильниками: <ul style="list-style-type: none">• Упорядочивание по времени• Простое добавление или удаление• Расширенные настройки
4	Требования к разделу с настройками:

	<ul style="list-style-type: none"> • Возможность изменить имя пользователя • Возможность изменить номер группы • Возможно изменить список доверенных номеров • Возможность выйти из аккаунта • Возможность настроить время пробуждения • Возможность изменить сообщение
5	<p>Требования к разделу пользователя:</p> <ul style="list-style-type: none"> • Выводится необходимая информация (имя пользователя, во сколько первое занятие, сколько в данный день занятий, какой день недели)

Таблица 1. Высокоуровневые требования к проекту.

1. Анализ технических требований и уточнение спецификаций

1.1. Выбор и описание методологии разработки

Выбранные методологии:

- Scrum (отслеживание недельных спринтов)
- Kanban (разбиение на задачи для каждого участника)

Scrum	Kanban
<p>Scrum – минимально необходимый набор мероприятий, артефактов, ролей, на которых строится процесс SCRUM-разработки, позволяющий за фиксированные небольшие промежутки времени, называемые спринтами, предоставить конечному пользователю работающий продукт с новыми бизнес-возможностями.</p> <p>В методологии Scrum команда является самоорганизующейся и самоуправляемой. Команда берет на себя обязательства по выполнению объема работы на спринт перед заказчиком.</p>	<p>Kanban – метод управления разработкой, реализующий принцип “точно в срок” и способствующий равномерному распределению нагрузки между работниками. При данном подходе весь процесс разработки прозрачен для всех членов команды. Задачи по мере поступления заносятся в отдельный список, откуда каждый разработчик может извлечь требуемую задачу.</p> <p>Kanban позволяет сделать процессы прозрачными, и непрерывно улучшать их на основе получаемых данных. Этот метод позволяет сфокусироваться на главном – непрерывном совершенствовании продукта и процессов.</p>

Таблица 2. Описание методологий разработки.

1.2. Выбор и описание технологического стека

- Среда разработки
 - Xcode
- Язык программирования
 - Swift
- Фреймворки
 - SwiftUI
 - Combine
 - UserNotifications
 - Foundation
 - AVFoundation
 - SwiftSoup
 - Firebase
- Разработка интерфейсов
 - Figma
- Управление базой данных и аутентификаций
 - Firebase
- Отслеживание задач по спринтам
 - Jira
- Контроль версий
 - GitHub

Подробно ознакомиться со всеми применяемыми инструментами можно в *Таблице 3*.

Инструмент	Описание
Среда разработки	Xcode – интегрированная среда разработки (IDE) для программного обеспечения для платформ macOS, iOS, watchOS и tvOS, разработанная корпорацией Apple.
Язык программирования	Swift – это надежный и интуитивно понятный язык программирования от

	Apple, при помощи которого можно создавать приложения для iOS, macOS, watchOS, tvOS. Он предоставляет разработчикам небывалую свободу творчества.
Фреймворки	<p>SwiftUI – это набор инструментов для создания пользовательского интерфейса, который позволяет нам декларативно разрабатывать приложения.</p> <p>Combine – это интересный фреймворк, который позволяет нам использовать мощь реактивного программирования без необходимости вводить какие-либо сторонние зависимости, что, в свою очередь, позволяет нам создавать логику, которая автоматически реагирует на изменения значений с течением времени.</p> <p>UserNotifications – это набор API, который предоставляет унифицированный, объектно-ориентированный способ работы с локальными и удаленными уведомлениями на этих платформах.</p> <p>Foundation – это фреймворк, который обеспечивает базовый уровень функциональности для приложений и платформ, включая хранение и постоянство данных, обработку текста, вычисление даты и времени, сортировку и фильтрацию, а также работу в сети.</p> <p>AVFoundation – предоставляет полный набор функций для обработки, синтезированию, контролю, импорту и экспорту аудиовизуальных носителей на платформах Apple.</p> <p>SwiftSoup – это чистая кроссплатформенная библиотека Swift (macOS, iOS, tvOS, watchOS,</p>

	<p>Linux) для работы с реальным HTML. Он предоставляет очень удобный API для извлечения данных и управления ими с использованием лучших методов, подобных DOM, CSS, jQuery. SwiftSoup реализует спецификацию WHATWG HTML5 и анализирует HTML в той же модели DOM, что и современные браузеры.</p> <p>Firebase – это фреймворк для разработки приложений с инструментами, которые помогут вам создавать, развивать и монетизировать ваше приложение.</p>
Разработка интерфейсов	<p>Figma – онлайн-сервис для разработки интерфейсов и прототипирования с возможностью организации совместной работы в режиме реального времени.</p>
Управление базой данных	<p>Firebase – это одно из BaaS-решений (Backend as a Service), которое дает разработчику массу возможностей. Это сервер, база данных, хостинг, аутентификация и не только в одной платформе.</p>
Отслеживание задач по спринтам	<p>Jira – коммерческая система отслеживания ошибок, предназначена для организации взаимодействия с пользователя, хотя в некоторых случаях используется и для управления проектами.</p>
Контроль версий	<p>GitHub – крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки. Веб-сервис основан на системе контроля версий Git.</p>

Таблица 3. Описание технологического стека.

1.3. Разработка вариантов использования

Действующие лица:

- Зарегистрированный пользователь
- Неверифицированный пользователь
- Незарегистрированный пользователь
- Система

Варианты использования для зарегистрированного пользователя:

- Посмотреть персональные данные
- Настроить персональные данные
- Посмотреть актуальные данные по расписанию
- Добавить/настроить/удалить будильник
- Редактировать список доверительных номеров
- Настроить доверительный номер
- Выйти из системы

Варианты использования для неверифицированного пользователя:

- Верификация

Варианты использования для незарегистрированного пользователя:

- Регистрация

Варианты использования для системы:

- Отправить OTP
- Установить уведомление-будильник

Диаграмма вариантов использования:

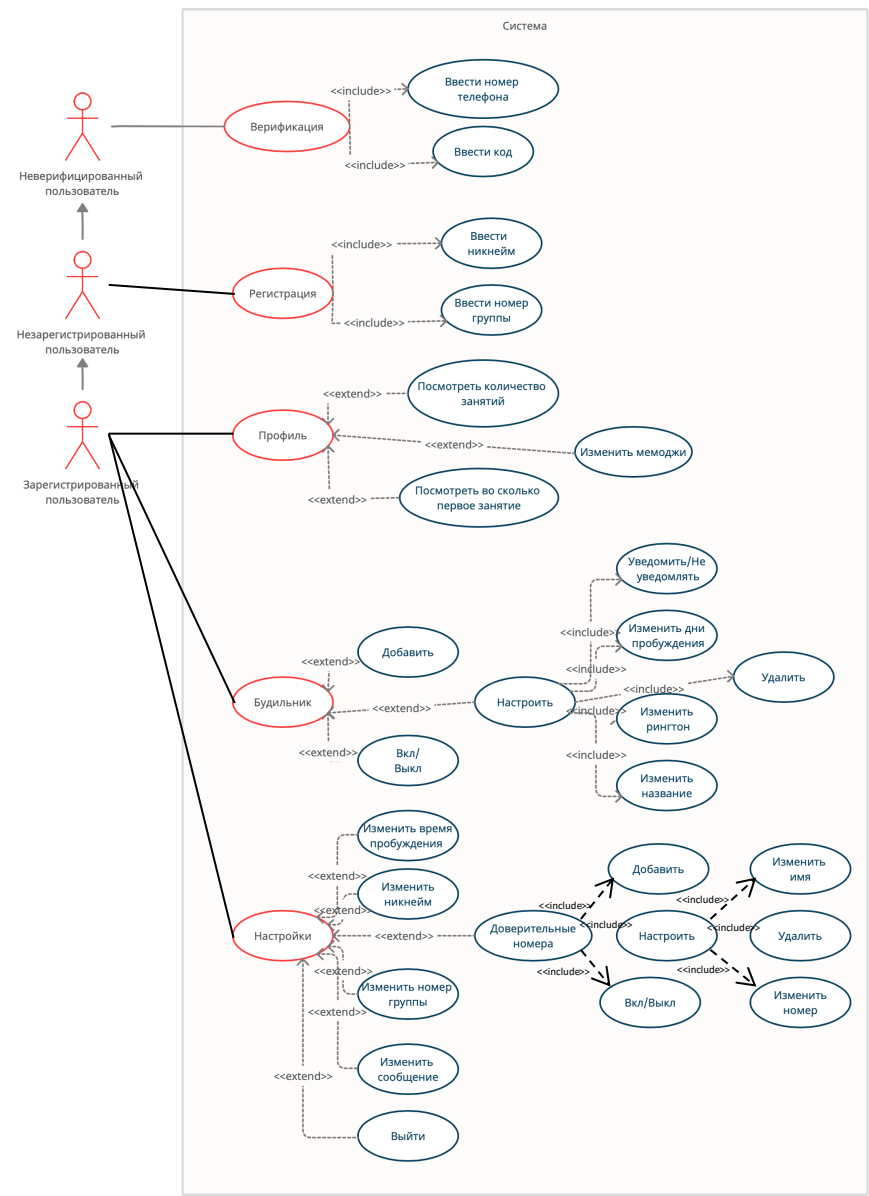


Рисунок 1. UseCase диаграмма.

Сценарии вариантов использования:

1. Сценарий процесса верификации неverified пользователя

Основное действующее лицо	Неверифицированный пользователь
Гарантия успеха	Пользователь верифицировался
Триггер	Пользователь впервые открывает приложение Пользователь вышел из системы

Основной сценарий	<ol style="list-style-type: none"> 1. Система предлагает пользователю ввести номер телефона 2. Пользователь вводит номер 3. Пользователь нажимает на кнопку “Verify” 4. Система отправляет пользователю OTP 5. Система переводит пользователя на страницу ввода код-пароля 6. Пользователь вводит полученный код-пароль 7. Пользователь нажимает на кнопку “Let`s go!” 8. Система верифицирует пользователя
Расширения	<ol style="list-style-type: none"> 2а. Пользователь некорректно вводит номер телефона <ol style="list-style-type: none"> 2а1. Система выводит сообщение об ошибке и предоставляет форму еще раз 4а. Пользователь не получил код-пароль <ol style="list-style-type: none"> 4а1. Пользователь нажимает на кнопку “Back to number entry”, чтобы убедиться в корректности введенного номера телефона и получения OTP 6а. Пользователь некорректно вводит код-пароль

2. Сценарий процесса регистрации незарегистрированного пользователя

Основное действующее лицо	Незарегистрированный пользователь
Гарантия успеха	Пользователь зарегистрировался
Триггер	Пользователь прошел процедуру верификации
Основной сценарий	<ol style="list-style-type: none"> 1. Система предлагает ввести имя пользователя 2. Пользователь вводит имя 3. Система предлагает ввести номер группы 4. Пользователь вводит номер группы 5. Пользователь нажимает на кнопку “Let`s go!” 6. Система регистрирует пользователя
Расширения	<ol style="list-style-type: none"> 2а. Пользователь некорректно вводит имя <ol style="list-style-type: none"> 2а1. Система выводит сообщение об ошибке 4а. Пользователь некорректно вводит номер группы <ol style="list-style-type: none"> 4а1. Систем выводит сообщение об ошибке

3. Сценарий процесса настройки данных зарегистрированным пользователем

Основное действующее лицо	Зарегистрированный пользователь
---------------------------	---------------------------------

Гарантия успеха	Пользователь настроил свои данные
Триггер	Пользователь переходит на страницу настроек
Основной сценарий	<ol style="list-style-type: none"> 1. Пользователь нажал на кнопку “Wake up time before class” 2. Система предлагает пользователю ввести за сколько минут до занятия устанавливать будильники 3. Пользователь вводит количество минут 4. Пользователь нажимает кнопку “Save” 5. Пользователь нажал на кнопку “Change nickname” 6. Система предлагает пользователю ввести новое имя 7. Пользователь вводит новое имя 8. Пользователь нажимает кнопку “Save” 9. Пользователь нажал на кнопку “Group number” 10. Система предлагает пользователю изменить номер группы 11. Пользователь вводит новый номер группы 12. Пользователь нажимает кнопку “Save”
Расширения	<ol style="list-style-type: none"> 3а. Пользователь вводит не число <ol style="list-style-type: none"> 3а1. Система выводит сообщение об ошибке и вновь предоставляет форму для заполнения 3б. Пользователь вводит число превышающее значение 300 (5 часов) <ol style="list-style-type: none"> 3б1. Система выводит сообщение об ошибке и вновь предоставляет форму для заполнения 7а. Пользователь оставляет поле с именем пустым <ol style="list-style-type: none"> 7а1. Система оставляет прежнее имя 10а. Пользователь оставляет поле с номером группы пустым <ol style="list-style-type: none"> 10а1. Система оставляет прежний номер группы

4. Сценарий процесса добавления будильника зарегистрированным пользователем

Основное действующие лицо	Зарегистрированный пользователь
Гарантия успеха	Пользователь добавил будильник
Триггер	Пользователь переходит на страницу будильников
Основной сценарий	<ol style="list-style-type: none"> 1. Пользователь нажал на кнопку “Add alarm”

	<ol style="list-style-type: none"> 2. Система предлагает пользователю выбрать время пробуждения 3. Пользователь выбирает время 4. Система предлагает пользователю выбрать дни повтора будильника 5. Пользователь выбирает дни повтора 6. Система предлагает пользователю выбрать рингтон 7. Пользователь выбирает рингтон 8. Система предлагает пользователю выбрать название будильника 9. Пользователь вводит название 10. Пользователь подтверждает введение данные название нажатием на кнопку “Save” 11. Пользователь подтверждает добавление нового будильника нажатием на кнопку “Save” 12. Система добавляет будильник, и он успешно отображается на странице будильников
Расширения	<p>12а. Пользователь оставляет поле с названием будильника пустым</p> <p>12а1. Система устанавливает название будильника по умолчанию</p>

5. Процесс удаления будильника зарегистрированным пользователем

Основное действующие лицо	Зарегистрированный пользователь
Гарантия успеха	Пользователь удалил будильник
Триггер	Пользователь перешел на страницу будильников
Основной сценарий	<ol style="list-style-type: none"> 1. Пользователь выбирает будильник, который ему необходимо удалить 2. Пользователь нажимает на кнопку настроек на выбранном будильнике 3. Система предлагает пользователю настроить параметры будильника 4. Пользователь нажимает на кнопку “Delete” 5. Система удаляет будильник

1.4. Пользовательские требования к ПО, составленные на основе сценариев вариантов использования

- **Возможность верификации**
 - При верификации пользователь указывает номер телефона
 - Должна быть предусмотрена процедура проверки номера телефона
 - Должна быть предусмотрена процедура проверки ОТР
 - Должна быть предусмотрена процедура повторной отправки ОТР
- **Возможность регистрации**
 - При регистрации пользователь указывает имя (никнейм) и номер группы
- **Возможность отправки уведомлений (будильников)**
 - При первом открытии приложения пользователь соглашается на отправку уведомлений
- **Возможность добавления будильников**
 - Пользователь указывает время, дни повтора, звук и название будильника
 - Должна быть предусмотрена процедура автоматического заполнения полей значениями по умолчанию
 - Должна быть предусмотрена процедура проверки заполнения поля названия будильника
 - Должна быть предусмотрена процедура сохранения изменений
 - Должна быть предусмотрена процедура отмены добавления
- **Возможность настройки будильников**
 - Пользователь имеет возможность настроить время, дни повтора, звук и название будильника
 - Должна быть предусмотрена процедура проверки заполнения поля названия будильника
 - Должна быть предусмотрена процедура сохранения изменений

- Должна быть предусмотрена процедура отказа от сохранения изменений
- Должна быть предусмотрена процедура удаления будильника
- **Возможность настройки персональных данных пользователя**
 - Пользователю должна быть предоставлена возможность изменить время пробуждения до занятия, никнейм, номер группы, текст сообщения для рассылки
- **Возможность добавления доверительных номеров**
 - Пользователь указывает имя и номер телефона
 - Должна быть предусмотрена процедура проверки заполнения поля имени и номера телефона
 - Должна быть предусмотрена процедура сохранения изменений
 - Должна быть предусмотрена процедура отмены добавления
- **Возможность настройки доверительных номеров**
 - Пользователь имеет возможность имя и номер телефона
 - Должна быть предусмотрена процедура проверки заполнения поля имени и номера телефона
 - Должна быть предусмотрена процедура сохранения изменений
 - Должна быть предусмотрена процедура отказа от сохранения изменений
 - Должна быть предусмотрена процедура удаления доверительного номера
- **Возможность выхода из системы**
 - Система должна удостовериться в намерении пользователя выйти из системы

Приведенные требования можно детализировать, с помощью составления фрагмента **Software Requirement Specifications (SRS)**.

1.5. Детальные требования (SRS или фрагмент технического задания)

Ссылка на URS	№ требования	Требование
1.	01.01	Верификация пользователя Вход в систему должен базироваться на верификации пользователя. Пользователь верифицируется с помощью номера мобильного телефона.
1.	01.02	Окно верификации – поле Окно верификации должно содержать поле: <i>Номер телефона</i> – это поле должно содержать цифровую последовательность длиной 10 символов, так как код страны устанавливается автоматически.
1.	01.03	База данных После успешного прохождения верификации система должно внести в базу данных номер телефона пользователя.
2.	02.01	Регистрация пользователя Пользователь регистрируется с помощью никнейма и номера группы
2.	02.02	Окно регистрации – поля Окно верификации должно содержать следующие поля: <i>Имя</i> – это поле должно содержать символьную последовательность длиной не более 15 символов <i>Группа</i> – это поле должно содержать цифровую последовательность и знак слэша.
2	02.03	База данных После успешного прохождения регистрации система должна внести в базу никнейм и номер группы пользователя.
3	03.01	Условие пользовательского соглашения Пользователь обязан разрешить доступ к уведомлениям
4	04.01	Окно добавления нового будильника – поле/переключатели/кнопки Окно добавления будильника должно содержать часы и минуты. <i>Время</i> – это скрол-колесо должно содержать часы и минуты. <i>Дни повтора</i> – эти семь кнопок должны содержать названия дней недели.

		<p><i>Рингтон</i> – эти шесть кнопок должны содержать название звуков.</p> <p><i>Название</i> – это поле должно содержать символьную последовательность длиной не более 15 символов.</p> <p><i>Уведомление доверительных номеров</i> – этот переключатель должен поддерживать два состояния: включен/выключен.</p>
4.	04.02	<p>Окно добавления нового будильника – значения по умолчанию</p> <p>Окно добавления будильника должно содержать поле/кнопки/переключатели со значениями по умолчанию:</p> <p><i>Время</i> – текущее.</p> <p><i>Дни повтора</i> – не установлены.</p> <p><i>Рингтон</i> – “Radar”</p> <p><i>Название</i> – “Alarm”</p> <p><i>Уведомление доверительных номеров</i> - выключен</p>
5.	05.01	<p>Окно настройки будильника – поле/переключатели/кнопки</p> <p>Аналогично требованию 04.01</p>
6.	06.01	<p>Окно настроек данных пользователя – поля</p> <p>Окно настроек данных пользователя должно содержать поля:</p> <p><i>Время пробуждения до занятия</i> – это поле должно содержать число минут от 1 до 300.</p> <p><i>Никнейм</i> – это поле должно содержать символьную последовательность длиной не более 15 символов.</p> <p><i>Номер группы</i> – это поле должно содержать цифровую последовательность и знак слеша.</p> <p><i>Сообщение</i> – это поле должно содержать символьную последовательность длиной не более 25 символов.</p>
7.	07.01	<p>Условие пользовательского соглашения</p> <p>Пользователь обязан согласиться выйти из системы.</p>
8.	08.01	<p>Окно настроек доверительных номеров – поля</p> <p>Окно настроек данных доверительных номеров должно содержать поля:</p> <p><i>Имя</i> – это поле должно содержать символьную последовательность длиной не более 15 символов</p> <p><i>Номер телефона</i> - это поле должно содержать цифровую последовательность длиной 10 символов, так как код страны устанавливается автоматически.</p>

Таблица 4. Детальные требования к проекту.

1.6. Оценка трудозатрат на выполнение проекта по UCP

1. Определение весовых показателей действующих лиц:

- Пользователь – 3
- Зарегистрированный пользователь - 3
- Система настройки будильников – 1
- Система регистрации - 1

2. Вычисление неурегулированного весового показателя действующих лиц.

- $UAW = 3 * 2 + 1 * 2 = 8$

3. Вычисление весовых показателей вариантов использования

- Настройка будильников по умолчанию включая автоматическое добавление будильников по номеру группы – 18. Вес = 15
- Уведомление друга – 3. Вес = 5

4. Вычисление неурегулированных весовых показателей вариантов использования.

- $UUCW = 15 + 5 = 20$

5. Вычисление неурегулированного общего показателя

- $UUCP = UAW + UUCW = 28$

6. Определение технической сложности

- $TCF = 0,6 + (0,1 * \Sigma(\text{Вес}_i * T_i)) = 0,6 + (0,1 * 80) = 8,6$
- $T1 = 4$. Значение = 4. * = 16
- $T2 = 4$. Значение = 3. * = 12
- $T3 = 4$. Значение = 3. * = 12
- $T4 = 3$. Значение = 1. * = 3
- $T5 = 1$. Значение = 0. * = 0
- $T6 = 3$. Значение = 3. * = 9
- $T7 = 4$. Значение = 4. * = 16
- $T8 = 3$. Значение = 0. * = 0

- $T9 = 2$. Значение = 3. * = 6
- $T10 = 4$. Значение = 2. * = 4
- $T11 = 1$. Значение = 1. * = 1
- $T12 = 1$. Значение = 1. * = 1
- $T13 = 1$. Значение = 0. * = 0
- Сумма = 80

7. Определение влияния условий разработки

- $EF = 1,4 + (-0,03 * \Sigma(\text{Вес}_i * F_i)) = 1,4 + (-1,35) = 0,05$
- $F1 = 1$. Значение = 3. * = 3
- $F2 = 2$. Значение = 3. * = 6
- $F3 = 2$. Значение = 3. * = 6
- $F4 = 1$. Значение = 1. * = 1
- $F5 = 5$. Значение = 4. * = 20
- $F6 = 4$. Значение = 3. * = 12
- $F7 = 1$. Значение = 0. * = 0
- $F8 = -1$. Значение = 3. * = -3
- Сумма = 45

8. Вычисление UCP

- $UCP = UUCP * TCF * EF = 28 * 8,6 * 0,05 = 12,04$

9. Оценка трудоемкости

- $F1-F6 < 3 = 4$
- $F7-F8 > 3 = 0$
- Общее кол-во = 4

Значит, нужно 28 чел/час на одну единицу UCP. Тогда трудозатраты на проект составят = 337,12 часов или около 5,6 чел/мес (5 дней в неделю по 3 часа в день, т. е. проект будет выполнен за полтора месяца).

На деле, у нас получилось примерно 312 часов, мы работали 52 дня по 6 часов в среднем.

1.7. План проекта

1. Определить требования к результату – вся команда
2. Разделить роли в команде – Карина Касимова
3. Подготовка теоретических материалов – Алина Потапова
4. Изучение необходимых теоретических материалов – вся команда
5. Создание репозитория в GitHub – Давид Нигматуллин, Ренат Айдаров
6. Заполнение задач в Jira – Давид Нигматуллин, Ренат Айдаров
7. Определение образа продукта – Карина Касимова
8. Создание дизайна приложения – Карина Касимова
9. Создание иконки приложения – Карина Касимова
10. Создание архитектуры приложения – Карина Касимова
11. Создание схемы базы данных – Карина Касимова
12. Создание базы под верификацию – Карина Касимова
13. Создание базы данных под регистрацию – Карина Касимова
14. Логика аутентификации – Карина Касимова
15. Логика регистрации – Карина Касимова
16. Верстка страницы регистрации – Давид Нигматуллин, Ренат Айдаров
17. Верстка страницы верификации – Давид Нигматуллин, Ренат Айдаров
18. Связывание логики верификации и регистрации с интерфейсом –
Алина Потапова
19. Проверка полей на странице регистрации – Карина Касимова
20. Установка иконки – Давид Нигматуллин, Ренат Айдаров
21. Создание панели вкладок – Давид Нигматуллин, Ренат Айдаров
22. Верстка страницы пользователя – Алина Потапова
23. Анимация приветствия (и его верстка) – Алина Потапова
24. Перенос данных будильников и времени – Алина Потапова
25. Анимация мемов – Давид Нигматуллин, Ренат Айдаров
26. Логика будильников – Алина Потапова

27. Верстка страницы будильников – Алина Потапова
28. Верстка страницы настройки будильника – Алина Потапова
29. Верстка страницы добавления будильника – Алина Потапова
30. Изучение принципов работы локальных уведомлений – Алина Потапова
31. Добавление локальных уведомлений – Алина Потапова
32. Добавление рингтонов уведомлений (будильников) – Алина Потапова
33. Сортировка будильников при добавлении/удалении – Алина Потапова
34. Верстка страницы настроек – Алина Потапова
35. Логика настроек – Алина Потапова
36. Верстка входящих в страницу настроек интерфейсов – Алина Потапова
37. Верстка страницы доверительных номеров – Алина Потапова
38. Локальное хранение данных – Алина Потапова
39. Логика доверительных номеров (добавление, удаление, редактирование) – Алина Потапова
40. Логика отправки сообщений доверительным номерам – Карина Касимова
41. Разработка алгоритма парсинга – Карина Касимова
42. Связывание расписания с будильниками – Карина Касимова
43. Связывание расписание со страницей пользователя – Карина Касимова
44. Проверка полей на странице настройки, добавления будильника, настройки данных пользователя – Алина Потапова
45. Модульное тестирование – Алина Потапова
46. Функциональное тестирование – вся команда
47. Создание документации для курса ОПД – Давид Нигматуллин, Айдаров Ренат
48. Создание презентации для курса ОПД – Давид Нигматуллин
49. Создание презентации для защиты проекта – Карина Касимова

- 50. Видео-презентация работы приложения – Алина Потапова
- 51. Создание отчетов по проекту – вся команда
- 52. Защита проекта – вся команда

1.8. Перечень работ, выполненных автором курсовой работы

Работы из плана работ по проекту, которые были выполнены мной:

1. Определение требований к результату
2. Разделение ролей в команде
3. Изучение необходимых теоретических материалов
4. Определение образа продукта
5. Создание дизайна приложения
6. Создание иконки приложения
7. Создание архитектуры приложения
8. Создание схемы базы данных
9. Создание базы под верификацию
10. Создание базы данных под регистрацию
11. Логика аутентификации
12. Логика регистрации
13. Проверка полей на странице регистрации
14. Логика отправки сообщений доверительным номерам
15. Разработка алгоритма парсинга
16. Связывание расписания с будильниками
17. Связывание расписания со страницей пользователя
18. Функциональное тестирование
19. Создание презентации для защиты проекта
20. Создание отчета по проекту
21. Защита проекта

2. Проектирование структуры и компонентов программного продукта

2.1. Архитектура разрабатываемого ПО

Система будет предназначена для создания будильников с расширенным набором настроек, также первый будильник будет добавляться автоматически на основе расписания. В случае, если пользователь не реагирует на сигнал будильника – доверенным лицам будет сделана рассылка. Кроме этого, пользователь может посмотреть получить актуальную информацию по расписанию на странице пользователя. Конечно же, пользователь может изменить персональную информацию, список доверительных номеров, редактировать информацию о доверительном лице и выйти из аккаунта.

Пользователями системы на данный момент разработки будут студенты политехнического университета.

Пользователи будут взаимодействовать с нашей системой с помощью мобильного приложения на платформе iOS.

Данные будут храниться на сервере и кэшироваться в приложении с помощью баз данных. Под верификацию существует своя база данных, под регистрация своя.

Система будет импортировать данные из сервера. Если нужно, система обновит данные хранящиеся в локальной памяти устройства.

Сервисы:

- Верификация (аутентификация)
- Регистрация
- Пользовательская страница
- Будильники
- Настройки
- Уведомления

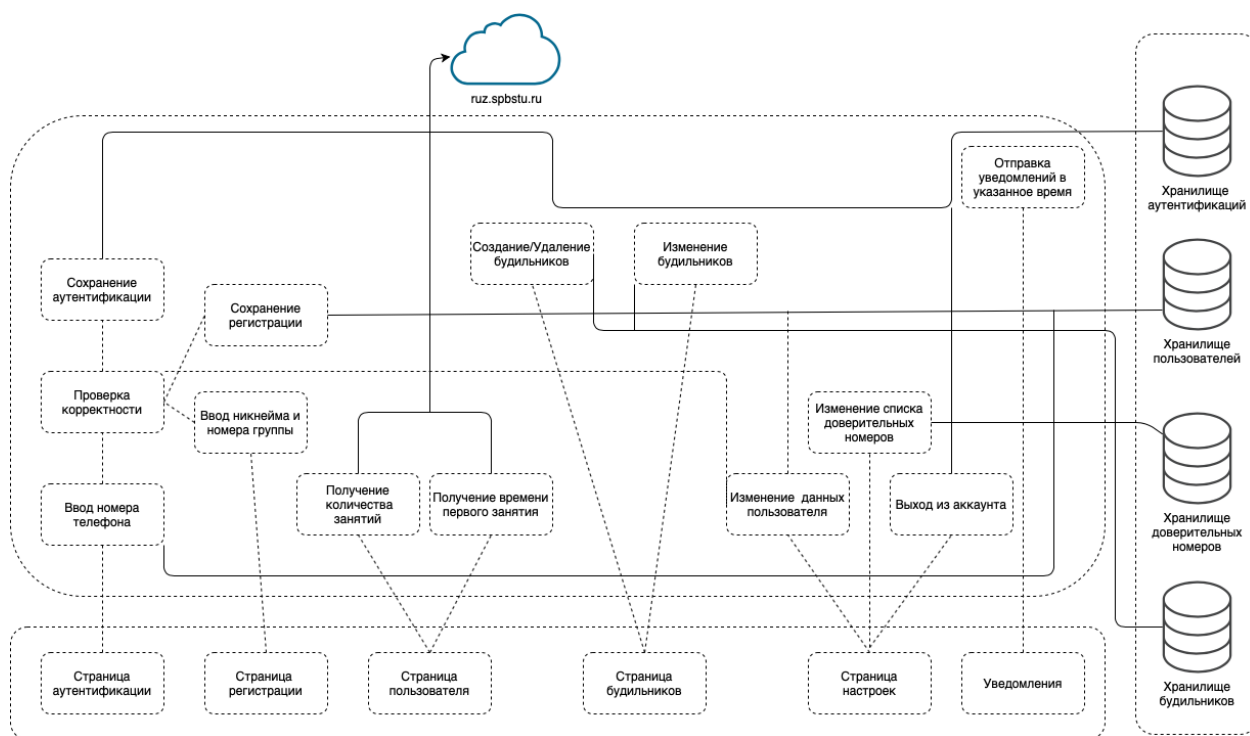


Рисунок 2. Архитектура разрабатываемого ПО.

2.2. Разработка концептуальной модели предметной области

Для подробного описания принципа взаимодействия компонентов ПО воспользуемся диаграммой классов:

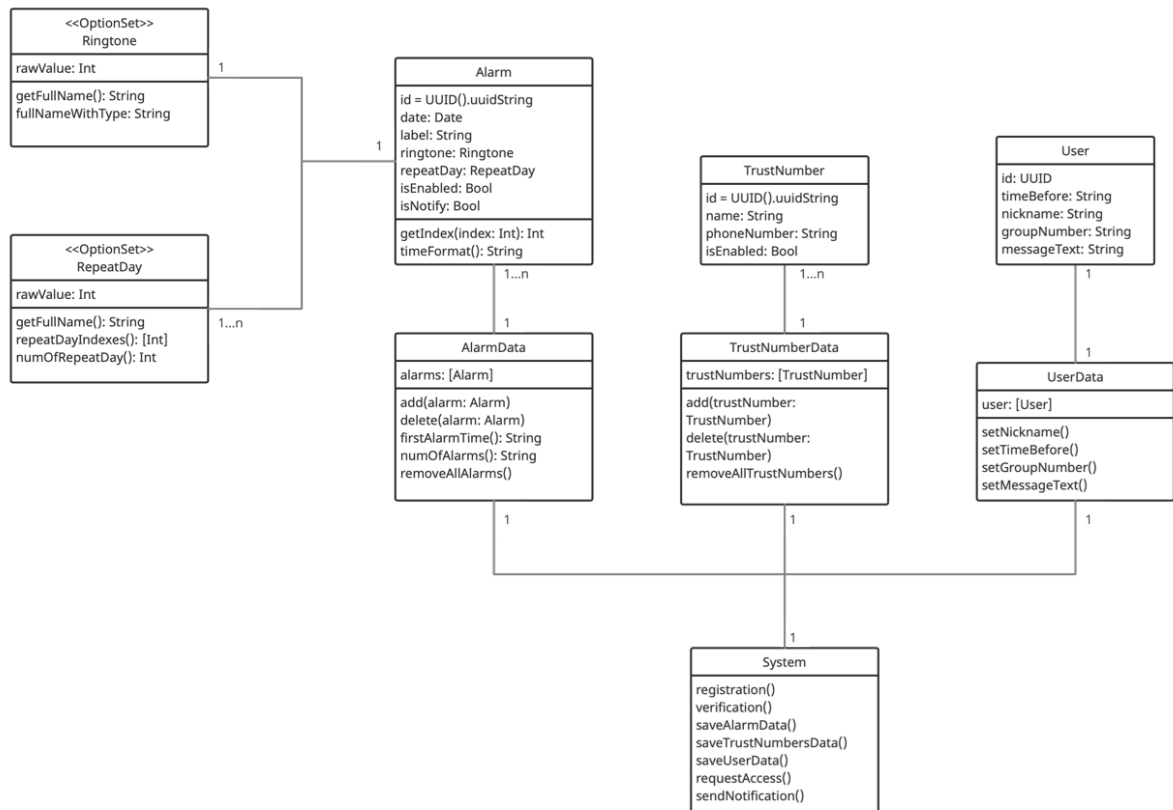


Рисунок 3. Диаграмма классов.

Подробно о каждом классе:

- Ringtone – используется для установки рингтона для будильника (по умолчанию “Radar”)
- RepeatDay – используется для выбора дней по которым будет повторяться будильник (по умолчанию повтора нет)
- Alarm – используется для работы с будильниками
- AlarmData – используется для работы с информацией будильника
- TrustNumber – используется для работы с доверенными номерами
- TrustNumber – используется для работы с информацией доверительного номера
- User – используется для установки персональных данных пользователя

- UserData – используется для работы с персональными данными пользователя
- System – используется для управления всеми классами, структурами, функциями и процедурами

2.3. Разработка интерфейса пользователя

Для создания интерфейса пользователя были разработаны следующие макеты экранов:



Рисунок 4. Дизайн страниц аутентификации, регистрации, OTP и логотипа.

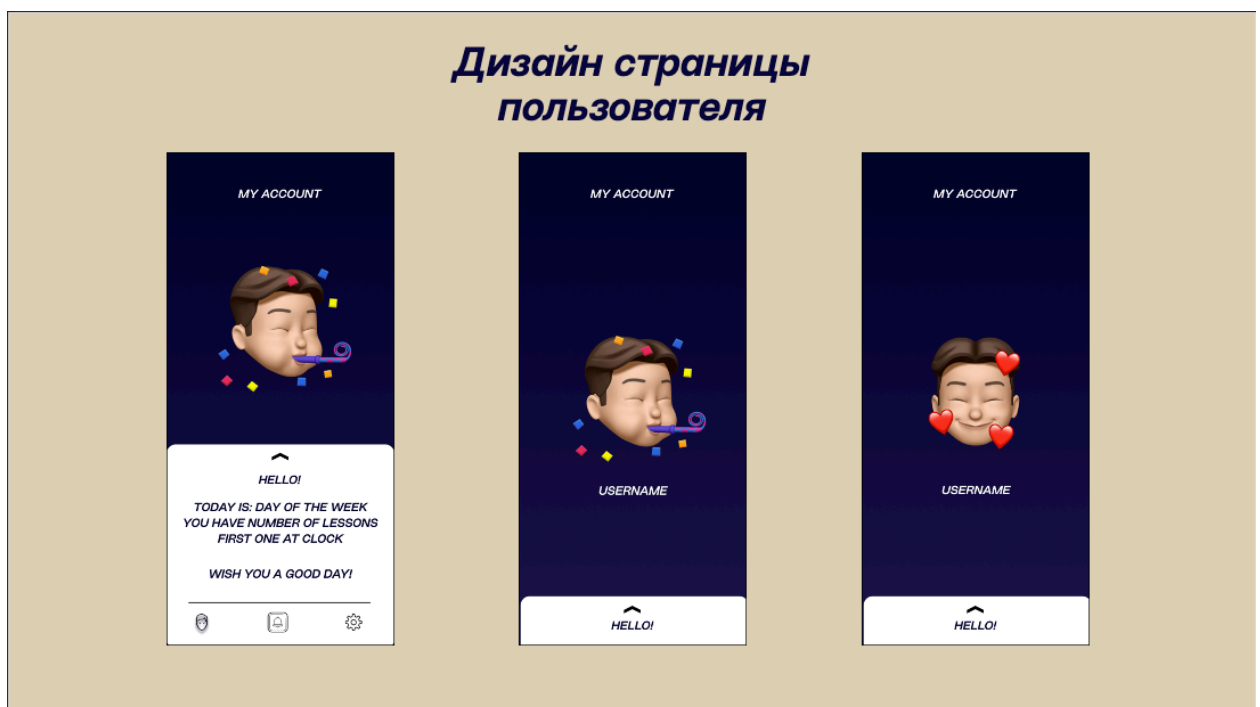


Рисунок 5. Дизайн страницы пользователя.

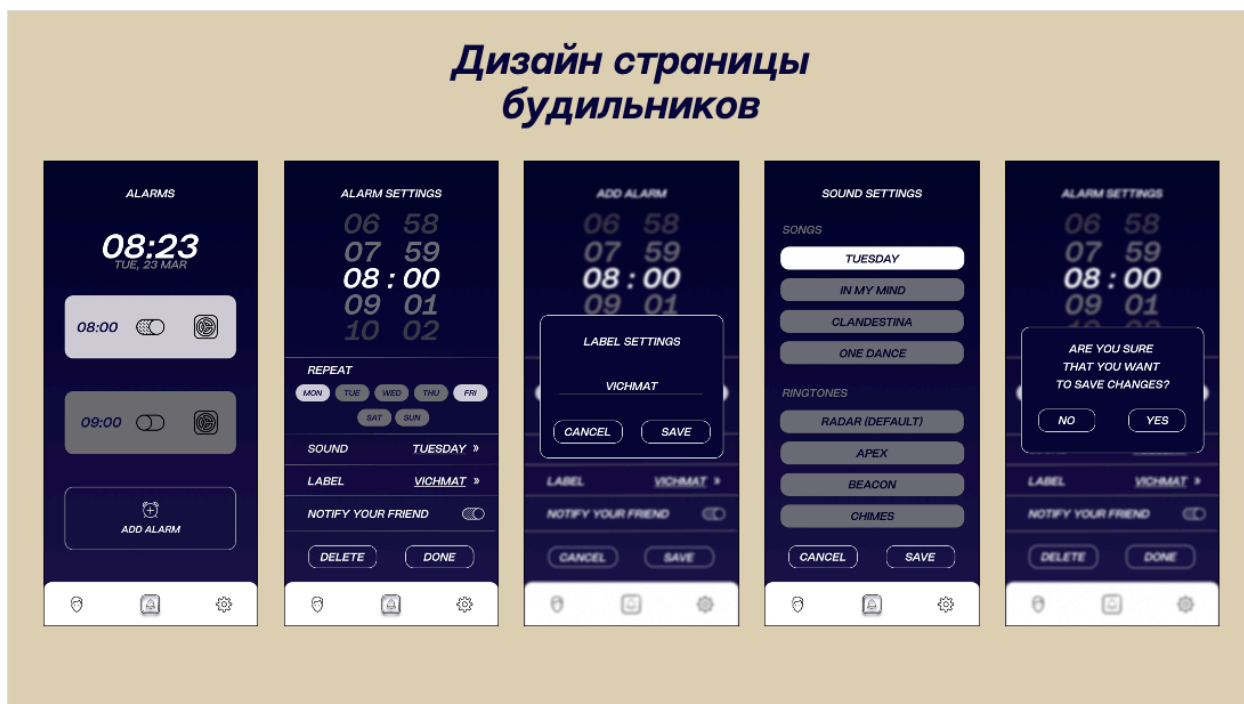


Рисунок 6. Дизайн страницы будильников, их настройки и добавления.



Рисунок 7. Дизайн страницы настроек и входящих в нее интерфейсов.

Некоторые макеты (страница будильников, страница доверительных номеров) являются прокручиваемыми.

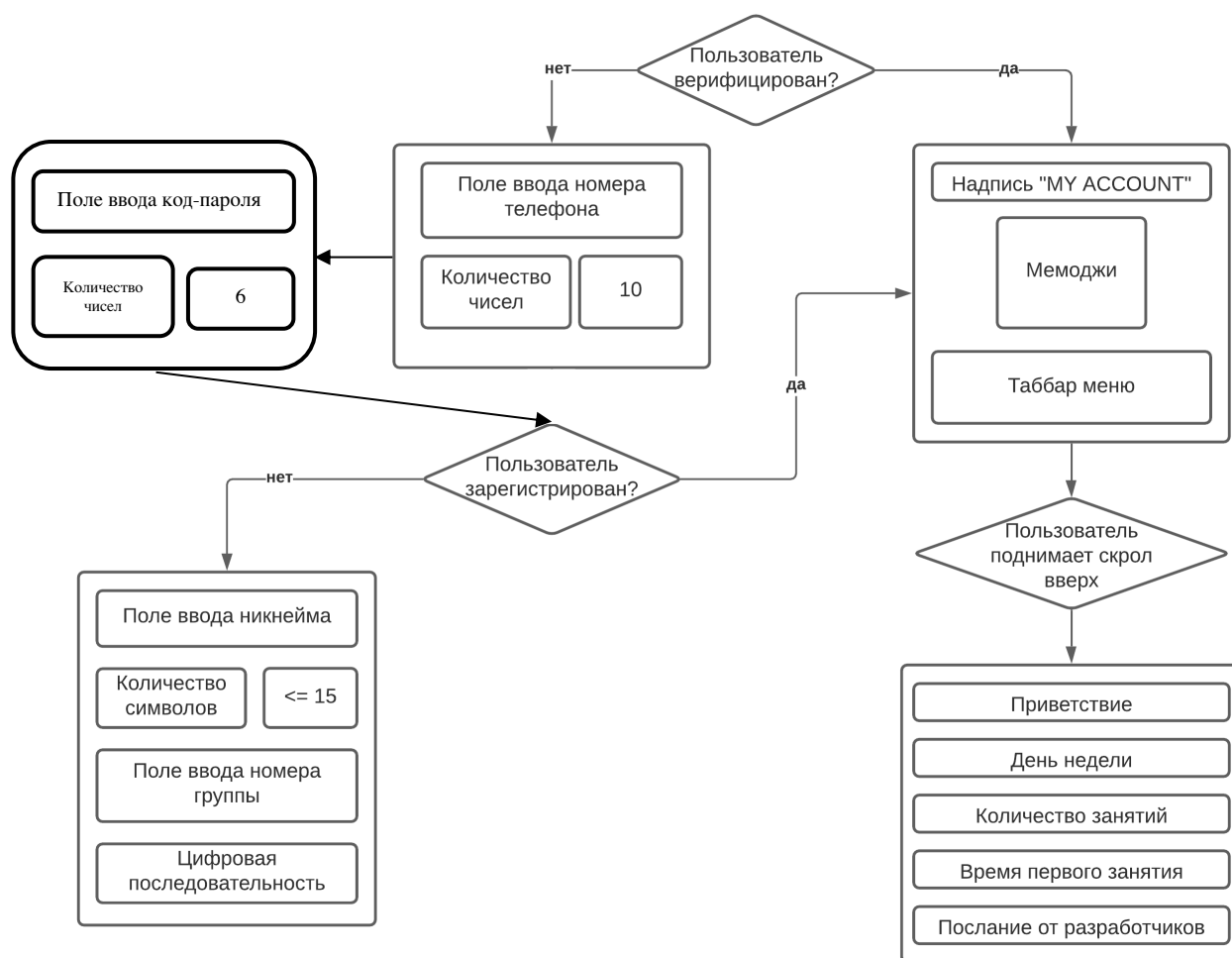


Рисунок 8. User-flow верификации и регистрации.

На данном User-flow предоставлены страницы верификации, регистрации и пользователя. Сначала идет проверка на верификацию, если пользователь верифицирован, то он переходит на страницу пользователя, иначе на страницу ввода код-пароля. После этого идет проверка на регистрацию, и аналогично верификации, пользователь переходит либо на страницу пользователя, либо на страницу регистрации и после на страницу пользователя. На главной странице пользователь может переключаться между страницами настроек и будильников, а также узнать всю актуальную информацию поднятием скrolла.

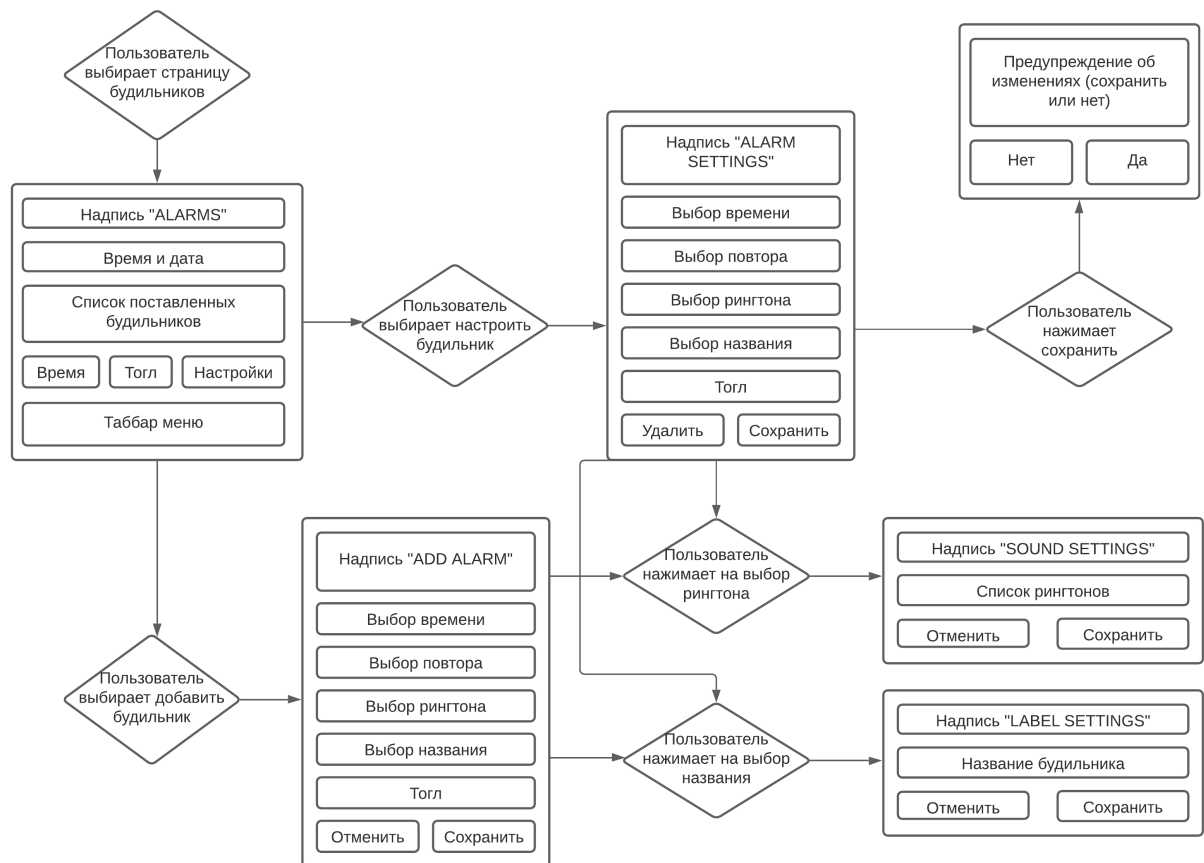


Рисунок 9. User-flow страницы будильников.

На данном User-flow предоставлена страница будильников. Когда пользователь переходит на нее он может ознакомиться с временем и датой, посмотреть список поставленных будильников, добавить или изменить будильник. Настройка и добавление будильника аналогичны: на этих страницах мы можем выбрать время, дни повторения, рингтон, название, переключить тогл на отправку сообщения доверительному лицу, и, в зависимости от страницы, либо удалить будильник, либо отменить добавление, а также сохранить изменения. При сохранении изменений выскакивает предупреждение об изменениях, с которым пользователь либо соглашается, либо нет. При настройке рингтона или названия выскакивает отдельное вью с настройками для них.

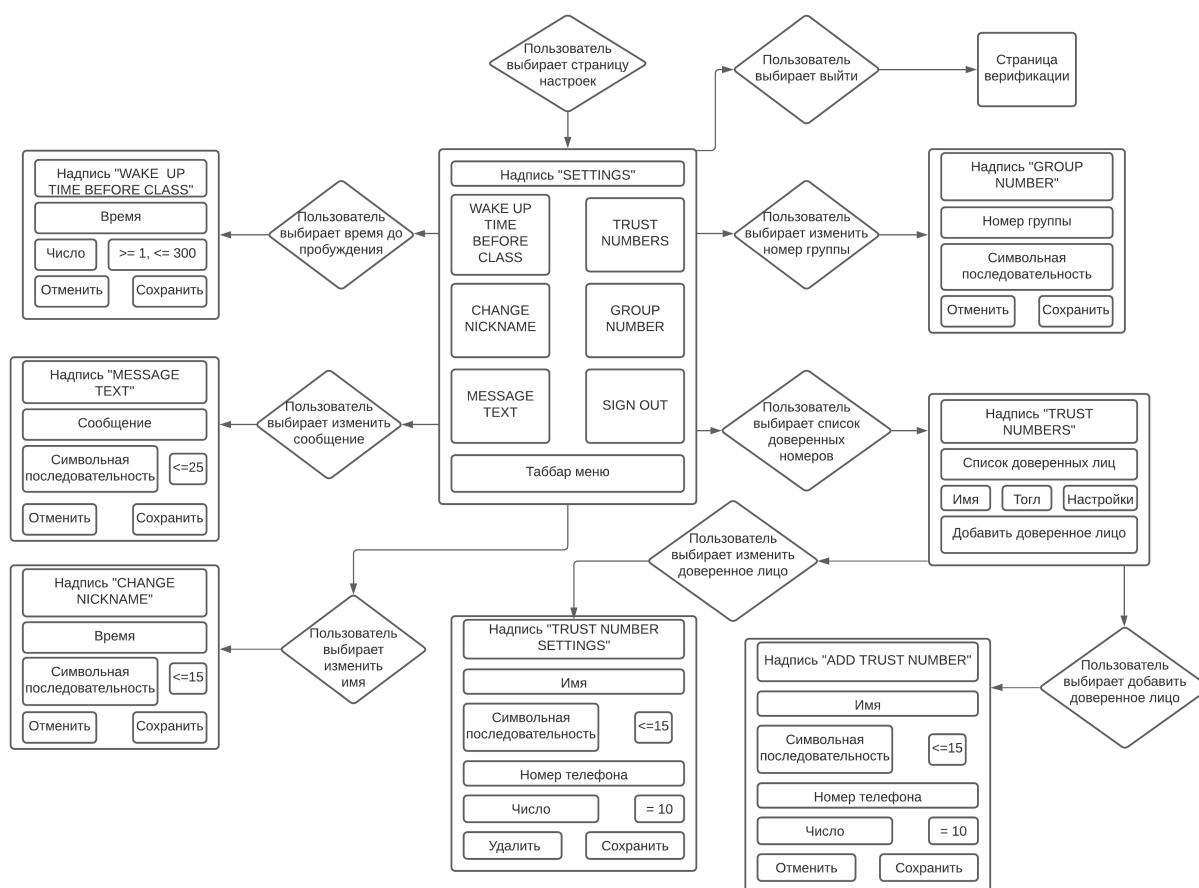
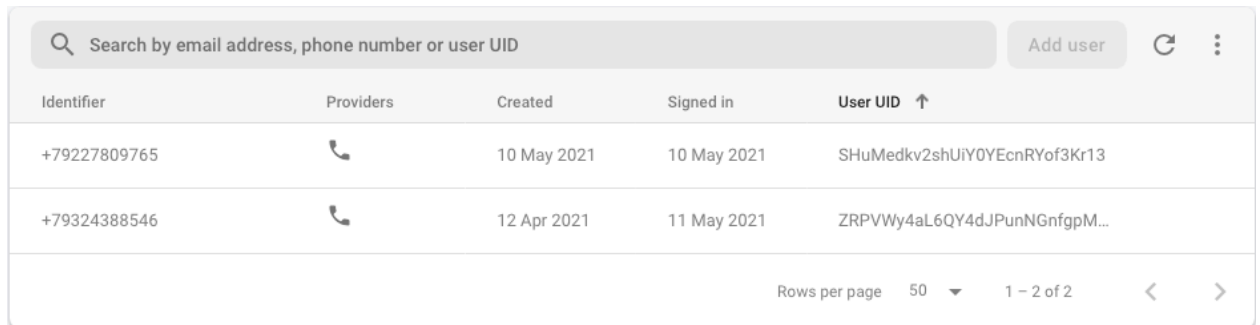


Рисунок 10. User-flow страницы настроек.

На данном User-flow предоставлена страница настроек. Когда пользователь переходит на нее, перед ним появляется меню настроек, в котором он может изменить время пробуждения, имя, номер группы, сообщение, список доверительных номеров и выйти из аккаунта. При изменении времени пробуждения, имени, номера группы и сообщения выскакивает соответствующее вью для изменения. При переходе на страницу со списком доверительных номеров мы видим список доверительных лиц, которые мы можем либо отключить, либо изменить, а также есть функция добавления доверительного лица. При добавлении и изменении выскакивают аналогичные вью, в которых мы вводим имя и номер телефона, и, в зависимости от страницы, либо удаляем доверенное лицо, либо отменяем добавление, а также сохраняем изменения. При выходе из аккаунта мы переходим на страницу верификации, которая была предоставлена выше.

2.4. Проектирование БД

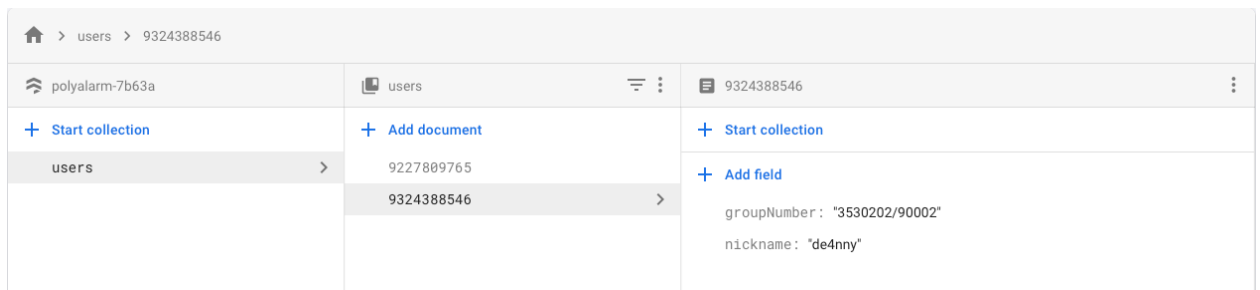
Для работы с пользователем у нас существует две базы данных: регистрации и аутентификации.



Identifier	Providers	Created	Signed in	User UID ↑
+79227809765	📞	10 May 2021	10 May 2021	SHuMedkv2shUiY0YEcnRYof3Kr13
+79324388546	📞	12 Apr 2021	11 May 2021	ZRPVWy4aL6QY4dJPunNGnfgpM...

Рисунок 11. База данных аутентификации.

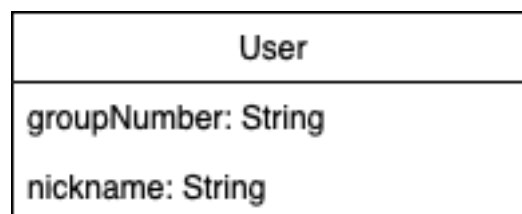
Что касается базы данных аутентификации, то в ней хранятся номера телефонов, когда пользователь был создан, когда последний раз входил и его UID.



polyalarm-7b63a	users	9324388546
+ Start collection	+ Add document	+ Start collection
users >	9227809765	+ Add field
	9324388546 >	groupNumber: "3530202/90002"
		nickname: "de4nny"

Рисунок 12. База данных регистрации.

В базе данных регистрации у нас есть коллекция пользователь, где каждый пользователь — это отдельный документ, который называется по номеру телефона, который пользователь вводит на этапе верификации. В каждом документе хранится структура User.



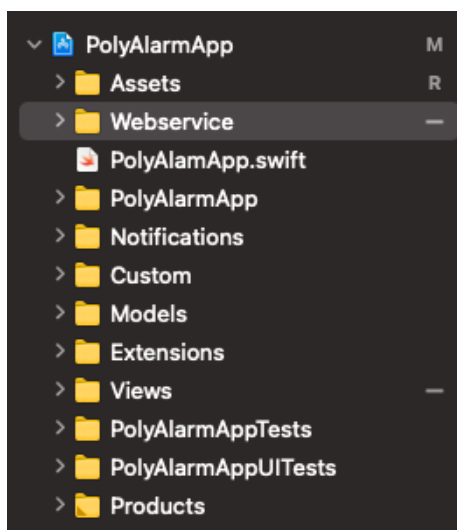
User
groupNumber: String
nickname: String

Рисунок 13. Структура User.

В этой структуре есть два строковых поля: номер группы и имя пользователя.

2.5. Проектирование компонентов

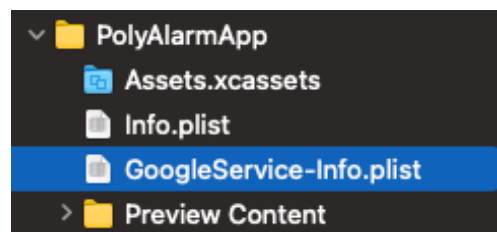
В приложении есть следующие основные пакеты, которые подразделяются на 9 категорий:



Я подробно расскажу о тех компонентах, с которыми работала, а с остальными вы можете ознакомиться в работе *Потаповой Алины*.

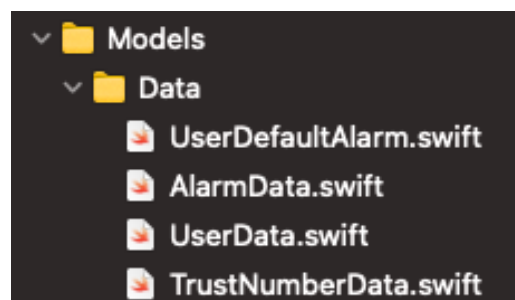
Блок PolyAlarmApp

В этом блоке содержатся списки различных программных свойств, например, BUNDLE_ID для работы с нашей базой данных Firebase.



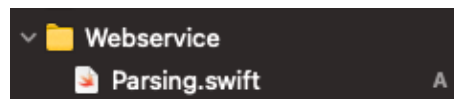
Блок Models

В данном блоке расположено перечисление типов и контейнеров. Я работала со структурами User (объект “Пользователь”) и TrustNumber (объект “Доверительный номер”).



Блок Webservice

В этом блоке один файл, в котором реализован алгоритм парсинга с сайта СПбПУ для того,

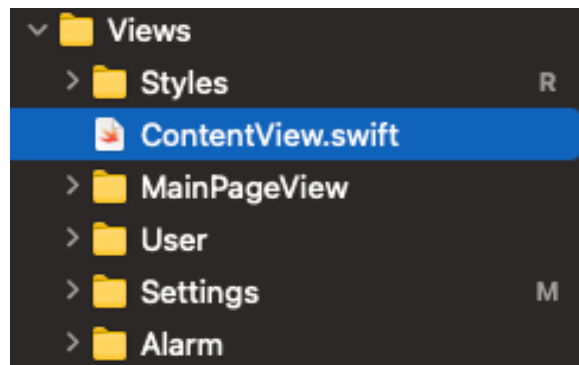


чтобы поставить первый будильник по расписанию, и отображать всю актуальную информацию о расписании на странице пользователя.

Блок Views

В блоке Views находятся реализации всех пользовательских интерфейсов, а также соединенная с ними логика.

При старте приложения окно отображает экземпляр ContentView, который определен в файле ContentView.swift.



Мной в этом файле была прописана логика верификации и регистрации.

Далее в папках MainPageView, User, Alarm, Settings лежат файлы, в которых прописана логика верификации, регистрации, отображения актуальной информации о расписании, установка первого будильника по расписанию, взятие доверительных номеров, отправка сообщения (отсутствует в финальной версии продукта, т. к. для реализации данной функции необходим аккаунт разработчика Apple).

2.6. Реализация компонентов

С реализацией компонентов верстки, локальных данных и т.д. вы можете ознакомиться в работе *Потаповой Алины*, а я расскажу про реализацию парсинга и работы с базой данных.

Парсинг

В основу моего алгоритма входит словарь из кодов институтов. Например, у нас есть номер группы “3530202/90002”. Первые две цифры — это код института. После этого я получаю полную ссылку для перехода на страницу института, так как изначально мы находимся на выборе института, а пользователь не вводит название. Коду института “35” соответствует часть URL страницы института - “95”. После конкатенации строк мы получаем URL института и попадаем на страницу со всеми группами института.

<https://ruz.spbstu.ru/faculty/95/groups>

После этого с помощью парсинга я получаю HTML-фрагмент ссылки с помощью того словаря. Фреймворк BeautifulSoup позволяет взять определенный фрагмент определенного объекта из кода HTML. В словаре по коду института можно также найти часть HTML-фрагмента, по которому ведется поиск.

```
<a class="groups-list__link" href="/faculty/95/groups/29860" data-reactid=".12wgha448hs.0.2.2.0.$1.1.$1.0">3530202/90002</a> = $0
```

Мы получаем ссылку (href), а ведем поиск по data-reactid. После этого мы также конкатенируем строки и получаем полный URL расписания группы. Дальше все совсем просто, ведь нам просто нужно взять время первой пары, ее название, количество всех пар (по количеству элементов в списке HTML).

База данных

Для работы с базой данных использовался фреймворк Firebase. Также для соединения логики с вью использовались протоколы, которые описаны в работе *Потаповой Алины*. Если приводить пример, то можно взять работу с парсингом, ведь перед тем, как парсить — нам нужно получить номер группы. Для этого в файле сначала подключается база данных, затем по текущей

аутентификации мы берем номер телефона (берется он не с базы, а именно с текущей сессии). Затем мы ищем документ, который называется по номеру телефону, который мы уже знаем, и берем данные из строки номера группы. Соответственно, проверяем на существование документа перед взятием.

3. Выбор стратегии тестирования и разработка тестов

3.1. Выбор стратегии тестирования

В нашем проекте был выбран метод тестирования “Черный ящик”. Это функциональное и нефункциональное тестирование без доступа к внутренней структуре компонентов системы. Подробнее вы можете ознакомиться в работе *Потаповой Алины*.

3.2. Тест план

С этим пунктом вы можете ознакомиться в работе *Потаповой Алины*.

Заключение

Этот семестр мне особенно запомнился, ведь нам удалось создать продукт, в котором заинтересованы не только мы, но и наши одноклассники и ребята с других групп. Мы уже пользуемся нашим приложением, и оно действительно упрощает нам жизнь. С момента оглашения идеи у всей команды горели глаза, чтобы реализовать это приложение. Я не пропустила не единого занятия. Где бы я не была, хоть в самолете, хоть в другом городе. Я скорее пыталась подключиться и рассказать о том, что нам удалось сделать, ведь это первый наш такой опыт, и я считаю, что мы не то, что справились, мы безумно довольны результатом. Помимо этого, я приобрела новые теоретические знания, опыт работы в команде и практический.

Я рада, что такой предмет существует и, что я была частью PolyAlarmTeam. Также хотелось бы выразить благодарность нашему преподавателю – Самочадину А. В. за позитивный настрой и прекрасное руководство. Безусловно огромное “спасибо” нашей команде за слаженную работу, а также университету за такую возможность.

Литература

1. Документация язык программирования Swift от компании Apple, URL:
<https://developer.apple.com/documentation/swift>
2. Документация Firebase от компании Google, URL:
<https://firebase.google.com/docs/engage>
3. Презентации к лекциям по дисциплине: “Программная инженерия”, URL:
<https://dl.spbstu.ru/course/view.php?id=2526>
4. Статьи на Habr (iOS-разработка, построение архитектуры и т.д.), URL:
<https://habr.com/ru/>
5. Документация по фреймворку SwiftSoup, URL:
<https://github.com/scinfu/SwiftSoup>

Приложение А. Руководство пользователя

Руководство пользователя в свободной форме вы можете найти в работе *Потаповой Алины*.

Приложение Б. Coding standard

Используемый нашей командой стандарт вы можете посмотреть в работе *Потаповой Алины*.

Приложение В. Фрагменты текстов программ

Parsing.swift

```
func gettingURL() {
    gettingGroupNumber()
    let facultyFromGroup = groupNumber.prefix(2)
    let groupURL = faculties[String(facultyFromGroup)]!
    let facultyURL = "https://ruz.spbstu.ru/faculty/\(String(describing:
groupURL))/groups"
    let url = URL(string: facultyURL)
    let request = NSMutableURLRequest(url: url!)
    let task = URLSession.shared.dataTask(with: request as URLRequest) { data,
response, error in
        if error != nil {
            print(error!)
        } else {
            if let unwrappedData = data {
                let dataString = NSString(data: unwrappedData, encoding:
String.Encoding.utf8.rawValue)
                print(dataString!)
            }
        }
    }
    task.resume()
}
```

OTPView.swift

```
func checkUserStatus() {
    let docStatus = db.collection("users").document(phoneNumberDB)
    docStatus.getDocument { (document, error) in
        if let document = document, !document.exists {
            self.showRegistrationView.toggle()
        } else {
            self.verifyUser()
        }
    }
}
```

VerificationView.swift

```
NavigationLink(destination: OTPView(show: $show, ID: $ID, phoneNumberDB:
$phoneNumberField), isActive: $show) {
    Button(action: {
        UIApplication.shared.endEditing()
        self.loadingVerify.toggle()

PhoneAuthProvider.provider().verifyPhoneNumber("+" + self.codeField +
self.phoneNumberField, uiDelegate: nil) {
        (ID, err) in
        if err != nil {
            self.msg =
(err?.localizedDescription)!

            self.alert.toggle()
            return
        }
        self.ID = ID!
        self.show.toggle()
        self.loadingVerify.toggle()
    }) {
        RoundedRectangle(cornerRadius: 15)
            .frame(width: 125, height: 45)
            .shadow(radius: 0.2)
            .padding(10)
            .foregroundColor(.darkBlue)
            .overlay(
                Text(Strings.letsGo)
                    .simpleStyle()
            )
    }
}

}.alert(isPresented: $alert, content: {
    Alert(title: Text(Strings.errorMessage), message:
Text(self.msg), dismissButton: .default(Text("Ok")))
}))
})
}.frame(width: 320, height: 240, alignment: .center)
```

RegistrationView.swift

```
private func saveUser(groupnumber: String, nickname: String) {
    let docRef = db.collection("users").document(phoneNumberDoc)
    docRef.setData([
        "groupNumber": groupnumber,
        "nickname": nickname
    ])
}
```


Приложение Г. Отчет о code review

Весь контроль версий мы осуществляли через GitHub. Работали через 2 ветки, соответственно, резервная моя и Алины. Мы ничего не помещали в code review, потому что каждый занимался своей частью работы. Например, Ренат и Давид занимались двумя папками на GitHub – это Docs и Design. Мы с Алиной работали с папкой PolyAlarmApp. Владелец репозитория являюсь я, и каждый раз перед очередным merge request мы либо созванивались, либо лично виделись с Алиной и обсуждали детали того или иного изменения, поэтому каждый merge request либо закрывался из-за найденных ошибок или улучшений, либо пушился мной.

0 Open ✓ 13 Closed		Author ▾	Label ▾	Projects ▾	Milestones ▾	Reviews ▾	Assignee ▾	Sort ▾
■	🔗	Fix notifications #13 by pamapp was merged 14 days ago						
■	🔗	Registration, notifications, TrustNumberData etc. #12 by pamapp was merged 29 days ago						
■	🔗	A lot of stuff 2.0 #11 by pamapp was merged on 18 Apr						
■	🔗	A lot of stuff #10 by pamapp was merged on 13 Apr						
■	🔗	Edit verify views, add sign out (logic, view) and modify views #9 by pamapp was merged on 12 Apr						
■	🔗	Add bottom sheet, custom TabView Bar, settings view, verification etc #8 by pamapp was merged on 10 Apr						
■	🔗	working with firebase (adding pod file, adding firebaseAuth url) #7 by sirsur was closed on 7 Apr						
■	🔗	Add missing files (LoginView, SignUpView) #6 by pamapp was merged on 6 Apr						
■	🔗	Add missing files #5 by pamapp was closed on 6 Apr						
■	🔗	Add main page, signUp, logIn views #4 by pamapp was merged on 6 Apr						
■	🔗	Upload Xcode Project #3 by pamapp was merged on 5 Apr						
■	🔗	Delete Xcode Project #2 by pamapp was merged on 5 Apr						
■	🔗	Upload Xcode Project #1 by pamapp was merged on 5 Apr						

Приложение Д. Описание тестирования модуля по методу базового пути

С данным приложением вы можете ознакомиться в работе *Потаповой Алины*.