

Кафедра инженерной кибернетики

Квалификация (степень): магистр

Направление подготовки: 09.04.03 «Прикладная информатика»

Профиль (программа) «Иновационные ИТ проекты»

Дисциплина: Современные инструментальные средства разработки

КУРСОВАЯ РАБОТА

на тему

«Проектирование и разработка системы для организации
процесса обмена информацией между сотрудниками»

Выполнили:

Системный аналитик: Шидакова Альбина

Аналитик-дизайнер: Прохорова Ольга

Разработчик серверной части: Шохрин Игорь

Разработчик клиентского приложения: Умников Никита

Группа: МПИ-18-2-2

Проверил: Тарханов И.А.

Оценка: _____

Дата: _____

Москва 2019

Оглавление

1 . Общие сведения.....	3
1.1 Полное наименование системы и ее условное обозначение	3
1.2 Наименование объединений разработчика	3
1.3 Цели создания системы	3
1.4 Задачи проекта.....	3
2 . Требования к функциям (задачам), выполняемым системой.....	4
3 Требования к системе в целом.....	6
3.1 Требования к персоналу системы.....	6
3.2 Требования к надежности	6
3.3 Требования к производительности	6
3.4 Требования к эргономике и технической эстетике	6
3.5 Требования к лингвистическому обеспечению	7
3.6 Требования к контролю, хранению, обновлению и восстановлению данных .	8
3.7 Требования к программному обеспечению.....	8
3.8 Требования к техническому обеспечению	8
4 . Сценарий использования.....	9
5 .Сценарии тестирования	16
6 .Прототип основных окон системы.....	18
7 . Перечень принятых сокращений	24
8 . Листинг кода клиентской части.....	26
9 . Листинг кода серверной части.....	28
10 . Основные API команды сервера.....	31
11 . Список литературы	34

1 . Общие сведения

1.1 Полное наименование системы и ее условное обозначение

Полное наименование системы: MISIS-Chat-Project

Условное обозначение: Чат, Система.

1.2 Наименование объединений разработчика

ИСПОЛНИТЕЛЬ: Группа МПИ 18-2-2 Прохорова Ольга, Умников Никита, Шидакова Альбина, Шохрин Игорь.

1.3 Цели создания системы

Целью создания Системы является автоматизация процесса обмена информацией между сотрудниками компании.

1.4 Задачи проекта

- 1) Разработка технического задания и проектирования диаграммы сценариев;
- 2) Моделирование прототипа системы;
- 3) Проектирование структуры базы данных;
- 4) Разработка серверной части системы (бэкенд)
- 5) Разработка клиентской части (фронтэнд)
- 6) Написание test case
- 7) Тестирование

2 . Требования к функциям (задачам), выполняемым системой

1) Регистрация

Доступ в чат должен быть доступен только по процедуре регистрации. Пользователю должна быть предоставлена специальная форма, в которой ему нужно будет ввести имя, фамилия, логин, пароль. Также должна быть альтернативная регистрация с помощью социальных сетей (ВК).

2) Авторизация

Для авторизации пользователю должна быть предоставлена специальная форма, в которой ему нужно будет ввести логин и пароль. При удачной авторизации пользователя, пользователь входит в чат. Если же пароль неверный, то необходимо уведомить пользователя об этом, и предложить вспомнить пароль.

3) Отправка сообщений

Пользователь должен иметь возможность написать и отправить сообщение. Поле написания сообщения содержит текстовое поле сообщения и кнопку «Отправить». После отправки сообщение должно появиться в поле чата.

4) Функция «Объявление»

Функция представляет собой вывод специального сообщения, которое будет выделено.

5) Статус пользователя.

Пользователь, находясь в чате, должен иметь возможность выбирать свой статус: онлайн, отошел, недоступен и так далее.

6) Мой профиль

Личный профиль пользователя. Пользователь заполняет поля. В личном профиле пользователи отображаются все данные, связанные с его анкетой.

7) Группы

Сервис, объединяющий пользователей по интересам. Представляет собой своего рода группу. Управление группами производится через панель управления или уполномоченным на то пользователем. Каждая группа имеет название,

описание и иконку, для отображения в списке пользователей. Пользователь может создать свою группу по интересам.

8) Написание собственных модулей с помощью API

Так как система имеет API(программный интерфейс приложения, интерфейс прикладного программирования— описание способов (набор классов, процедур, функций, структур или констант), которыми одна компьютерная программа может взаимодействовать с другой программой. Часто реализуется отдельной программной библиотекой или сервисом операционной системы. Используется программистами при написании всевозможных приложений.), пользователь, имеющий необходимые знания может доработать систему «под себя» или написать новые модули.

9) Просмотр календаря и погоды

Пользователь должен иметь возможность воспользоваться календарем, делать необходимые заметки на определенную дату в календаре, а также посмотреть погоду на сегодня и ближайшие дни.

3 Требования к системе в целом

3.1 Требования к персоналу системы

Для работы с системой от персонала не должно требоваться специальных технических навыков, знания технологий или программных продуктов, за исключением общих навыков работы с персональным компьютером и стандартным веб-браузером (например, MS Internet Explorer 7.0 или выше).

3.2 Требования к надежности

При возникновении сбоев в аппаратном обеспечении, включая аварийное отключение сервера, программный комплекс должен автоматически восстанавливать свою работоспособность после устранения сбоев.

Программный комплекс должен обеспечивать корректную обработку аварийных ситуаций, вызванных неверными действиями пользователей, неверным форматом или недопустимыми значениями входных данных. В указанных случаях система должна выдавать пользователю соответствующие аварийные сообщения, после чего возвращаться в рабочее состояние, предшествовавшее неверной (недопустимой) команде или некорректному вводу данных.

3.3 Требования к производительности

Время отклика системы на запрос пользователя не должно превышать 2 секунд.

Допустимое количество одновременно работающих вместе пользователей не должно быть более 1000 человек.

3.4 Требования к эргономике и технической эстетике

Взаимодействие пользователей с программным комплексом должно осуществляться посредством визуального графического интерфейса (GUI). Ввод-

вывод данных, прием управляющих команд и отображение результатов их исполнения должны выполняться в интерактивном режиме, в реальном масштабе времени. Интерфейс должен быть рассчитан на преимущественное использование «персональных компьютеров», т.е. управление системой должно осуществляется с помощью набора экранных меню, кнопок, значков и т.п. элементов. Клавиатурный режим ввода должен использоваться главным образом при заполнении / редактировании текстовых и числовых полей экранных форм.

3.5 Требования к лингвистическому обеспечению

При реализации серверной части системы должны применяться следующие языки высокого уровня и библиотеки:

- 1) python;
- 2) requests (версия: 2.22.0)
- 3) vk_api (версия: 2.0.2)
- 4) flask (версия: 1.0.2)
- 5) datetime (входит в стандартный пакет поставки python)
- 6) urllib3 (версия: 1.25.2)
- 7) sqlalchemy (версия: 1.3.3)
- 8) pypsocks (версия: 1.7.0)

При реализации клиентской части системы должны применяться следующие языки высокого уровня и библиотеки:

- 1) html;
- 2) javascript
- 3) css;
- 4) bootstrap 4.0;
- 5) font awesome 4.7;
- 6) jquery 3.2.1;
- 7) material design icon 2.2.0.

Для хранения данных должна быть использована SQL Server.

Пользователи должны взаимодействовать с системой на уровне графического пользовательского интерфейса.

Все функции системы должны поддерживать русский.

3.6 Требования к контролю, хранению, обновлению и восстановлению данных

К контролю данных предъявляются следующие требования:

- хранение исторических данных в системе должно производиться не более чем за 1 год. По истечению данного срока данные должны переходить в архив;
- исторические данные, превышающие двухлетний порог, должны быть удалены.

3.7 Требования к программному обеспечению

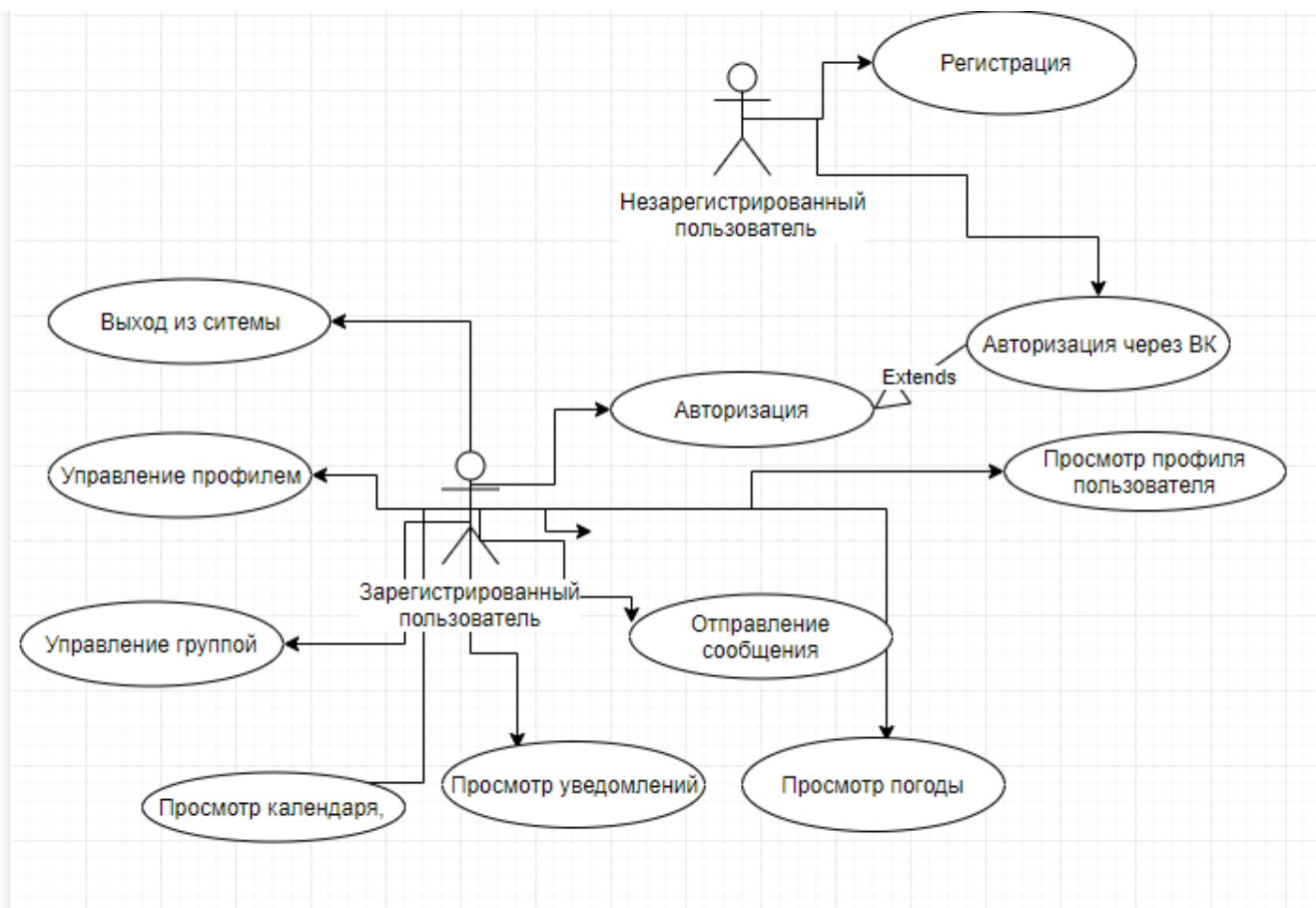
Техническое обеспечение рабочих станций:

- 1) Процессор, 2 ГГц или выше;
- 2) Оперативная память не менее 1024 Мб;
- 3) Видеокарта и монитор с разрешающей способностью – 1024x768, 1280x1024, 1600x1200, 1920x1440, 2048x1536;
- 4) Клавиатура;
- 5) Манипулятор «мышь»;

3.8 Требования к техническому обеспечению

Система должна быть рассчитана на функционирование в следующей программной среде: персональные компьютеры (преимущественно) и мобильные устройства;

4 . Сценарий использования



Регистрация

Название прецедента: регистрация нового участника чата

Действующее лицо: незарегистрированный пользователь

Цель: добавить участника в систему,

Предусловия: пользователь заходит на сайт чата

Главная последовательность:

1. пользователь выбирает в *появившемся* окне пункт «Регистрация»;
2. система показывает пользователю *окно регистрации*, содержащее поля для ввода имени, фамилии, e-mail, логина и пароля, а также кнопки «Регистрация»;

3. пользователь вводит имя, фамилию, *e-mail* и желаемый логин и пароль, нажимает кнопку «*Регистрация*»;
4. система добавляет нового участника;
5. пользователю открывается *главное меню* и выводится уведомление о том, что участник был добавлен успешно.

Альтернативная последовательность (добавление участника, уже имеющегося в системе):

1. пользователю отображается уведомление о том, что запрашиваемый логин занят.

Альтернативная последовательность (сбой в работе чата):

1. пользователю отображается уведомление о том, что произошел сбой (при этом данные, введенные в формы *окна регистрации* не сохраняются).

Альтернативная последовательность

1. пользователь выбирает в *появившемся окне* пункт «*Авторизация через ВК*»;
2. система показывает пользователю *окно авторизации ВК*, содержащее поля для ввода логина и пароля, а также кнопки «*Войти*»;
3. пользователь вводит логин и пароль от учетной записи ВК, нажимает кнопку «*Войти*»;
4. система открывает главную страницу чата;
5. пользователю выводится уведомление о том, что участник успешно авторизовался.

Альтернативная последовательность

- 1) пользователю отображается уведомление о том, что сервис ВК временно недоступен.

Авторизация

Название прецедента: авторизация участника чата

Действующее лицо: зарегистрированный пользователь

Цель: войти в систему

Предусловия: пользователь заходит на сайт чата

Главная последовательность:

- 1) пользователь выбирает в *появившемся окне* пункт «*Авторизация*»;
- 2) система показывает пользователю *окно авторизации*, содержащее поля для ввода логина и пароля, а также кнопки «*Войти*»;
- 3) пользователь вводит логин и пароль, нажимает кнопку «*Войти*»;
- 4) система открывает главную страницу чата;
- 5) пользователю выводится уведомление о том, что участник успешно авторизовался.

Альтернативная последовательность (ввод неверного логина/пароля):

- 1) система выводит уведомление об ошибочном логине/пароле и просит повторить ввод;

Альтернативная последовательность (сбой в работе чата):

- 1) пользователь заходит на сайт системы;
- 2) пользователю отображается уведомление о том, что произошел сбой.

Просмотр профиля пользователя

Название прецедента: просмотр профиля пользователей

Действующее лицо: зарегистрированный пользователь

Цель: ознакомиться с личными данными о пользователе

Предусловия: пользователь авторизовался в системе

Главная последовательность:

1. пользователь выбирает в *появившемся окне* пункт «*Контакты*»;
2. система показывает пользователю *список с контактами*;
3. пользователь может найти необходимого участника и нажать «аватарку» участника.
4. система открывает окно профиля, содержащее данные об участнике.

Отправления сообщения

Название прецедента: отправления сообщения

Действующее лицо: зарегистрированный пользователь

Цель: отправить сообщение пользователю

Предусловия: пользователь авторизовался в системе

Главная последовательность:

1. пользователь выбирает в левом меню окна необходимого участника и нажимает на данный пункт;
2. система открывает окно с диалогом двух (и/или более) пользователей;
3. пользователь в специальном поле вводит необходимое сообщение и нажимает на кнопку «Отправить»;
4. система отображает данное сообщение с указанием даты и времени в диалоге, для разных пользователей цвет сообщений разный;
5. пользователь-получатель видит отправленное ему письмо, также может на него ответить;

Альтернативная последовательность (сбой в работе чата)

1. система отображает уведомление о сбое (текст сообщения не сохраняется);

Управление профилем

Название прецедента: управление профилем

Действующее лицо: зарегистрированный пользователь

Цель: работа с профилем

Предусловия: пользователь авторизовался в системе

Главная последовательность:

- 1) пользователь выбирает в *появившемся окне* пункт «Профиль»;
- 2) система показывает окно с данными о пользователе;

Альтернативная последовательность (изменить профиль):

1. пользователь выбирает в *появившемся окне* пункт «Профиль»;
2. система показывает окно с данными о пользователе;
3. пользователь вносит изменения в соответствующие поля и нажимает кнопку «Сохранить»
4. система сохраняет изменения и отображает сообщение об успешном выполнении команды.

Альтернативная последовательность (изменить профиль с ошибкой):

- 1) система не сохраняет изменения и отображает сообщение об ошибке сохранения.

Управление группой

Название прецедента: управление группой (групповым чатом)

Действующее лицо: зарегистрированный пользователь

Цель: создать группу

Предусловия: пользователь зашел в систему

Главная последовательность:

- 1) пользователь нажимает на кнопку «Создать»;
- 2) система показывает отображает окно с полями для заполнения;
- 3) пользователь заполняет необходимые поля и нажимает на кнопку «Добавить участника»;
- 4) система открывает список со всеми пользователями;
- 5) пользователь выбирает необходимых участников и нажимает на кнопку «Добавить»;
- 6) система создает группу.

Альтернативная последовательность

1. Пользователь не заполнил необходимые поля

2. система отображает сообщение о необходимости заполнения обязательных полей.

Просмотр календаря

Название прецедента: просмотреть календарь

Действующее лицо: зарегистрированный пользователь

Цель: просмотреть календарь

Предусловия: авторизовался в системе

Главная последовательность:

- 1) пользователь выбирает в *появившемся окне* пункт «Календарь»;
- 2) система показывает окно с календарем;

Просмотр погоды

Название прецедента: просмотреть погоду

Действующее лицо: зарегистрированный пользователь

Цель: узнать прогноз погоды

Предусловия: авторизовался в системе

Главная последовательность:

- 1) пользователь выбирает в *появившемся окне* пункт «Погода»;
- 2) система показывает окно с погодой;
- 3)

Просмотр уведомлений

Название прецедента: просмотреть уведомление

Действующее лицо: зарегистрированный пользователь

Цель: прочитывать уведомление

Предусловия: пришло уведомление

Главная последовательность:

- 1) пользователь выбирает в *появившемся окне* пункт «Уведомление»;
- 2) система показывает окно с уведомлениями;

- 3) пользователь нажимает на пришедшее уведомление;
- 4) система открывает полное сообщение.

5 .Сценарии тестирования

№	Название	Описание	Результат
1	Регистрация с помощью ВК	Регистрация в системе через ВК Предварительные шаги: Зайти на сайт чата Шаги: 1. Выбрать способ регистрации «Регистрация ВК» 2. Нажать кнопку «Регистрация ВК». 3. Ввести логин и пароль от учетной записи ВК 4. И нажать кнопку «Войти»	Вводимое значение Ожидаемый результат Логин:TEST1 Пароль:TEST «Вы успешно зарегистрировались в системе» Логин:TEST Пароль:TEST Ошибка-«Неверный логин или пароль. Введите еще раз»
2	Регистрация вручную	Регистрация в системе вручную Предварительные шаги: Зайти на сайт чата Шаги: 1. Выбрать способ регистрации «Другое» 2. Нажать кнопку «Другое». 3. Ввести логин и пароль, адрес электронной почты 4. Нажать «Сохранить»	Вводимое значение Ожидаемый результат Логин:TEST1 Пароль:TEST «Вы успешно зарегистрировались в системе» Логин:TEST Пароль:TEST Ошибка-«Неверный логин или пароль. Введите еще раз»
3	Создание чата	Создание чата Предварительные шаги: зайти на сайт чата Шаги: 1.Перейти на вкладку «Диалоги» 2. Нажать кнопку со значком «+» 3. Появляется окно в котором необходимо выбрать людей, которые будут добавлены в чат 4. Затем необходимо нажать на кнопку «+» 5. При выборе нескольких человек появляется окно, в котором необходимо ввести название чата.	Вводимое значение Ожидаемый результат Создание диалога (чата)
4	Отображение календаря	Отображение календаря Предварительные шаги: Зайти на сайт чата Шаги: 1. Кликнуть на кнопку «Календарь» 2. Произойдет автоматическое отображение календаря.	Вводимое значение Ожидаемый результат Отображение календаря

5	Отображение погоды	<p>Отображение погоды</p> <p>Предварительные шаги:</p> <p>Зайти на сайт чата</p> <p>Шаги:</p> <ol style="list-style-type: none"> 1. Кликнуть на кнопку «Погода» 2. Произойдет автоматическое отображение погоды. 	<p>Вводимое значение</p> <p>Ожидаемый результат</p> <p>Отображение погоды</p>
6	Поиск чата	<p>Поиск чата</p> <p>Предварительные шаги;</p> <p>Зайти на сайт чата</p> <p>Шаги:</p> <ol style="list-style-type: none"> 1. Кликнуть на клавишу “Поиск”. 2. Ввести значение в ячейку для поиска. 3. Кликнуть на кнопку “Поиск” 4. Автоматически происходит вывод найденного значения, 	<p>Вводимое значение:</p> <p>“значение для поиска”</p> <p>Ожидаемый результат</p> <p>“Вывод полной достаточной информации”</p> <p>Ошибка “Поиск не дал результатов”</p>

6 .Прототип основных окон системы

Прототип системы создавался с помощью конструктора ninjamock.

Данный прототип можно посмотреть по ссылке <https://ninjamock.com/s/787RLDx>

В прототипе представлено следующие формы:

- Регистрация;
- Диалоги (чаты);
- Профиль;
- Редактирование профиля;
- Отображение погоды;
- Отображение календаря;
- Заполнение данных профиля;
- Уведомления;
- Создание диалога;
- Поиск;

The image shows a web browser window with a navigation bar at the top containing buttons: Главная, Поиск, Группы, Уведомления, Календарь, Погода, Администратор, and Выход. Below the navigation bar is a section titled "Создание диалога". This section contains several form elements: two "Text input" fields for "Введите название чата" and "Введите описание"; a "Выберите аватар" section with a "Файл для загрузки" button and a "Выберите..." button; a list of names (Андреев В, Славная Д, Восенцкий В) each with a "Да" or "Нет" radio button; and "Сохранить" and "Отмена" buttons at the bottom.

Рисунок 1 – Отображение формы создания диалога

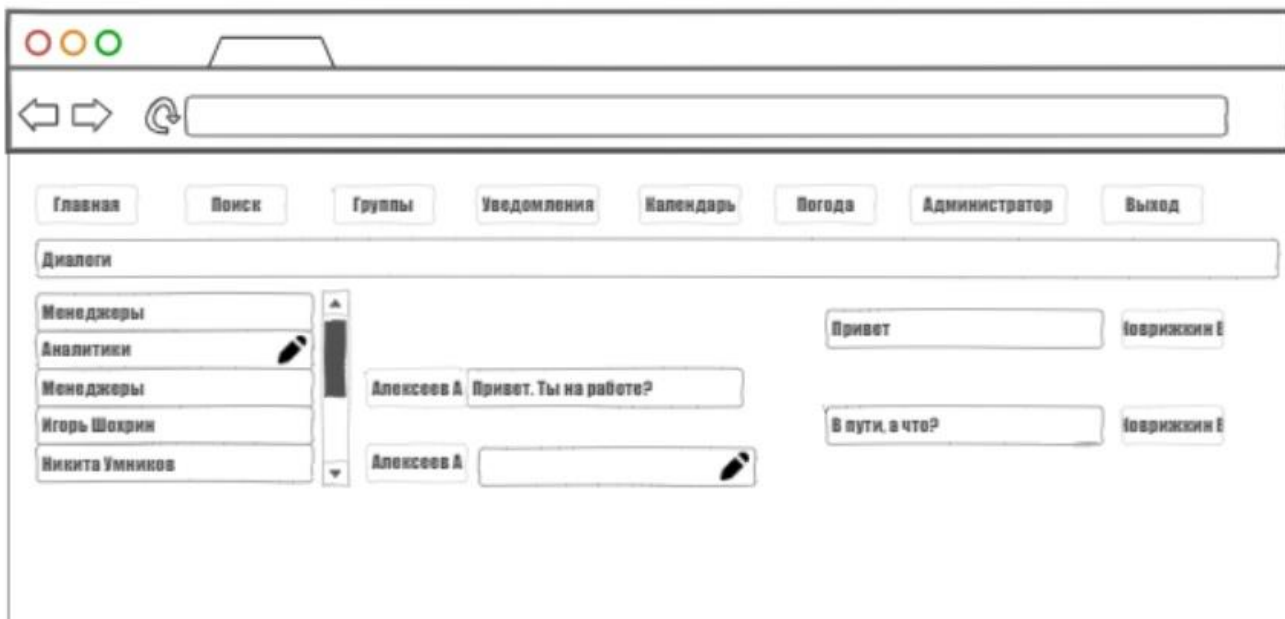


Рисунок 2 – Отображение формы диалогов (чатов)

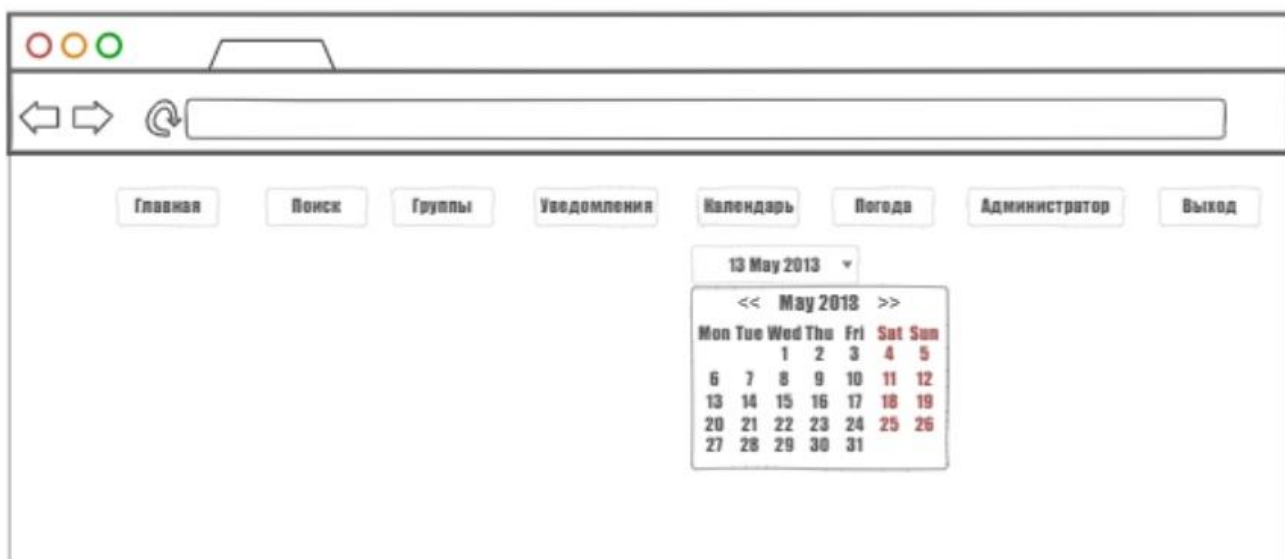


Рисунок 3 – Отображение формы календаря

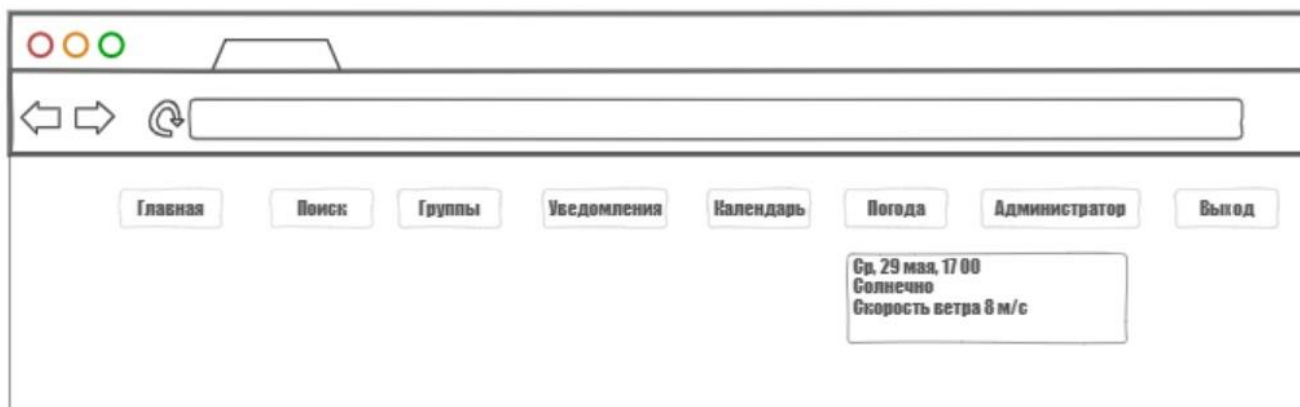


Рисунок 4 – Отображение погоды



Рисунок 5 – Отображение формы входа и регистрации

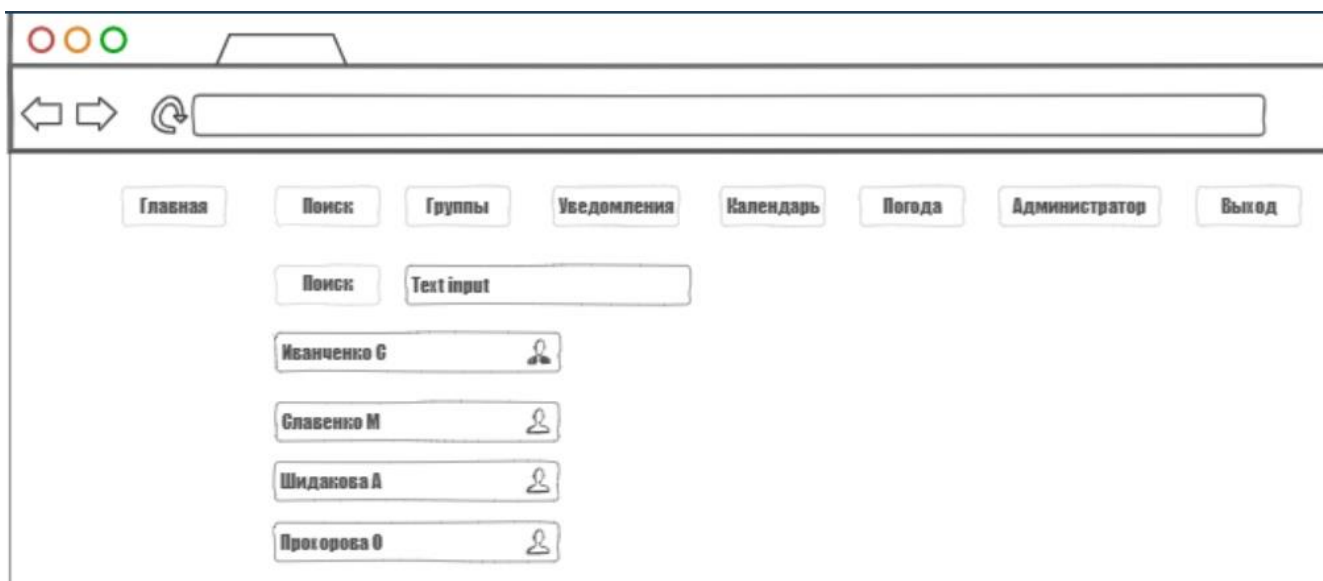


Рисунок 6 – Отображение формы поиска

Рисунок 7 – Отображение формы регистрации (ввод основной информации)

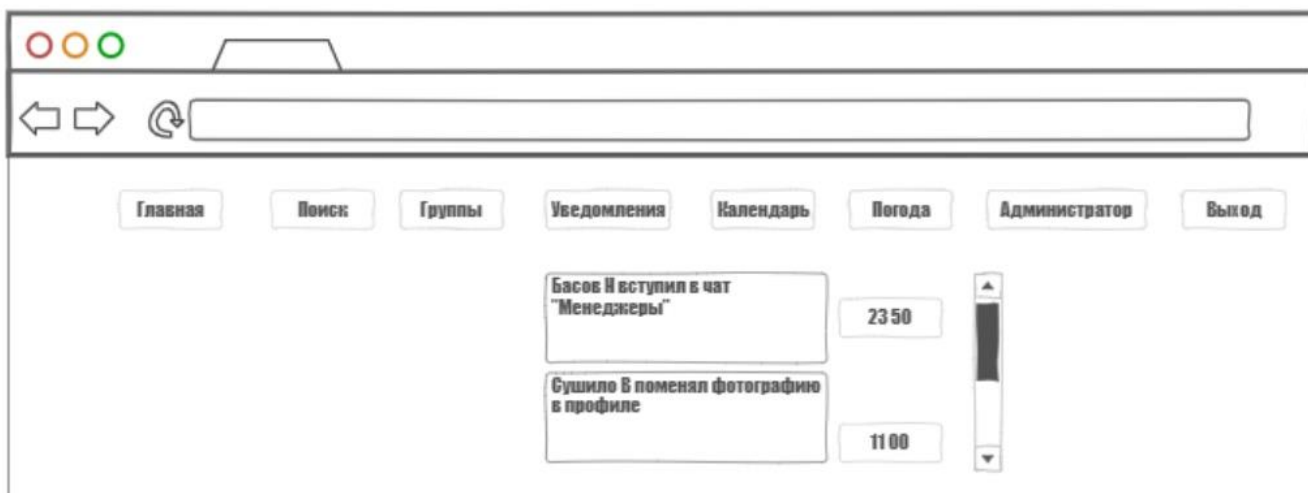


Рисунок 8 – Отображение формы уведомлений



Рисунок 9 – Отображение формы редактирования профиля



Рисунок 10 – Отображение главной страницы

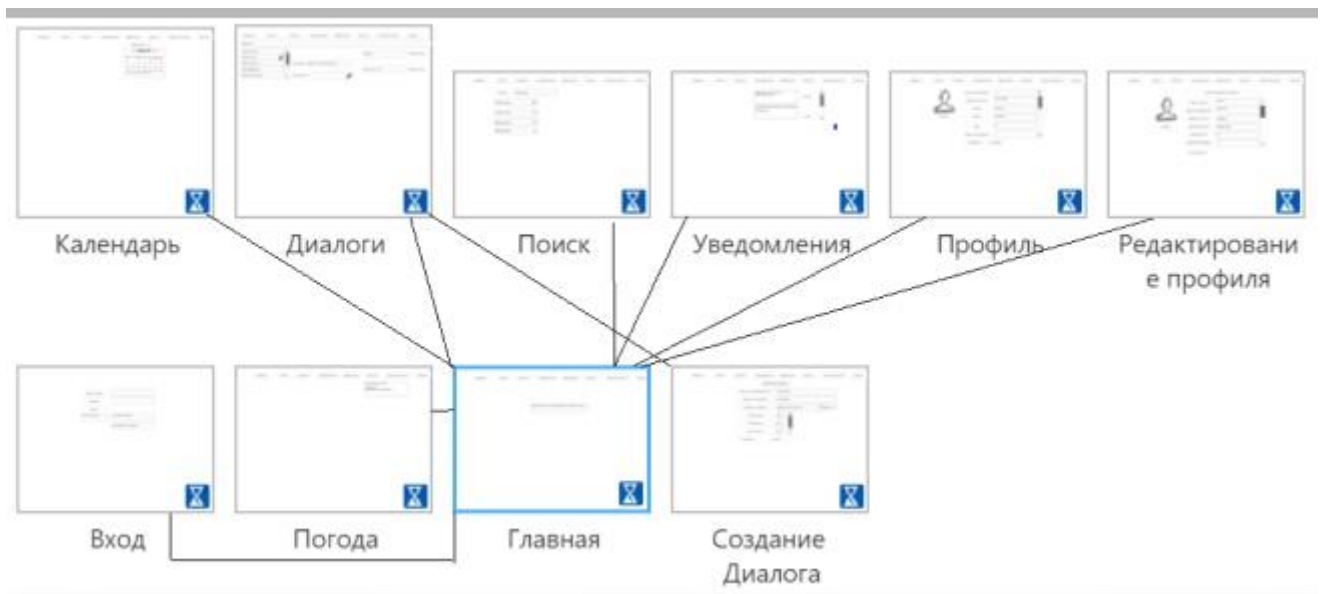


Рисунок 11 – Взаимодействие форм

Все формы данного прототипа представлены на <https://ninjamock.com/s/787RLDx>

7 . Перечень принятых сокращений

№	Обозначение/сокращение	Определение
1	ТЗ	Техническое задание
2	ВК	Социальная сеть Вконтакте
3	ГГц	Гигагерц
4	GUI	Графический интерфейс
5	API	Программный интерфейс приложения

8 . Листинг кода клиентской части

```
from flask import Flask, render_template, flash, redirect, request,
url_for,session,jsonify,json
from flask_login import LoginManager,UserMixin,current_user, login_user,logout_user
import os
import requests,datetime

app = Flask(__name__)
login = LoginManager(app)
app.secret_key = os.urandom(12)

from flask_login import UserMixin
all_user = {}
class User(UserMixin):#Класс пользователя

@login.user_loader
def load_user(user):#Получение информации о пользователе

@app.route('/main') #Главная страница
def main():

@app.route('/out') #Выйти из аккаунта
def out():
@app.route('/') #Самая первая страница
def home():

@app.route('/login', methods=['GET', 'POST']) #вход
def do_admin_login():

@app.route('/logvk', methods=['GET', 'POST']) #вход через вк
def logvk():

@app.route('/reg', methods=['GET', 'POST']) #регистрация
def reg():
@app.route('/regvk', methods=['GET', 'POST']) #регистрация через вк
def regvk():
@app.route('/search', methods=['GET', 'POST']) #поиск по сообщениям
def search():
@app.route('/groups') #диалоги
def groups():
```

```

@app.route('/friends', methods=['GET', 'POST']) #участики диалога
def friends():
    @app.route('/notify') #уведомления
    def notify():
        @app.route('/profile', methods=['GET', 'POST'] ) #профиль
        def profile():
            @app.route('/calen') #календарь
            def calen():
                @app.route('/get_len', methods=['GET', 'POST']) #получение сообщения
                def get_len():
                    @app.route('/sm', methods=['GET', 'POST']) #отправление сообщения
                    def sm():
                        @app.route('/dailm', methods=['GET', 'POST']) #смена диалога
                        def dailm():
                            @app.route('/changeprof', methods=['GET', 'POST'] ) #редактирование профиля
                            def changeprof():
                                @app.route('/pogoda', methods=['GET', 'POST'] ) #погода
                                def pogoda():
                                    @app.route('/createdial') #создание диалога
                                    def createdial():
                                        @app.route('/create', methods=['GET', 'POST']) #вспомогательная функция для
                                        диалога
                                        def create():
                                            @app.route('/sendprof', methods=['GET', 'POST']) # вспомогательная функция для
                                            профиля
                                            def sendprof():
                                                if __name__ == '__main__':
                                                    app.run(host='192.168.31.116',port=5000,debug=True)

```

9 . Листинг кода серверной части

```
app = Flask(__name__)

@app.route('/for_test', methods=['POST'])
def for_test():
    input_json = request.form["test_name"]
    print('data from client:', input_json)
    print(input_json['cmd'])
    if (input_json['cmd'] == 'VK_Autorization'):
        try:
            a = kontakt.get_user_photo(input_json['username'],
input_json['password'])
            except vk.exceptions.VkAuthError:
                return render_template('for_test.html', {'answer': 'Incorrect username or
password'})
            print("asdsad", a)
            # return render_template('for_test.html', {'answer': 'Authorization OK'})
            return render_template('for_test.html', {'answer': a})
            #return render_template('for_test.html', test_value=test_value)

@app.route('/')
def home():
    return render_template('for_test.html')
    def update_user_info(table, conn, input_json):
        dct = input_json.copy()
        # map(dct.pop, ["id_user", "cmd"])
        del dct["id_user"]
        del dct["cmd"]
        # conn.execute(sqlalchemy.update(table).where(table.c.id_user ==
input_json["id_user"]).values(dct))
        conn.execute(sqlalchemy.update(table).where(table.c.id_user ==
input_json["id_user"]).values(dct))

    def get_msg_from_dialog(table, table_user, conn, input_json):
        lst = []
        d = conn.execute(sqlalchemy.select([table]), autoincrement=True)
        m = []
        for i in d:
            if(i["id_dialog"] == input_json["id_dialog"]):
                # ans = get_user_info(table_user, conn, {"id_user" : i["id_user"]})
```

```

        # lst.append({"id_user" : i["id_user"], "first_name" : ans["first_name"],
"last_name" : ans["last_name"], "photo" : ans["photo"], "msg" : i["text"]})
        lst.append({"id_user": i["id_user"], "msg": i["text"], "time" : i["time"]})
    return lst

def get_sex(id):
    if(id == 1): return "Female"
    if(id == 2): return "Male"
    if(id == 0): return "None"

def get_user_from_dialog(table, conn, input_json):
    d = conn.execute(sqlalchemy.select([table]), autoincrement=True)
    m = []
    for i in d:
        if(i["id_dialog"] == input_json["id_dialog"]):
            m.append(i["id_user"])
    if(len(m) > 0):
        return {"ans": m}
    return {"ans": "This dialog not found"}

def get_user_info(table, conn, input_json):
    d = conn.execute(sqlalchemy.select([table]), autoincrement=True)
    m = []
    s = {}
    for i in d:
        if(i["id_user"] == input_json["id_user"]):
            for k in range(len(d.keys())):
                if(d.keys()[k] != "pas"):
                    s[d.keys()[k]] = str(i[k])
            return s
    return {"ans": "User with this id not find"}

def registration(table, conn, input_json):
    d = conn.execute(sqlalchemy.select([table]), autoincrement=True)
    for i in d:
        if (i["login"] == input_json["login"]):
            return {"ans": "Username is taken"}
    conn.execute(table.insert().values(pas = str(input_json["pas"]), login =
str(input_json["login"])))
    ans = login(table, conn, input_json)
    return ans

```

```

def login(table, conn, input_json):
    d = conn.execute(sqlalchemy.select([table]), autoincrement=True)
    for i in d:
        if(i["login"] == input_json["login"]):
            if(i["pas"] == input_json["pas"]):
                return {"ans" : "Authorization success", "id" : i["id_user"]}
            else:
                return {"ans" : "Incorrect login or password"}
    return {"ans": "Incorrect login or password"}

```

10 . Основные API команды сервера

Поддерживаемые команды:

VK_Authorization - авторизация пользователя через ВК:

Атрибуты:

username - Имя пользователя в ВК

password - пароль в ВК.

Ответ:

```
{"ответ": "неверное имя пользователя или пароль" / " авторизация ОК"}
```

Login - авторизация в чате:

Атрибуты:

pas - пароль

login - Имя пользователя (оно уникально для каждого пользователя)

Ответ:

```
{"ans": "авторизация прошла успешно" / "неверный логин или пароль", "id":
```

```
I ["id_user"]}
```

Registration - регистрация в чате (заполнение анкеты):

Атрибуты:

pas - пароль

login - Имя пользователя (оно уникально для каждого пользователя)

Ответ:

```
{"ans": "Username is taken"} или
```

```
{"ans": "авторизация прошла успешно", "id": i ["id_user"]}
```

get_msg - получить сообщение из бд:

Атрибуты:

Ответ:

```
{"Ошибка":"0"/"1", "msg": {...}}
```

send_msg - отправить сообщение

Аттрибуты:

text - Текст сообщения

id_dialog - ID диалога

id_user - ID пользователя

Ответ:

```
{"ans": "хорошо"}
```

create_new_user - создать нового пользователя (регистрация на сайте)

Аттрибуты:

first_name - Имя пользователя

last_name - Фамилия пользователя

login - уникальный(!) логин пользователя

pas - пароль пользователя

D_birth - Дата рождения

age - Возраст, в перспективе - вычисляется автоматически

sex - Пол

city - Город

photo - Фото

Ответ:

```
{"ans": "хорошо"}
```

create_new_dialog - создать новый диалог

Аттрибуты:

Name - имя диалога

create_date - Дата создания

capacity - Вместимость, в перспективе автоматическое вычисление

get_users - получить список пользователей с контактной информацией

Аттрибуты:

first_name - Имя пользователя

last_name - Фамилия пользователя

D_birth - Дата рождения

age - Возраст, в перспективе - вычисляется автоматически

sex - Пол

city - Город

Ответ:

```
{"ans" : {Все вышеперечисленные поля в формате "ключ - значение"}}
```

get_dialogs - получить список диалогов,

Атрибуты:

Name - имя диалога

create_date - Дата создания

capacity - Вместимость, в перспективе автоматическое вычисление

get_user_from_dialog - получить список пользователей, состоящих в диалоге

get_user_info

Атрибуты:

id_user - ID пользователя

Ответ:

```
{"ans" : {Все поля пользователя в формате "ключ - значение"}}
```

11 . Список литературы

I. Законодательные и нормативные акты

1. ГОСТ 19.201-78. «Единая система программной документации. Техническое задание. Требования к содержанию и оформлению»;
2. ГОСТ 34.602-89. Техническое задание на создание автоматизированной системы;

II. Монографии, учебники, учебные пособия

3. Бабич А.В UML: Первое знакомство/ - М.: Национальный Открытый Университет "ИНТУИТ", 2016.-209 с
4. Храмцов П.Б., Брик С.А., Русак А.М., Сурин А.И Основы web-технологий /. - М.: Национальный Открытый Университет "ИНТУИТ", 2016.- 166

1) Медведев А.В. Программа производственной (в том числе и преддипломной) практики для студентов, обучающихся по направлению 09.03.03 «Прикладная информатика» (программа подготовки бакалавра). – М.: Финансовый университет, кафедра «Информационные технологии», 2015. – 31 с.;

III. Статьи из периодической печати

5. UML-Моделирование предметной области при проектировании информационной системы салона красоты / Журба А.К., Продан Е.А., Сушкова М.С. //Инновационная наука. -2015. -№ 7-8. -С. 52-55

IV. Иностранная литература

6. Ларман К. Применение UML и шаблонов проектирования/пер. с англ. Шелестова А. Ю.. - М.: Вильямс, 2017. - 727 с

V. Интернет-ресурсы

7. Информационный портал «Научная электронная библиотека», 2014-2018, Режим доступа - <http://www.book.ru>.(Дата обращения: 19.05.2019).

8. Информационный портал «Научная электронная библиотека», 2014-2018 Режим доступа -<http://www.elibrary.ru>. (Дата обращения: 19.05.2019).
9. Информационный портал «Научная электронная библиотека», 2014-2018, Режим доступа - <http://www.znaniium.ru>. (Дата обращения: 19.05.2019).