Министерство образования и науки Российской Федерации

Иркутский государственный университет

Институт математики и информационных технологий

**Техническое задание**

На тему: «Проектирование проекта».

Выполнили: Студенты группы 2341

Козлов Алексей, Катышев Артём, Циберман Николай.

Оглавление

[1. Общая информация 3](#_Toc37599447)

[1.1. Назначение. 3](#_Toc37599448)

[1.2. Цель создания. 3](#_Toc37599449)

[1.3. Функциональные требования. 3](#_Toc37599450)

[1.4. Нефункциональные требования 3](#_Toc37599451)

[2. Диаграммы UML 4](#_Toc37599452)

[2.1. Диаграмма прецедентов 4](#_Toc37599453)

[2.2. Диаграмма активности 5](#_Toc37599454)

[2.3. Диаграмма последовательности 6](#_Toc37599455)

[2.4. Диаграмма компонентов 7](#_Toc37599456)

[2.5. Диаграмма классов 8](#_Toc37599457)

[2.6. Диаграмма состояний 9](#_Toc37599458)

[2.7. Диаграмма развёртывания 10](#_Toc37599459)

[3. Проектирование базы данных 11](#_Toc37599460)

[3.1. ER-диаграмма 11](#_Toc37599461)

[4. Выбор языка программирования, формата приложения, реализация БД и т.д. 12](#_Toc37599462)

[5. Разработка дизайна веб-сайта 12](#_Toc37599463)

# ****1. Общая информация****

## 1.1. Назначение.

Платформа предназначена для просмотра расписания занятий и домашнего задания в сети интернет.

## 1.2. Цель создания.

Целью создания платформы является предоставление студентам доступа к просмотру расписания и домашнего задания .

## 1.3. Функциональные требования.

На этапе анализа и разработки требований проекта были выявлены следующие функциональные требования:

1. Регистрация пользователя в информационной системе
2. Авторизация пользователя в информационной системе
3. Присоединение учащихся к группе
4. Добавления расписания
5. Добавления дз
6. Добавление дедлайнов

## 1.4. Нефункциональные требования

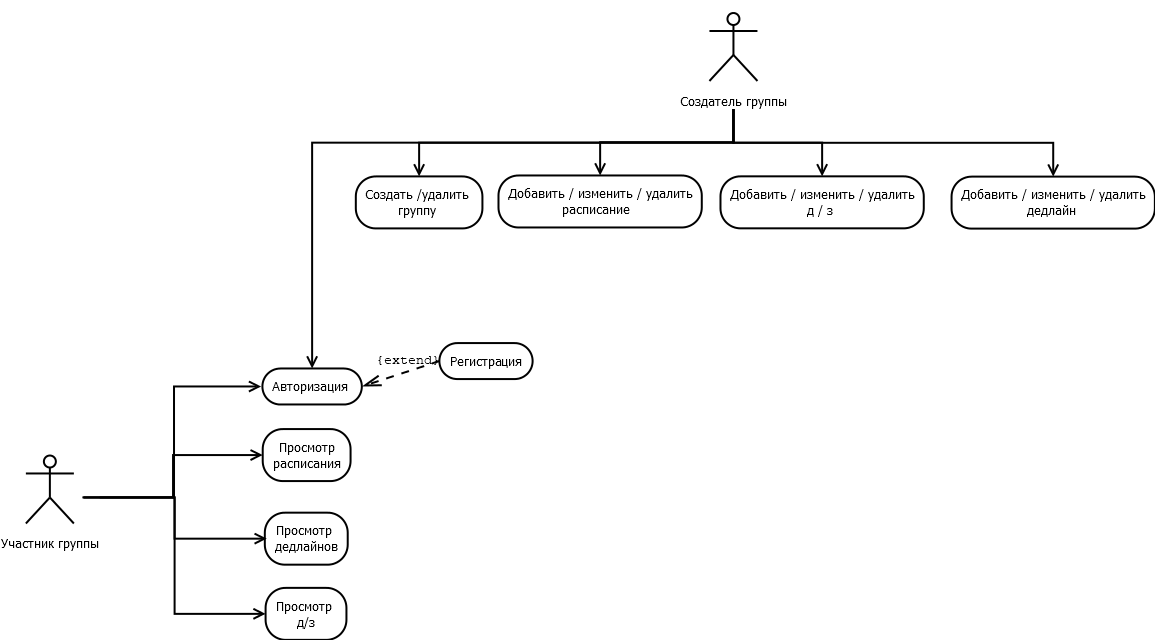
На этапе анализа и разработки требований проекта были выявлены следующие нефункциональные требования:

1. Система должна быть доступна для операционного использования в течение 24 часов.
2. Пользователю должны быть доступны все системные функции после пяти минут обучения работе с данной системой.
3. Время выполнения типичных задач от 2 до 10 минут.
4. Система должна добавлять карточки с домашним заданием в течении 5 секунд.
5. Система должна проверять достоверность пользователя в течение 5 секунд.

# 2. Диаграммы UML

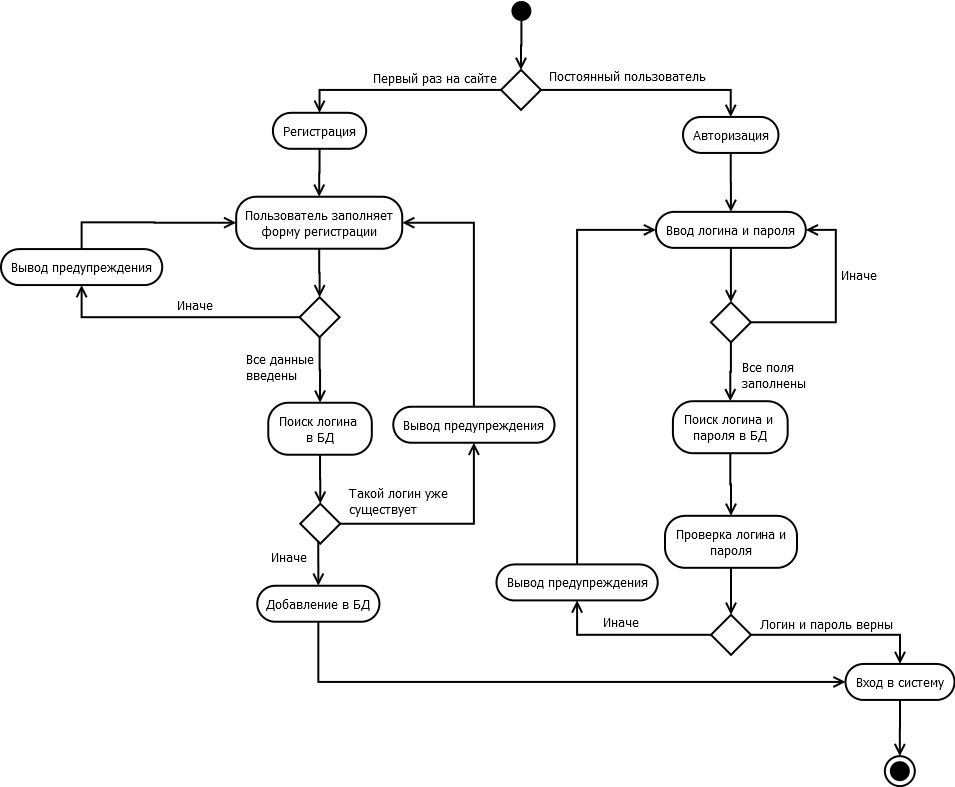
## 2.1. Диаграмма прецедентов

Данная диаграмма отображает взаимоотношения между акторами(стилизованные человечки) и прецедентами(эллипсы с надписью).



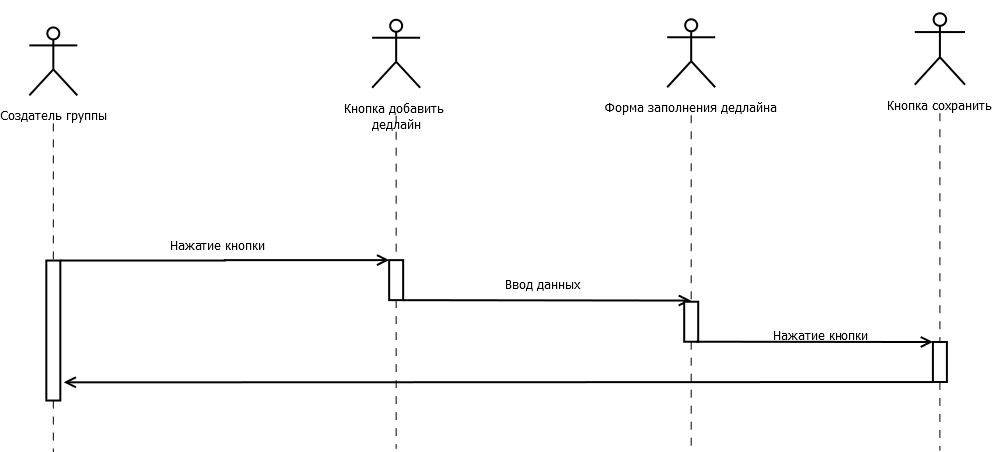
## 2.2. Диаграмма активности

Диаграмма активности, отражает динамические аспекты поведения системы. По существу, эта диаграмма представляет собой блок-схему. Ниже на рисунке представлен алгоритм регистрации и авторизации по которому будет работать наша система.



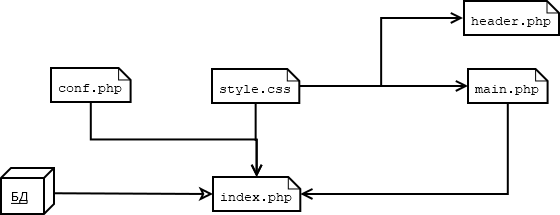
## 2.3. Диаграмма последовательности

Диаграмма последовательности — диаграмма, на которой для некоторого набора объектов на единой временной оси(пунктирная линия) показан жизненный цикл какого-либо определённого объекта и взаимодействие актеров ИС в рамках какого-либо определённого прецедента. С помощью диаграммы деятельности мы описываем последовательность действий для каждого прецедента, необходимая для достижения поставленной цели. Ниже на рисунке представлена диаграмма описывающая действие «добавить дедлайн» для прецедента «создатель группы».



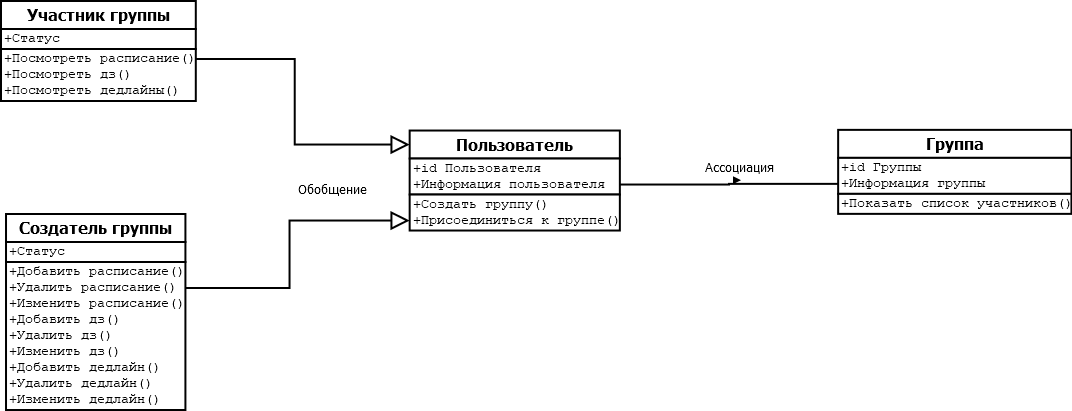
## 2.4. Диаграмма компонентов

Диаграмма компонентов - элемент языка моделирования UML, статическая структурная диаграмма, которая показывает разбиение программной системы на структурные компоненты и связи (зависимости) между компонентами. В качестве физических компонентов могут выступать файлы, библиотеки, модули, исполняемые файлы, пакеты и т. п. На рисунке ниже представлена диаграмма компонентов нашего проекта.



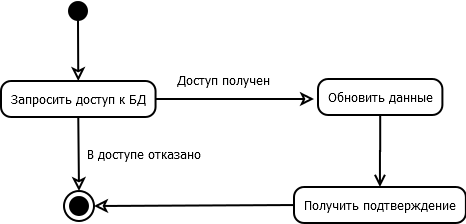
## 2.5. Диаграмма классов

Диаграмма классов определяет типы классов системы и различного рода статические связи, которые существуют между ними. На диаграммах классов изображаются также атрибуты классов, операции классов и ограничения, которые накладываются на связи между классами. На рисунке ниже представлена диаграмма классов нашего проекта.



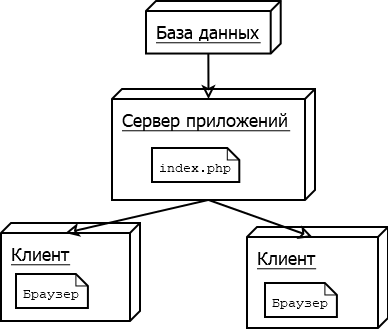
## 2.6. Диаграмма состояний

Диаграмма состояний показывает, как объект переходит из одного состояния в другое. Очевидно, что диаграммы состояний служат для моделирования динамических аспектов системы. Ниже на рисунке показан простой пример диаграммы состояний.



## 2.7. Диаграмма развёртывания

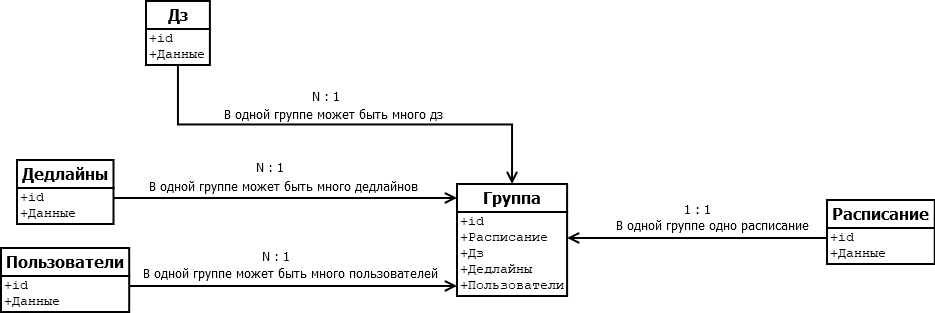
Диаграмма развёртывания моделирует физическое развертывание артефактов на узлах. Ниже на рисунке представлена диаграмма развёртывания нашего сайта.



# 3. Проектирование базы данных

## 3.1. ER-диаграмма

ER- диаграммы позволяют использовать наглядные графические обозначения для моделирования сущностей и их взаимосвязей. Ниже на рисунке представлена ER- диаграмма нашего проекта.



# 4. Выбор языка программирования, формата приложения, реализация БД и т.д.

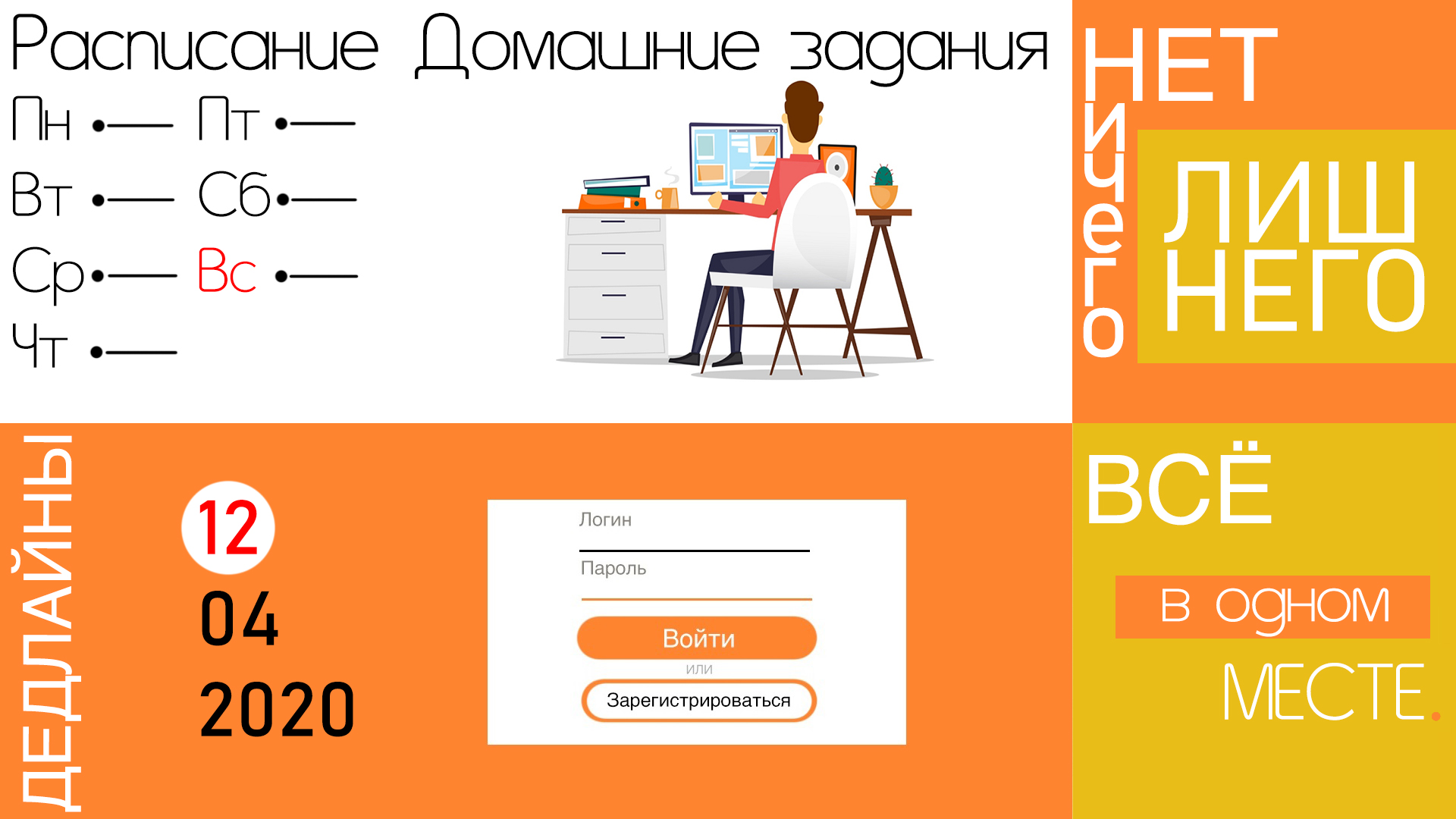
Наш продукт будет реализован в виде сайта.

* Серверная ос: Ubuntu 19.10 server
* Фрэймворк (пока под вопросом)
* Сервер: nginx
* Бэкэнд: MySql, phpMyAdmin, php7.2
* Фронтэнд: html, css, js

# 5. Разработка дизайна веб-сайта

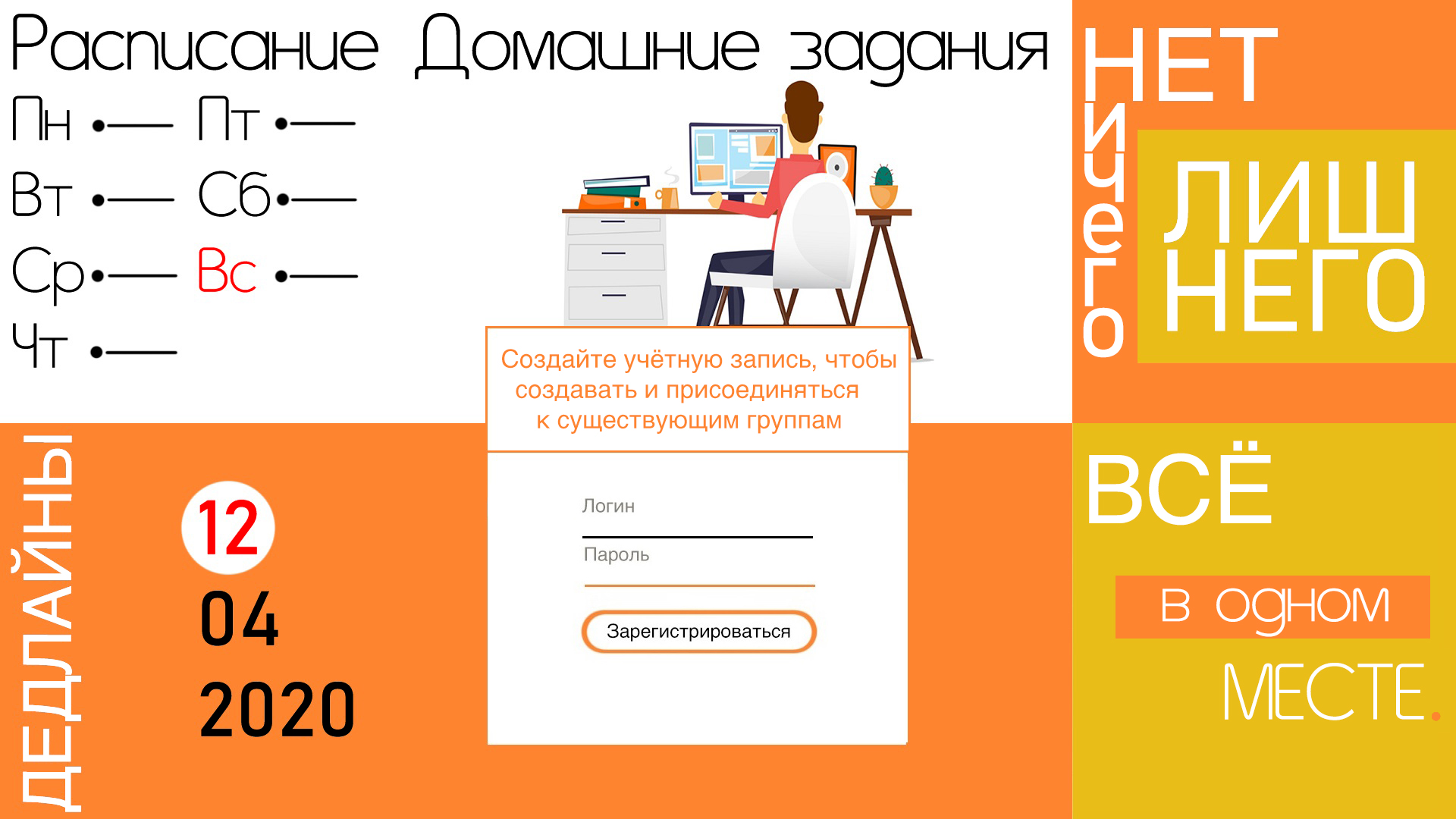
Для главной страницы нашего сайты мы использовали правило «пяти секунд», чтобы в течении пяти секунд пользователь понял, какие проблемы решает предлагаемый продукт.

Так предположительно будет выглядеть главная страница сайта:

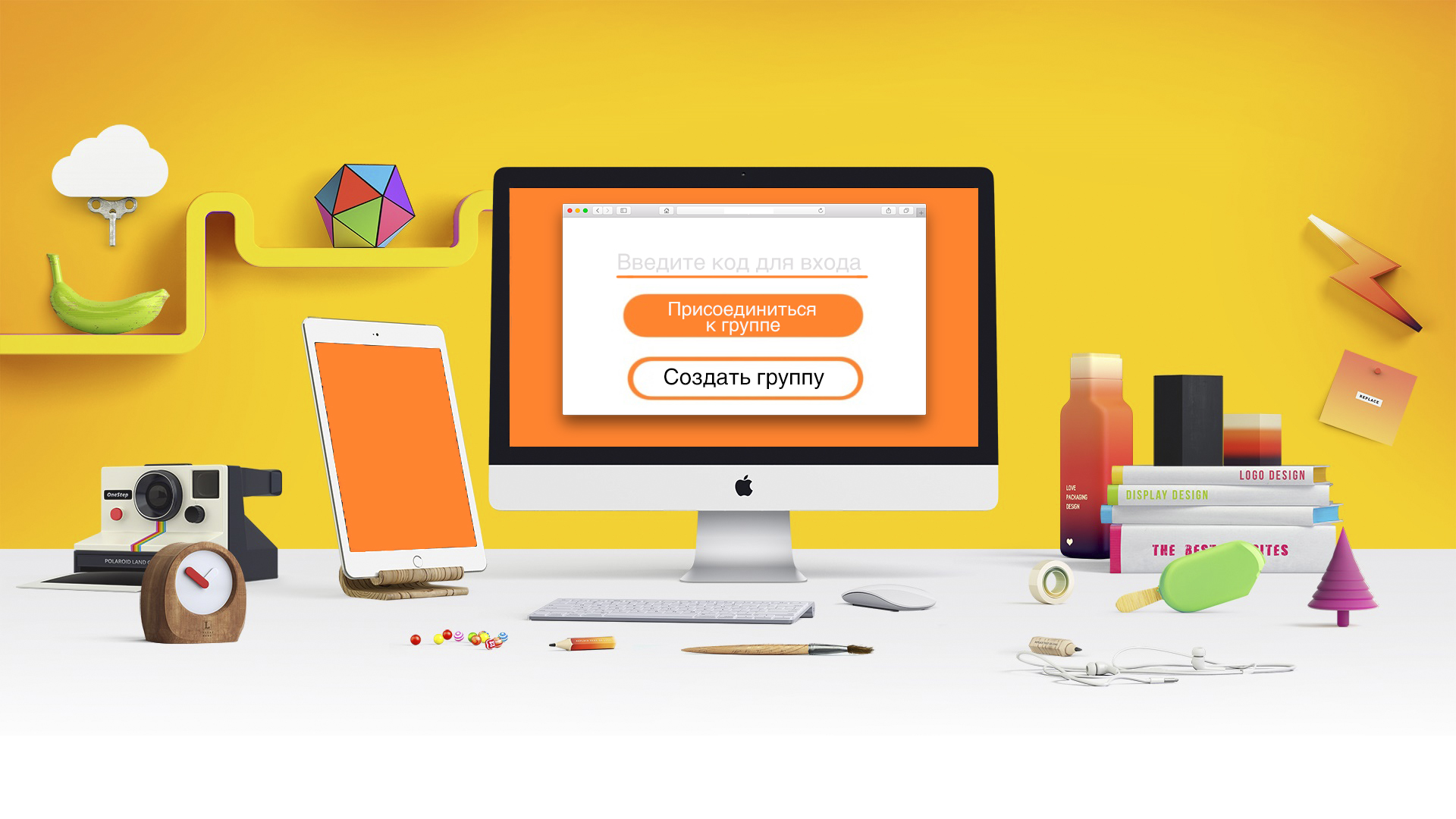


Поля «Логин» и «Пароль» находятся выше кнопки регистрации, что подразумевает, что ими будут пользоваться чаще. По этой же причине кнопка «Войти» имеет заливку и больше выделяется.

При нажатии кнопки “Зарегистрироваться” всплывает соответственное окно:

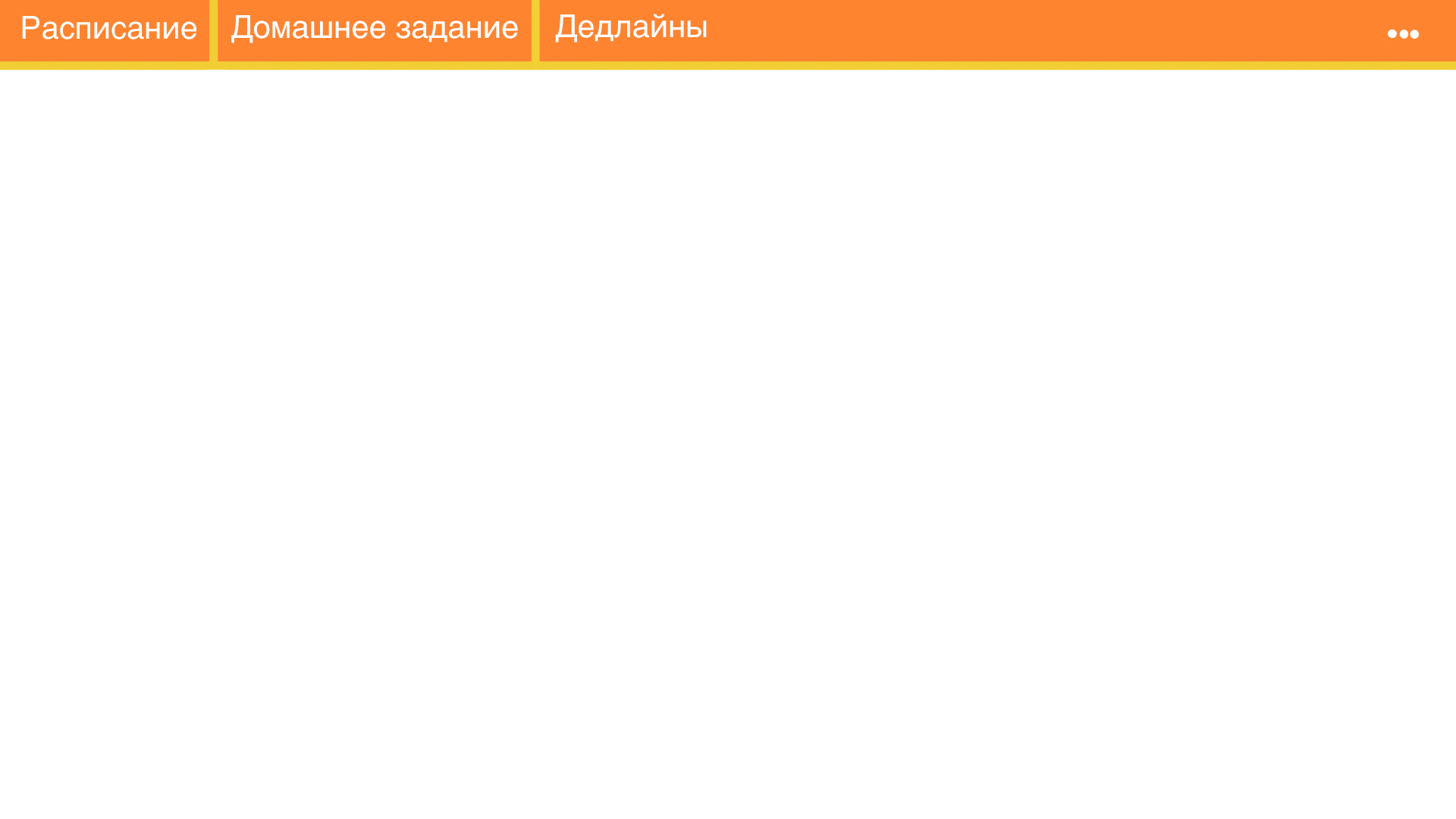


Далее независимо ввёл пользователь логин и пароль через кнопку “войти” или же зарегистрировался (не касается случая, если система запомнила пользователя), он попадает на следующую страницу:



Если пользователь присоединиться к группе, то он окажется в своей группе.

Так выглядит страница группы:



Пустота, но если пользователь выберет одну из кнопок, то она подсветиться и покажется соответствующая информация. Рассмотрим на примере кнопки

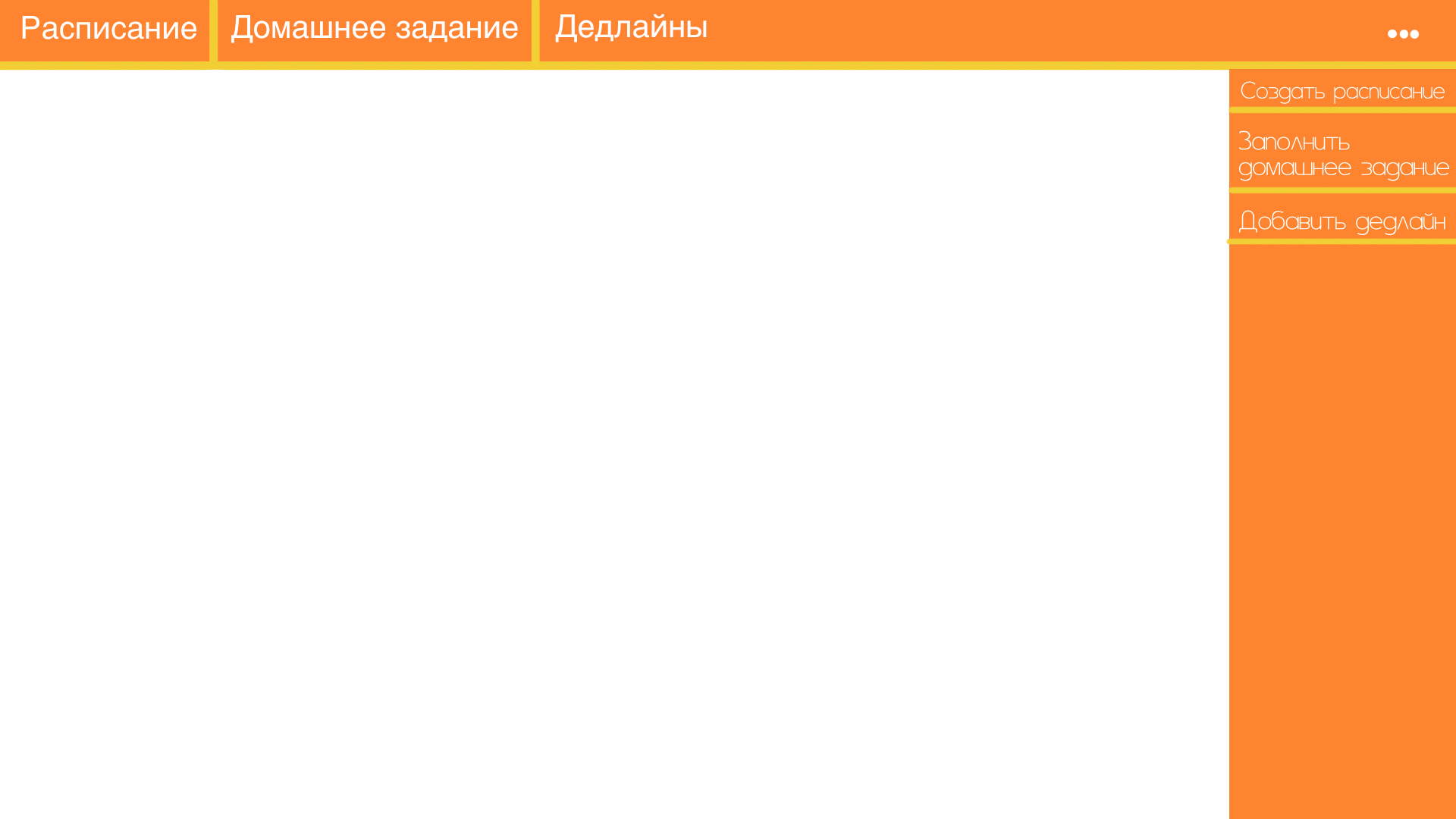
«Расписание»:



Это не конечный вариант, возможны изменения.

Добавлять информацию в поля “Расписание”, “Домашнее задание” и “Дедлайны” может только создатель группы с использованием кнопки “три точки” в правом верхнем углу.

При нажатии выдвигается соответствующее меню:



На счёт реализации создания самой группы пока не определились, т.к ещё до конца не определились со всеми возможностями «Создателя группы».

Цветовая схема:

|  |
| --- |
|  |