Балтийский государственный технический университет «Военмех» им. Д.Ф. Устинова

Кафедра И5

«Информационные системы и программная инженерия»

Разработка прототипа

дисциплина: «Прототипирование программного обеспечения»

тема: «Прототип мессенджера Cool»

Выполнил:

студент группы И974

Мигачев Илья Николаевич

Проверил:

Магомедов И.Н.

**СОДЕРЖАНИЕ**

[1 ПОСТАНОВКА ЗАДАЧИ 3](#_Toc59106067)

[2 МЕТОД ПРОТОТИПИРОВАНИЯ 4](#_Toc59106068)

[3 РАЗРАБОТКА ПРОТОТИПА 5](#_Toc59106069)

[4 ШАБЛОНЫ ПРОЕКТИРОВАНИЯ 12](#_Toc59106070)

[ВЫВОДЫ 13](#_Toc59106071)

[ЗАКЛЮЧЕНИЕ 14](#_Toc59106072)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 15](#_Toc59106073)

# **ПОСТАНОВКА ЗАДАЧИ**

Мессенджер – это клиентская программа, необходимая для мгновенного обмена сообщениями, которая представляет собой приложение, чаще всего устанавливаемое на смартфон или телефон пользователя. В большинстве случаев привязка (регистрация аккаунта) производится к номеру телефона. Сразу же после установки и регистрации аккаунта можно начинать общение с другими пользователями данного приложения.

Требуется разработать прототип приложения “Cool”.

Приложение должно обеспечивать выполнение перечисленных ниже возможности:

1. просмотр фотографий пользователей;
2. ведение диалога с пользователями;
3. добавление новых фотографий;
4. поиск фотографий по ключевым словам.

# **МЕТОД ПРОТОТИПИРОВАНИЯ**

Прототип – это схематичный набросок, эскиз страниц сайта или приложения с изображенными на нем элементами. К примеру, кнопки, «меню», формы и вариации. Прототипом можно назвать промежуточный и упрощенный вариант программной системы [1].

Прототипирование – процесс быстрого создания прототипа (эскиза, наброска), с целью создания базовой структуры и анализа будущего дизайна.

Существует несколько видов прототипирования. Выбранным методом прототипирования является эволюционное прототипирование. Оно предполагает последовательное увеличение достоверности исходного образца, пока, в конце концов, он не становится законченной системой. Эволюционное прототипирование — широко распространенный подход к разработке интерфейсов. Эта модель прототипирования может быть полезна для выявления больших тонкостей в аспектах дизайна и его совершенствования. На рисунке 1 представлена модель данного метода.



Рисунок 1 – модель эволюционного прототипирования

Данный вид прототипирования был выбран, потому что был разработан полноценный прототип приложения и при детальном его рассмотрении и анализе, выявлялись недочеты начального образца. С каждым новым циклом рассмотрения прототипа, недостатки были исправлены, а достоверность исходного образца повышалась.

# **РАЗРАБОТКА ПРОТОТИПА**

Для создания прототипа была использована платформа прототипирования Figma.

Figma — кросс-платформенный онлайн-сервис для дизайнеров интерфейсов и веб-разработчиков. Разработка интерфейсов происходит в онлайн-приложении [2]. Все файлы хранятся в облаке Figma. Не надо вспоминать, где лежат ваши макеты, скачивать и после того, как внесли изменения, загружать их обратно. Изменения в Figma сохраняются автоматически. Figma позволяет сделать кликабельную версию сайта или мобильного приложения, создавая множество экранов. Можно протестировать и настроить любую функцию: кнопки, всплывающие меню, анимированные и модальные окна.

Все экраны размещены на одной странице. Переключаться между окнами не нужно. Вы создаете связи и видите, как происходит работа над интерфейсом.

Для верстальщиков и программистов в Figma есть специальная панель Code, на которой можно скопировать CSS-стили для Web, а также код-разметки расположения и стилей для Android и iOS.

В Figma можно отслеживать историю действий каждого сотрудника, просматривать и восстанавливать предыдущие версии документа. Сервис автоматически сохраняет версии при закрытии вкладки проекта, или если пользователь не вносил изменения в последние 30 минут [3].

Главными преимуществами Figma является:

* позволяет совместно работать над документами — создавать и редактировать командой в режиме реального времени;
* хранит документы в облаке — макеты не занимают место на диске, их не нужно заливать, чтобы показать коллегам или заказчику;
* кроссплатформенность — возможность работать с редактором на Windows, Mac, Linux;
* возможность создавать интерактивные прототипы, векторные сетки, главные и зависимые компоненты.

Ниже на рисунке 2 представлена общая схема переходов в созданном прототипе.

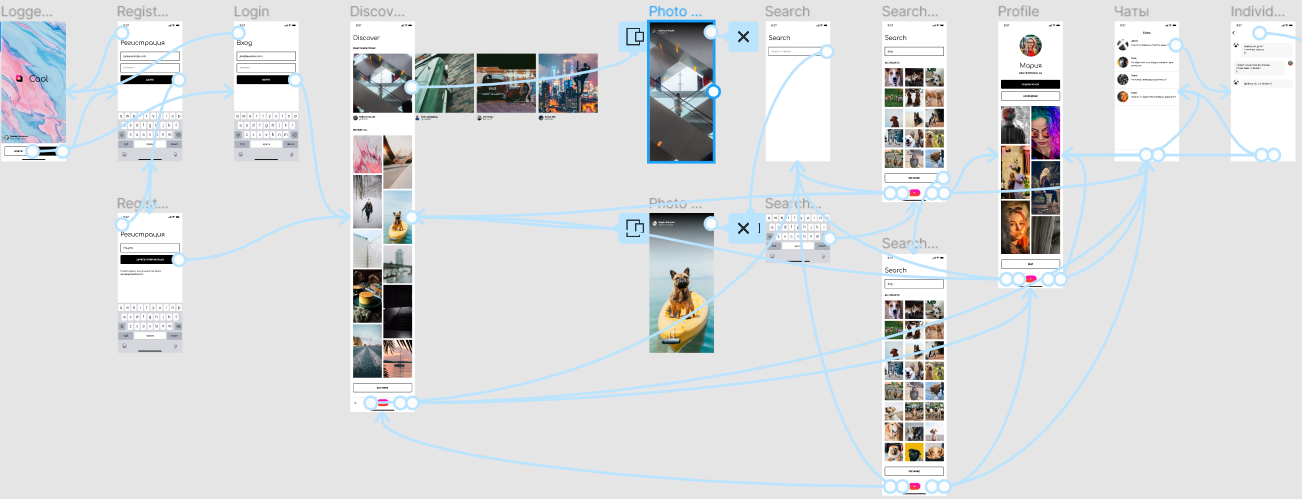


Рисунок 2 – схема переходов в прототипе

Ниже рассмотрены некоторые созданные формы.

На рисунке 3 изображено окно начальной страницы.

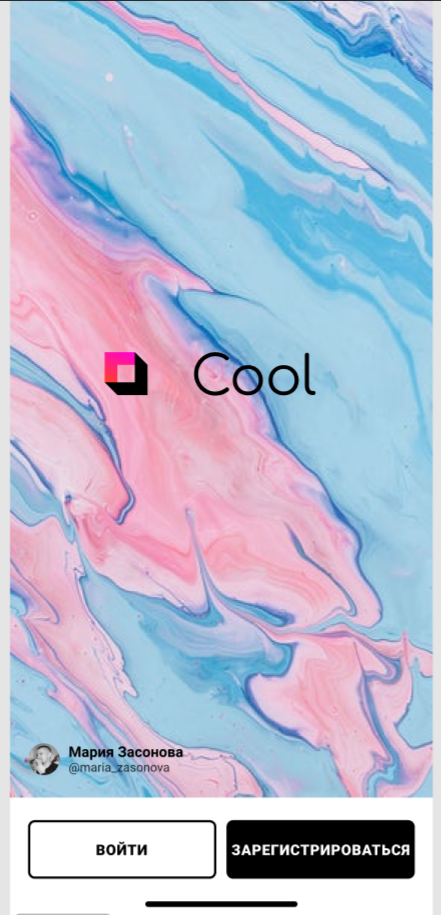


Рисунок 3 – главное меню

На рисунках 4 и 5 изображены практически идентичные формы регистрации и входа соответственно.



Рисунок 4 – Форма регистрации

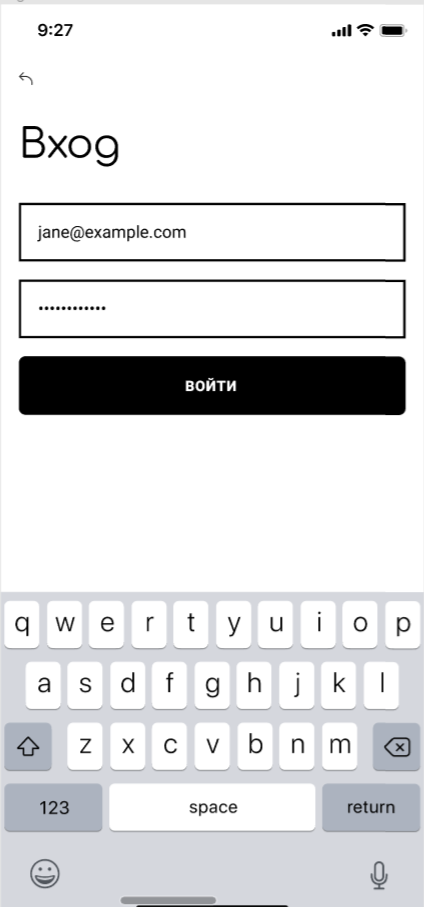


Рисунок 5 – Форма входа

На рисунках 6 и 7 изображены форма со списком диалогов и форма с развернутым одним диалогом соответственно.

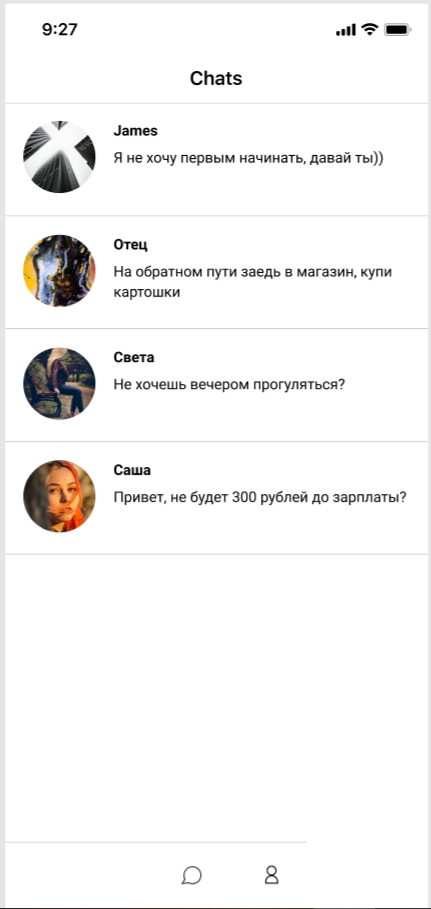


Рисунок 6 – Форма со списком диалогов

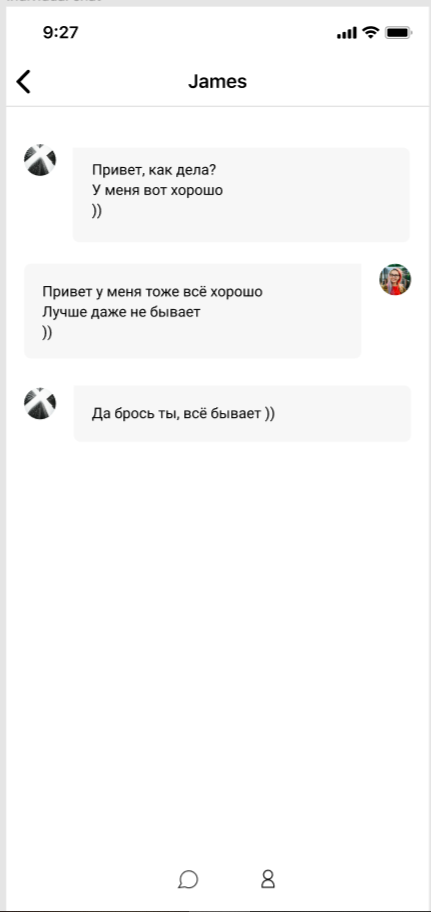


Рисунок 7 – Форма диалога с другим пользователем

Далее на рисунках 8 и 9 показаны соответственно стартовая форма поиска фотографий с возможностью поиска по ключевым словам и страница поиска с введенным ключевым словом “собачке”.

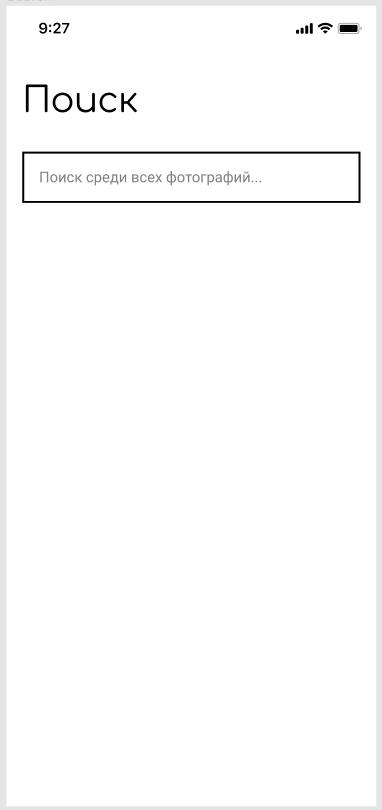


Рисунок 8 – Стартовая страница формы поиска

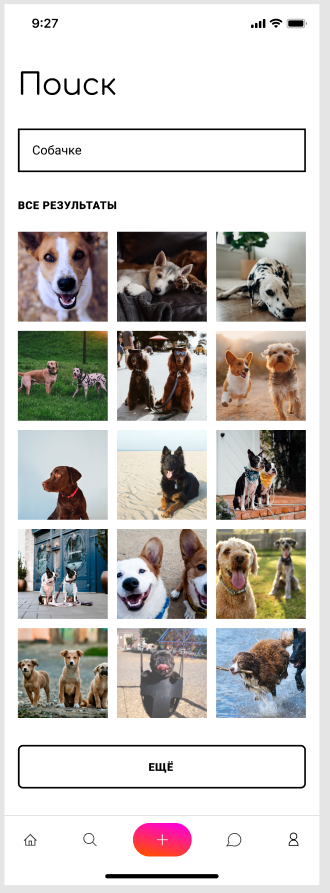


Рисунок 9 – Страница формы поиска

# **ШАБЛОНЫ ПРОЕКТИРОВАНИЯ**

В качестве шаблона проектирования был выбрал шаблон «Фасад». Это структурный паттерн проектирования, который предоставляет простой интерфейс к сложной системе классов [6]. Он «знает», каким классам нужно переадресовать запрос, и какие данные для этого нужны. Используя данный паттерн, реализуем интерфейс к подсистеме авторизации пользователей.

Положительные аспекты использования данного шаблона:

* изолирует клиентов от компонентов сложной подсистемы.

Отрицательные аспекты использования данного шаблона:

* «Фасад» рискует стать объектом, привязанным ко всем классам программы.

class Authorization {

constructor()

{

this.login = document.forms['form']['Login'].value;

this.password = document.forms['form']['password'].value;

this.exit = false

}

getLogin()

{

return this.login;

}

getPassword()

{

return this.password;

}

checkData()

{

if (strLogin.length == 0)

{

document.getElementById('loginf').innerHTML = '\*данное поле обязательно для заполнения';

this.exit = false;

}

else

document.getElementById('loginf').innerHTML = '';

if (strPassword.length == 0)

{

document.getElementById('passwordf').innerHTML = '\*данное поле обязательно для заполнения';

this.exit= false;

}

else

document.getElementById('passwordf').innerHTML = '';

}

checkExit(data)

{

serverData = JSON.parse(data);

if (serverData.success == true)

{

this.exit = true;

}

else {

if (data.login == false)

{

document.getElementById('loginf').innerHTML = '\*неверный логин';

this.exit = false;

}

else

{

document.getElementById('passwordf').innerHTML = '\*неверный пароль';

this.exit = false;

}

}

if (this.exit == true)

{

window.localStorage.setItem("user", this.login);

window.location.href = "MenuPage.html";

}

}

errorExit()

{

alert("Error");

}

}

//ajax запрос на сервер

$("submit").click( function (e) {

Authorization aut();

aut.checkData();

let login = aut.getLogin();

let password = aut.getPassword();

$.ajax({

url: "http://localhost/login.php",

type: 'POST',

data: {

login: login,

password: password

},

success: aut.checkExit,

error: aut.errorExit

});

# **ВЫВОДЫ**

При выполнении данного задания, были изучены различные методы прототипирования, из которых был выбран метод эволюционного прототипирования.

Также была изучена платформа прототипирования Figma, позволившая создать прототип приложения.

Также были изучены различные паттерны проектирования. В качестве шаблона был выбран паттерн «Фасад», который лучше остальных подходит под данный прототип приложения. Главным критерием к шаблону была его легкая реализуемость, а также возможность изоляции пользователя от компонентов сложной подсистемы.

# **ЗАКЛЮЧЕНИЕ**

В результате выполнения задания был разработан прототип приложения “Мессенджер Cool”, выбран эволюционный метод прототипирования и шаблон проектирования Фасад, а также были выполнены все функциональные требования к прототипу.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Прототипирование пользовательского интерфейса URL: http://www.4stud.info/user-interfaces/prototypes-and-concept-design.html [Электронный ресурс] (дата обращения: 16.12.2020).
2. Эволюционное прототипирование [Электронный ресурс]: URL: http://www.4stud.info/user-interfaces/prototypes-and-concept-design.html (дата обращения 20.12.2020)
3. Что такое Figma и как ей пользоваться. URL: https://skillbox.ru/media/design/chto\_takoe\_figma/ [Электронный ресурс] (Дата обращения: 16.12.2020)
4. Figma: обзор программы для веб-дизайна URL: https://sendpulse.com/ru/blog/figma [Электронный ресурс] (Дата обращения: 16.12.2020)
5. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного программирования. Паттерны проектирования. СПб.: Питер, 2001.
6. Паттерн “Фасад”. URL: https://refactoring.guru/ru/design-patterns/facade [Электронный ресурс] (дата обращения: 16.12.2020).