Курсов проект

Дисциплина: Проектиране и интегриране на софтуерни системи

Фаза 3: Реализация на системата

Версия 1.1

|  |  |  |
| --- | --- | --- |
| Фак. № | Име на студент | Секция от документа |
| 61950 | Иван Младенов | 1,2,3,4,5,6,7 |

Януари, 2019

Съдържание

[1 Въведение 3](#_Toc534970184)

[1.1 Цел 3](#_Toc534970185)

[1.2 Резюме 3](#_Toc534970186)

[1.3 Дефиниции и акроними 3](#_Toc534970187)

[2 Използвани технологии 4](#_Toc534970188)

[3 Реализация на базата от данни 5](#_Toc534970189)

[4 Реализация на бизнес логиката 6](#_Toc534970190)

[5 Реализация на потребителския интерфейс 7](#_Toc534970191)

[6 Внедряване на системата 8](#_Toc534970192)

# Въведение

## Цел

Настоящият документ има за цел да покаже използваните технологии, реализацията на бизнес логиката, архитектурата на приложението, както и потребителския интерфейс на системата за размяна на подаръци SecretSanta.

## Резюме

Документът предоставя описание на системното проектиране и реализацията на системата за размяна на подаръци SecretSanta. Втората част на документа предоставя използваните технологии, разделени на три части – за всеки от модулите на системата. Третата част описва схемата на базите данни. В четвъртата част е представена реализацията на бизнес логиката. Петата част от документа показва потребителския интерфейс. Шестата част описва стъпките, необходими за внедряването на системата. Последната седма част показва разпределението на задачите по проекта.

# Използвани технологии

Проектът е изграден от 3(три) отделни системи, за всяка от които са използвани различни технологии:

* **Сървър**

Сървърното приложението е реализирано с ASP.NET Core 2.0 и езика C#. За IDE е използвано Visual Studio 2017. Приложението комуникира чрез REST API

* **Клиент**

Клиентската част е реализирана като single page application (SPA), използвайки библиотеката React.js заедно с Redux. Използвано IDE е Visual Studio Code. Клиента комуникира със сървъра чрез REST API и с чат сървъра чрез Web Sockets

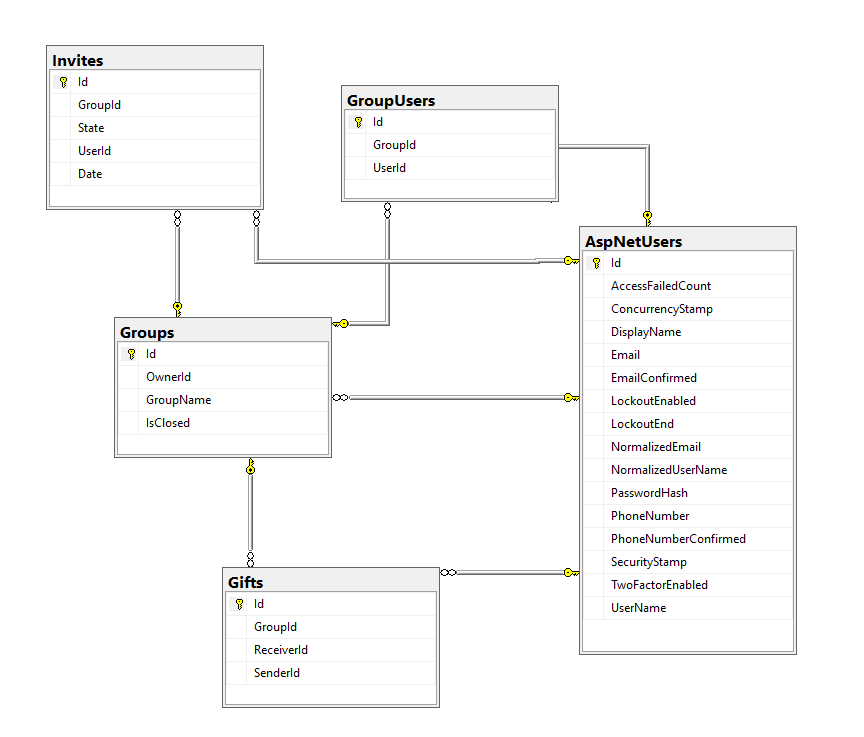
* **Чат сървър**

Приложението за чат е реализирано чрез езика Javascript върху изпълняващата среда Node.js с IDE Visual Studio Code. Приложението комуникира с клиентската част чрез използването на Web Sockets

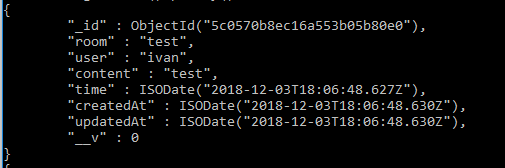
# Реализация на базата от данни

*Системата използва две бази данни – една за сървърното приложение, отговаряща за основната бизнес логика, и една за чат приложението, в която се пазят изпратените съобщения.*

*За сървърното приложение е използвана релационна база данни Microsoft SQL Server 2016. Прикачена е диаграма на схемата на базата.*



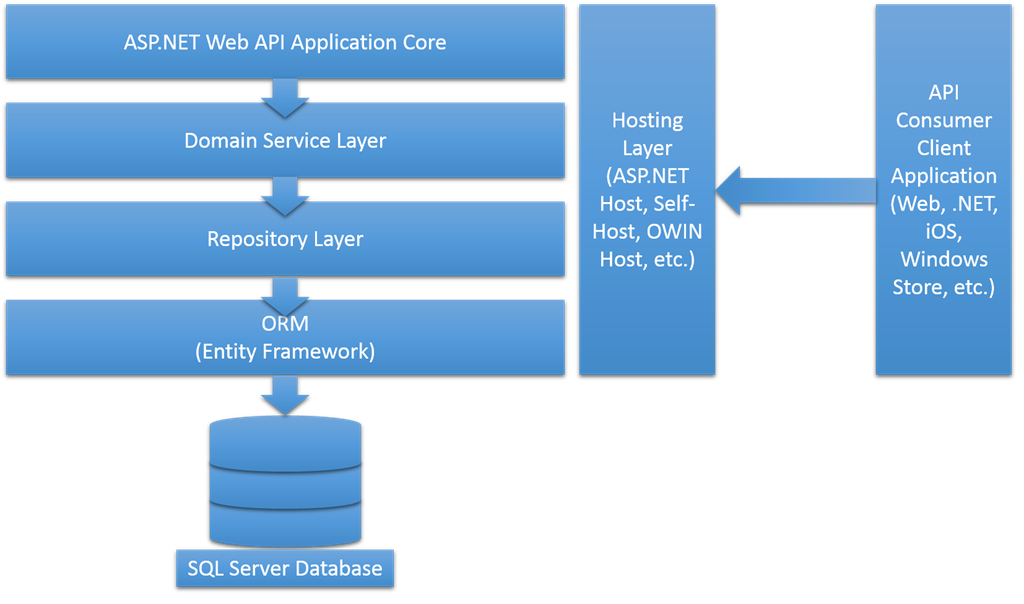
*Чат приложението използва нерелационна база данни (документно-базирана) MongoDB. В тази база данни има само една таблица – messages, която съдържа изпратените съобщения. Това е примерно съобщение*



# Реализация на бизнес логиката

*Системата е разделена на 3 основни подсистеми – сървър, клиентско приложение и чат сървър.*

*Сървърното приложение е на база на слоева архитектура (layered). Всеки слой комуникира само със съседните му чрез програмни интерфейси. Така конкретната имплементация на даден слой може лесно да бъде подменена без да трябва да се преправя старият код.*



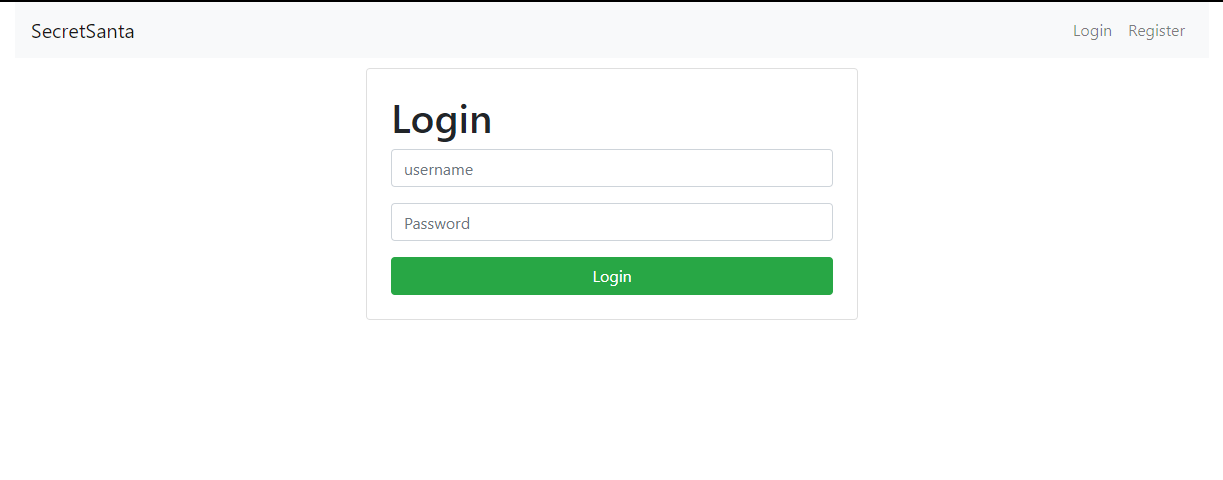
*Основната бизнес логика е имплементирана в Service Layer-а на приложението. Controller-ите изполват интерфейсите от Service Layer-а и създават REST API за комуникация с други приложения или клиенти. Service Layer-ът използв Data Access Layer, който използва Repository and Unit of Work Pattern за комуникация с базата данни.*

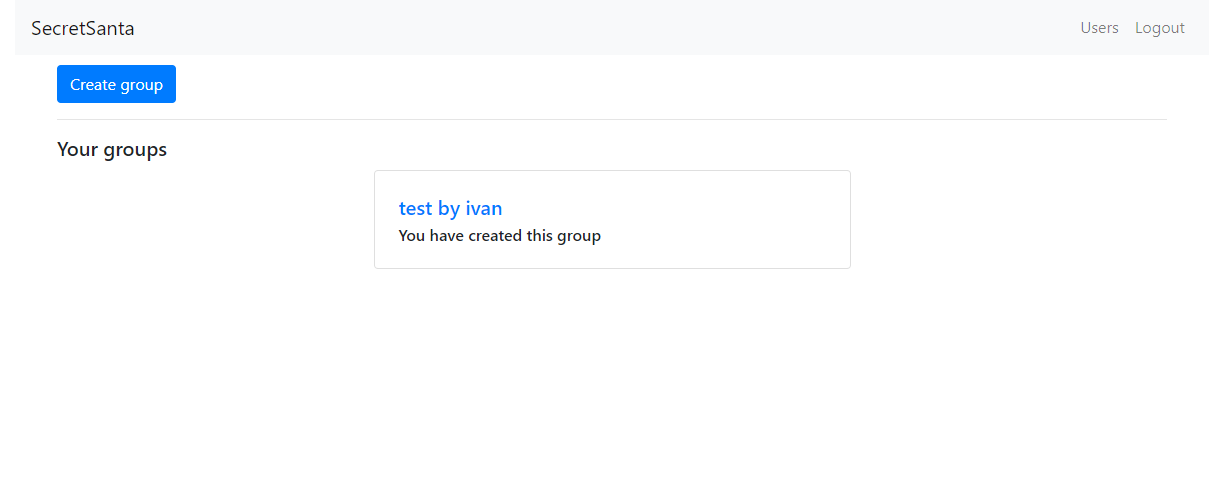
*Клиентското приложение е разделено на отделни domain-и (модули), водейки се от концепциите в Domain Driven Design. Всеки отделен модул отговаря за конкретно entity в приложението (например message, notification, group, user и т.н).*

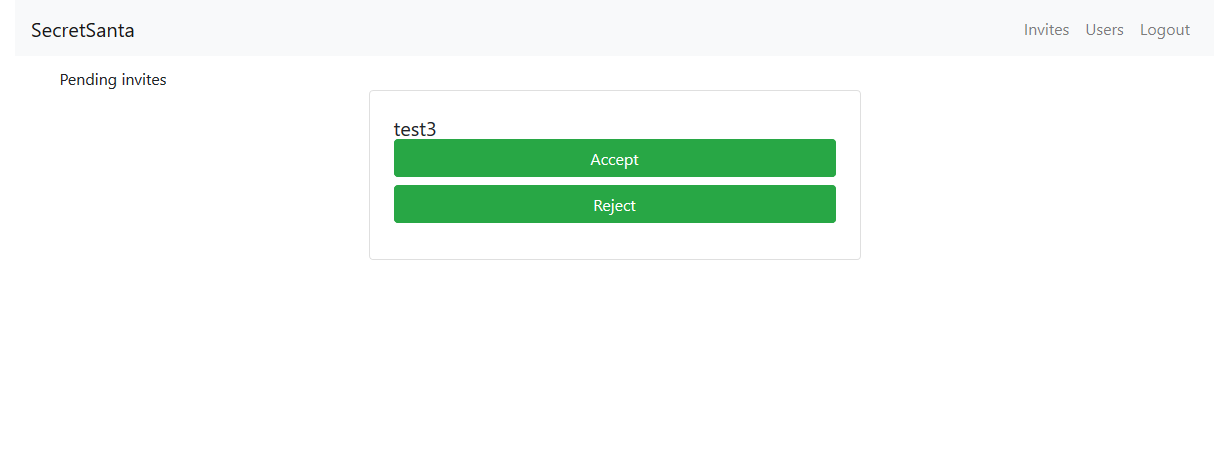
*Чат сървъра използва MVC архитекрура и Object Relational Mapper (ORM) за комуникация с базата данни. Библиотеката socket.io се използва за комуникацията през Web Sockets и REST API, което зарежда всичките съобщения за дадена група.*

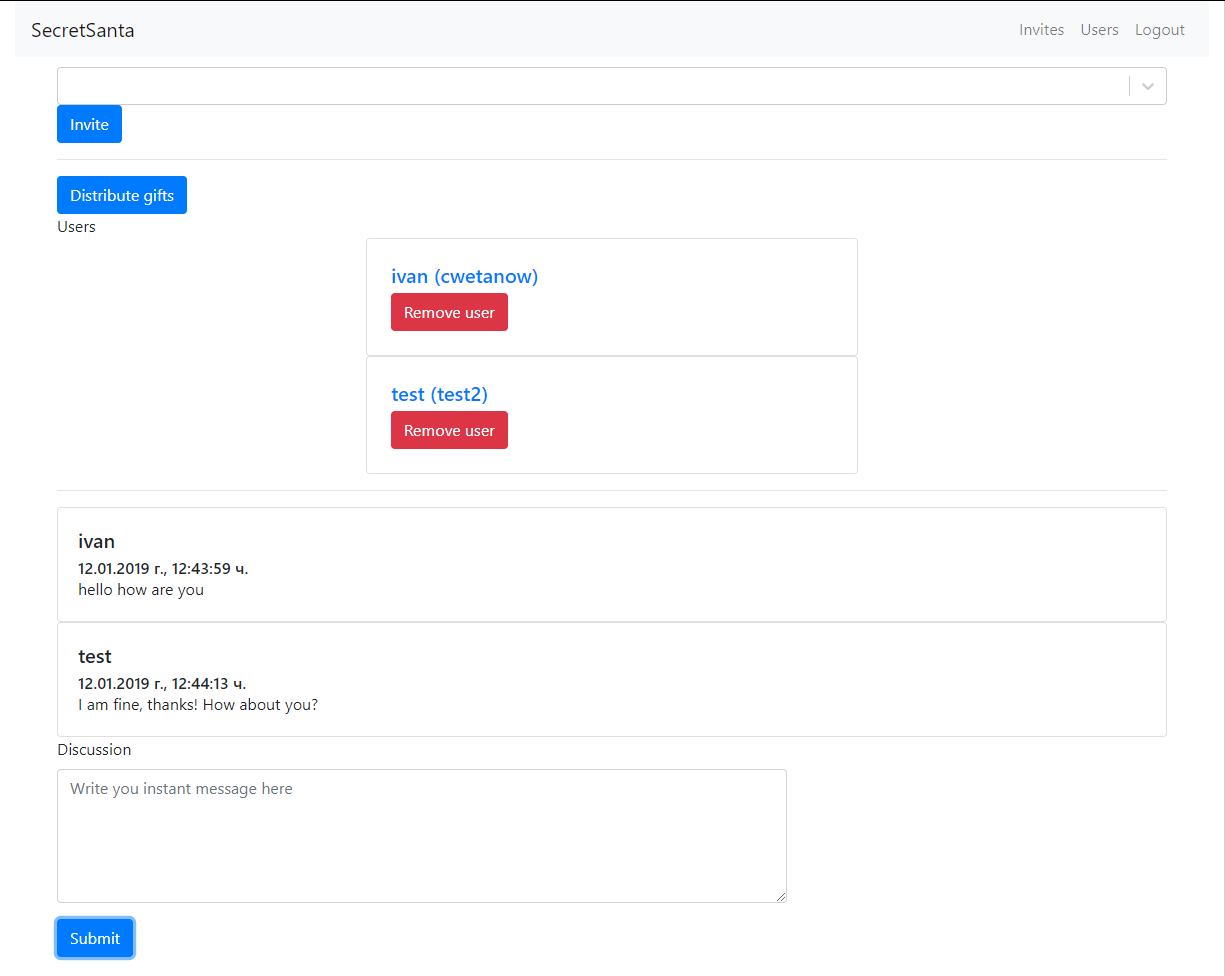
# Реализация на потребителския интерфейс

*Потребителският интерфейс представлява клиентско Single Page Application, реализиране с помощта на React.js и Redux. Има основна структура на html документа, и всеки един компонент (изглед) се рендерира на определена част от екрана.*

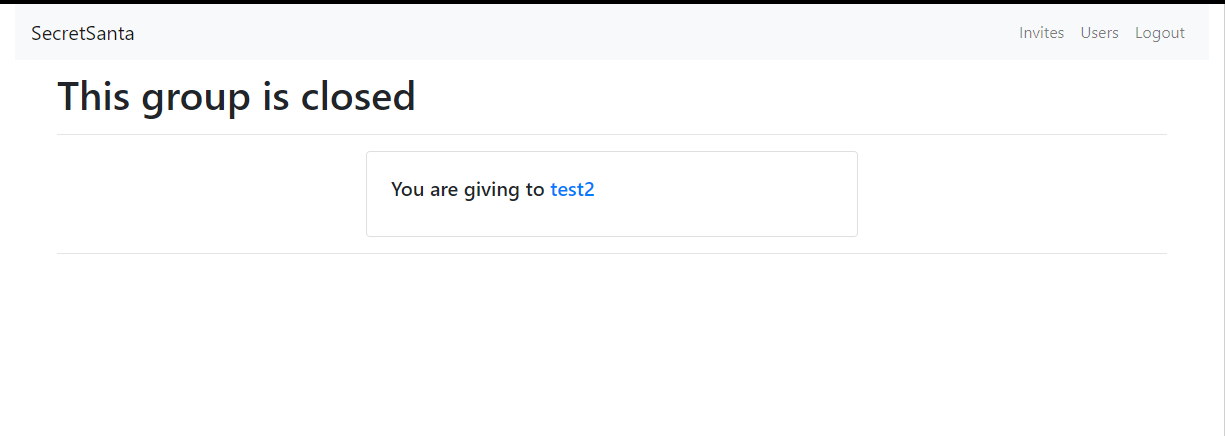
*Екран за вписване в системата*

*Основен екран след вписване на потребител, в който са показани всичките му групи, както и бутон за създаване на нова група.*

*Екран със всички неотговорени покани*



*Основен изглед на група, когато текущият потребител е създател на групата. Може да изпраща покани за нов потребител, да разпредели подаръците, да премахва потребител от групата и да вижда и пише в чата.*

*Излгед на затворена група, след разпределение на подаръците*

# Внедряване на системата

*Опишете изискванията и последователността от действия за внедряване на системата.*

*За внедряването на системата се изисква сървър с инсталирани на него:*

* *ASP.NET Core 2.0*
* *Microsoft SQL Server 2016*
* *Node.js*
* *MongoDB*

*За стартиране на чат сървъра са нужни следните стъпки:*

1. *Да се поставят определени настройки в chat-server/.env файла, а именно PORT (порт, на който ще бъде пуснато приложението) и CONNECTION\_STRING (връзка с базата данни)*
2. *Да се отвори терминал в папката chat-server*
3. *Изпълнява се командата* ***npm install*** *– инсталира всички пакети, нужни на приложението*
4. *Изпълнява се командата* ***npm start –*** *стартира приложението*

*За стартиране на сървърното приложение са нужни следните стъпки:*

1. *Да се добавян нужните настройки във файла server/SecretSanta.Web/appsettings.json за връзка с базата данни*
2. *Изпълняват се командате* **dotnet ef migrations add InitialCreate** и **dotnet ef database update** за създаване на базата данни
3. *Изпълнява се командата* ***dotnet run --project SecretSanta.Web.csproj*** *за стартиране на приложението*

*За стартиране на клиентското приложение са нужни следните действия:*

1. *Изпълнява се командата* ***npm install*** *– инсталира всички пакети, нужни на приложението*
2. *Изпълнява се командата* ***npm start –*** *стартира приложението*