Курсов проект

Дисциплина: Проектиране и интегриране на софтуерни системи

Фаза 2: Анализ на изискванията и проектиране на системата

Версия 1.x

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

Ноември, 2018

Съдържание

[1 ВЪВЕДЕНИЕ 3](#_Toc372276266)

[1.1 Цел 3](#_Toc372276267)

[1.2 Обхват 3](#_Toc372276268)

[1.3 Резюме 3](#_Toc372276269)

[1.4 Дефиниции и акроними 3](#_Toc372276270)

[2 СИСТЕМНА АРХИТЕКТУРА 4](#_Toc372276271)

[2.1 Архитектурно проектиране 4](#_Toc372276272)

[2.2 Описание на декомпозицията 4](#_Toc372276273)

[2.3 Обосновка на проектирането 4](#_Toc372276274)

[3 ДАННОВО ПРОЕКТИРАНЕ 5](#_Toc372276275)

[3.1 Описание на данните 5](#_Toc372276276)

[3.2 Речник на данните 5](#_Toc372276277)

[4 КОМПОНЕНТНО ПРОЕКТИРАНЕ 6](#_Toc372276278)

[5 ПРОЕКТИРАНЕ НА ПОТРЕБИТЕЛСКИЯ ИНТЕРФЕЙС 7](#_Toc372276279)

[5.1 Обобщение на потребителския интерфейс 7](#_Toc372276280)

[5.2 Екранни изображения 7](#_Toc372276281)

[5.3 Екранни обекти и действия 7](#_Toc372276282)

[6 МАТРИЦА НА ИЗИСКВАНИЯТА 8](#_Toc372276283)

[7 ПРИЛОЖЕНИЯ 9](#_Toc372276284)

# ВЪВЕДЕНИЕ

## Цел

*Опишете на целта на документа.*

Настоящият документ описва изискванията към системата за размяна на подаръци SecretSanta

## Обхват

*Опишете обхвата на софтуерната система и изяснете целта и ползите от нея.*

## Резюме

*Създайте резюме на документа и опишете структурата му.*

## Дефиниции и акроними

*Дефинирайте всички термини, понятия и акроними, използвани в документа.*

# АНАЛИЗ НА ИЗИСКВАНИЯТА

## Функционални изисквания

Примерен формат на таблица за описание на функционални изисквания.

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | Изискване | Коментар | Приоритет |
| FR\_01 | Системата задължително трябва да предоставя функционалност за регистация на потребител | При регистрация, всеки потребител задава уникално потребителско има, име и парола | 1 |
| FR\_02 | Системата задължително трябва да предоставя функционалност за вписване на потребител | Вписването се извършва с потребителско име и парола | 1 |
| FR\_03 | Препоръчително е системата да предоставя функционалност за отписване на потребител |  | 3 |
| FR\_04 | Системата задължително трябва да предоставя функционалност за търсене в списъка с потребители | Всеки потребител може да получи списък на всички останали потребители в системата. Този списък трябва да позволява странициране, сортиране и търсене по име | 1 |
| FR\_05 | Системата задължително трябва да предоставя функционалност за преглед на профил на даден потребител |  | 1 |
| FR\_06 | Системата задължително трябва да предоставя функционалност за създаване на групи | Всяка група трябва да има уникално име | 1 |
| FR\_07 | Системата задължително трябва да предоставя функционалност за създателят на група да изпраща покана за присъединяване | Създателят на групата може да изпраща покана към потребител за присъединяване към групата | 1 |
| FR\_08 | Системата задължително трябва да предоставя функционалност потребителят да може да прегледа всички чакащи покани | Списъкът с покани трябва да има възможност за странициране и сортиране по дата на получаване | 1 |
| FR\_09 | Системата задължително трябва да предоставя функционалност за приемане или отхвърляне на чакаща покана | Когато поканата е приета, потребителят се присъединява в групата. Ако бъде отхвърлена, се премахва от списъка с чакащи покани | 1 |
| FR\_10 | Системата задължително трябва да предоставя функционалност за създателят на групата да стартира процеса по 'свързване' | Свързването на потребители става случайно. Потребител не може да подарява сам на себе си. Всеки потребител се свързва точно с един друг, на който да подарява и точно един, от когото да получи подарък | 1 |
| FR\_11 | Системата задължително трябва да предоставя функционалност за потребителите да получат списък с групите, в които членуват | Този списък трябва да позволява странициране и да е сортиран по азбучен ред | 1 |
| FR\_12 | Системата задължително трябва да предоставя функционалност за проверка дали вече има свързан получател в дадена група | Ако процесът е стартиран, потребителят трябва да получи информация на кого трябва да подарява. В противен случай получава съобщение, че все още няма разпределение | 1 |
| FR\_13 | Системата задължително трябва да предоставя функционалност за преглед на списък с потребителите в групата | Списъкът с потребителите в група трябва да позволява странициране | 1 |
| FR\_14 | Системата задължително трябва да предоставя функционалност за премахване на член на групата от създателят |  | 1 |
| FR\_15 | Препоръчително е системата да позволява на членовете на дадена група да пишат съобщения в общия чат |  | 2 |

## Нефункционални изисквания

### Изисквания към потребителския интерфейс

*Опишете характеристиките на потребителския интерфейс – формати, шаблони, структура на менюто, съобщения за грешка и др. съобщения, функционални клавиши и др.*

### Производителност

Системата трябва да позволява едновременна работа на поне 50 потребителя.

### Наличност

Системата трябва да е налична поне 95% от времето.

### Сигурност

Системата трябва да използва алгоритъм за хеширане на пароли в базата данни, за да не позволява при случай на пробив в базата, паролите да изтекат.

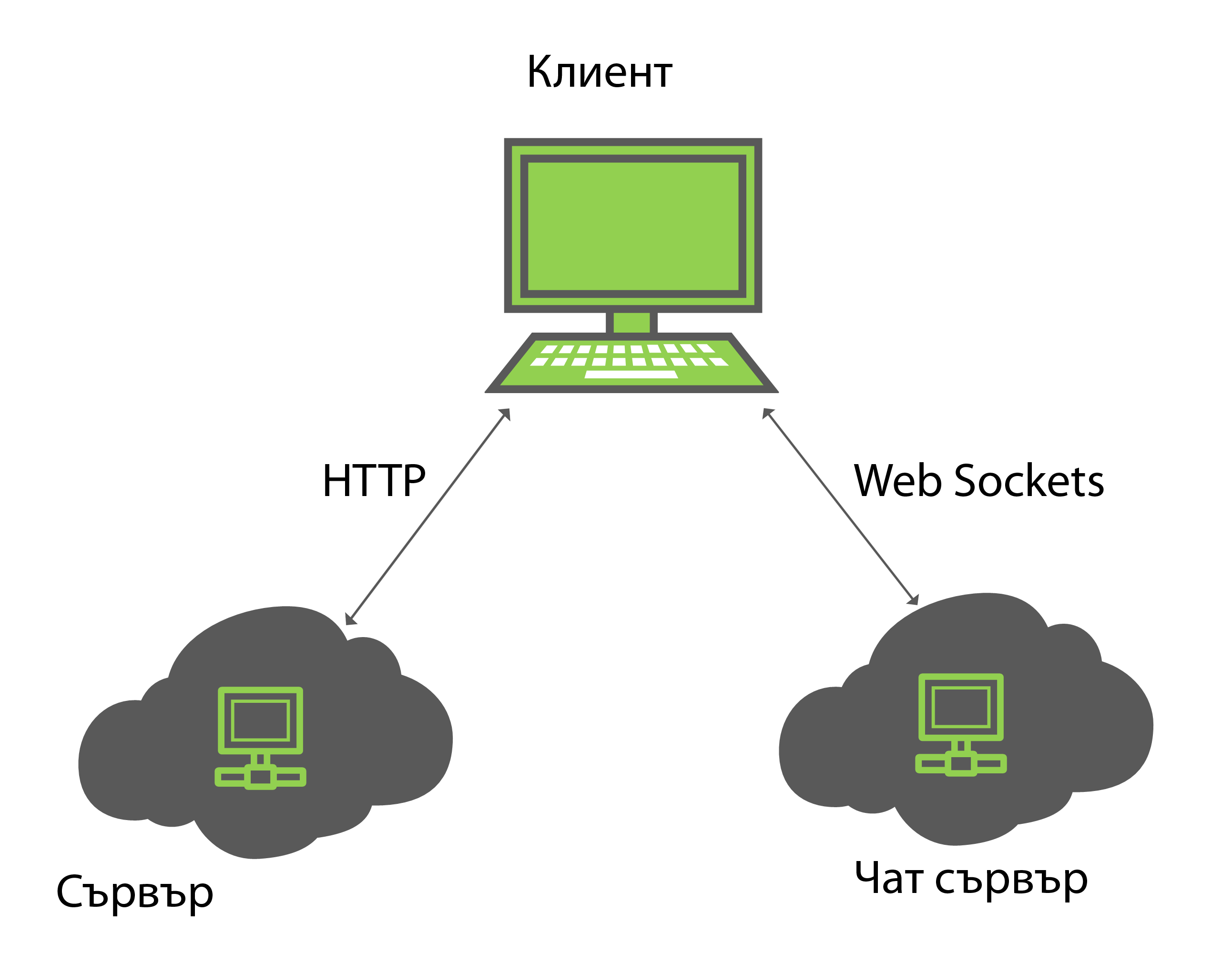
# СИСТЕМНА АРХИТЕКТУРА

## Архитектурно проектиране

Системата е разделена на следните модули:

* Сървър (сървърно приложение, реализирано с ASP.NET Core 2.0 и езика C#)
* База данни на приложението (релационна база от данни на Microsoft SQL Server)
* Чат сървър (сървърно приложение, реализирано на node.js)
* База данни на чат приложението (нерелационна база данни на MongoDB)
* Клиент (single page приложение, реализирано с библиотеката React.js)

Сървърното приложение съдържа всичката бизнес логика на приложението. То комуникира с базата от данни, също така комуникира с клиента чрез REST протокола. Чат приложението използва нерелационна база от данни за складиране на съобщенията и комуникира с клиента чрез web sockets. Двете сървърни приложения нямат връзка едно с друго, като комуникацията минава през клиентското приложение.



## Описание на декомпозицията

*Декомпозирайте подсистемите, идентифицирани при архитектурното проектиране. Осигурете функционално или обектно-ориентирани описание на системните компоненти. При функционалното описание използвайте диаграми на данновия поток (data flow diagrams) от високо ниво и структурни декомпозиционни диаграми. При обектно-ориентирани описание осигурете модел на подсистемите, обектни диаграми, обобщаващи йерархични диаграми (generalization  hierarchy  diagrams), агрегиращи йерархични диаграми (aggregation  hierarchy diagrams), интерфейсни спецификации и др.*

## Обосновка на проектирането

*Архитектурата на системата, описана в Секция 3.1 позволява лесно разделяне на отговорностите на отделните модули и лесна смяна на използваната технология. Също така работата по всеки компонент може да е независима от другите. Клиентското приложение отговаря за представянето на данните на потребителя, без да извършва сложна бизнес логика. За нея е отговорно съврърното приложение, което използва релационна база данни, поради факта, че таблиците в нея отговарят на отделните типове данни в системата. Приложението за чат е отделено от сървъра, за да позволява скалиране в случай на много едновременни съобщения и неочаквана натовареност. Също така използва нерелационна база данни, защото така отделните записи са независими един от друг и нямат връзки с другата част от системата.*

# ДАННОВО ПРОЕКТИРАНЕ

## Описание на данните

*User – потребител на системата*

*Group – група в системата*

*Invite – покана за присъединяване към група*

*GroupUser – членство на даден потребител в група*

*Gift – разпределен подарък в група*

*Message – единично съобщение в чата на група*

## Речник на данните

# КОМПОНЕНТНО ПРОЕКТИРАНЕ

*Системата прилага определен алгоритъм за разпределяне на подаръците между присъстващите в групата, реализиран в метода SecretSantaShuffle.*

private IEnumerable<int> SecretSantaShuffle(int numberOfUsers)

{

var assignments = new List<int>();

for (var i = 0; i < numberOfUsers; i++)

{

assignments.Add(i);

}

var isValid = false;

while (!isValid)

{

assignments.Shuffle();

isValid = true;

for (var i = 0; i < numberOfUsers; i++)

{

if (assignments[i] == i)

{

isValid = false;

}

}

}

return assignments;

}

Методът приема като параметър броя на потребителите в групата и създава списък, като на всеки потребител слага уникален индекс и число, равно на съответния индекс. След това разбръква списъка произволно, докато на всеки индекс в списъка отговаря различно число. Това позволява никой потребител да не трябва да подарява на себе си.

# ПРОЕКТИРАНЕ НА ПОТРЕБИТЕЛСКИЯ ИНТЕРФЕЙС

## Обобщение на потребителския интерфейс

*Опишете функционалността на системата от потребителска гледна точка. Изяснете как потребителят ще използва функционалността на системата и механизмите за обратна връзка към него.*

## Екранни изображения

*Приложете изображения, показващи потребителския интерфейс. Не се изискват реални екранни изображение (screenshots) от системата, но ако проектът е в по-напреднала фаза, могат да се приложат.*

# МАТРИЦА НА ИЗИСКВАНИЯТА

*Създайте матрица на съответствието между изискванията към системата, дефинирани в спецификацията на изискванията, и компонентите и данновите структури, описани в настоящия документ. Използвайте табличен формат, за да покажете кои системни компоненти удовлетворяват функционалните изисквания към системата. Реферирайте функционалните изисквания по кодовете, зададени в спецификацията на изискванията (FR1.01, FR1.02, ...).*