**Курсов проект**

**Дисциплина: Проектиране и интегриране на софтуерни системи**

**Фаза 2: Анализ на изискванията и проектиране на системата**

**Версия 1.0**

|  |  |  |
| --- | --- | --- |
| **Фак. №** | **Име на студент** | **Секция от документа** |
| 61685 | **Красимир Николов** |  |
| 61701 | **Валентин Кирилов** |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Ноември, 2017**

1. **Съдържание**

[1 ВЪВЕДЕНИЕ 3](#_30j0zll)

[1.1 Цел 3](#_1fob9te)

[1.2 Обхват 3](#_3znysh7)

[1.3 Резюме 3](#_2et92p0)

[1.4 Дефиниции и акроними 3](#_tyjcwt)

[2 СИСТЕМНА АРХИТЕКТУРА 4](#_3dy6vkm)

[2.1 Архитектурно проектиране 4](#_1t3h5sf)

[2.2 Описание на декомпозицията 4](#_4d34og8)

[2.3 Обосновка на проектирането 4](#_2s8eyo1)

[3 ДАННОВО ПРОЕКТИРАНЕ 5](#_17dp8vu)

[3.1 Описание на данните 5](#_3rdcrjn)

[3.2 Речник на данните 5](#_26in1rg)

[4 КОМПОНЕНТНО ПРОЕКТИРАНЕ 6](#_lnxbz9)

[5 ПРОЕКТИРАНЕ НА ПОТРЕБИТЕЛСКИЯ ИНТЕРФЕЙС 7](#_35nkun2)

[5.1 Обобщение на потребителския интерфейс 7](#_1ksv4uv)

[5.2 Екранни изображения 7](#_44sinio)

[5.3 Екранни обекти и действия 7](#_3j2qqm3)

[6 МАТРИЦА НА ИЗИСКВАНИЯТА 8](#_2jxsxqh)

[7 ПРИЛОЖЕНИЯ 9](#_z337ya)

# ВЪВЕДЕНИЕ

## Цел

Настоящият документ описва изготвянето на основните изисквания, архитектурата на базата данни, системната архитектура и потребителските изисквания за текущия проект. Този документ цели да служи като добър източник на информация по горепосочените етапи от изграждането на системата. Документът е насочен както към разработчиците на проекта, така и към всички останали, които работят по него под една или друга форма.

## Обхват

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

Една от основните целите на този проект е да се изгради RESTful API, които предоставя достъп до съдържание от новини. Данните, които могат да се достъпват ще бъдат използвани от различни клиентски приложения, които ще предоставят данните до крайните потребители.

## Резюме

Документът е разделен на следните части:

* анализ на изискванията - съдържа описание на системните изисквания по начин, който подпомага процеса на проектиране, както и процеса на тестване на системата. Изброени са функционалните и нефункционалните изисквания на системата.
* системна архитектура - съдържа описание на модулната структура на системата и взаимовръзките между отделните модули от гледна точка на функционалността ѝ. Идентификация на подсистемите от високо ниво заедно с техните роли и отговорности. Описание на взаимодействието между подсистемите. Съдържа диаграма, представяща модулната архитектура на системата.
* данново проектиране - съдържа описание на това как информационният домейн на системата се трансформира в даннови структури. Изяснено е как данните се съхраняват, организират и обработват. Описана е същността и типовете на данновите елементи на системата.
* компонентно проектиране - съдържа описание по систематичен начин на съществените компоненти на системата.
* проектиране на потребителския интерфейс - съдържа описание на функционалността на системата от потребителска гледна точка и това как потребителят ще използва функционалността на системата и механизмите за обратна връзка към него. Предоставя изображения, показващи потребителския интерфейс.
* матрица на изискванията

## Дефиниции и акроними

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

# АНАЛИЗ НА ИЗИСКВАНИЯТА

Приоритет на изискванията:

* Приоритет 1 – изискването е „задължително“
* Приоритет 2 – изискването е „необходимо“ за подобряване на софтуера
* Приоритет 3 – изискването е „препоръчително“

## Функционални изисквания

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | Изискване | Коментар | Приоритет |
| FR\_01 | Системата трябва да предоставя API интерфейс за взимане на всички статии | Номерът на поръчката се генерира като комбинация от датата и часа на нейното създаване и идентификатора на потребителя. | 1 |
| FR\_02 | Системата трябва да предоставя API интерфейс за взимане на всички категории | Статиите в системата са асоцирани с дадена категория | 2 |
| FR\_03 | Системата трябва да предоставя API интерфейс за взимане на всички тагове | Статиите в системата са асоцирани с определени тагове | 2 |
| FR\_04 | Системата трябва да предоставя API интерфейс за взимане на всички автори | Статиите в системата са асоцирани с даден автор | 2 |
| FR\_05 | Системата трябва да предоставя API интерфейс за взимане на всички стати по зададени критери | Статиите могат да се достъпват по дадена категория, таг или в зависимост от датата на публикуване | 2 |
| FR\_06 | Системата трябва да предоставя клиентско приложение, което да визуализира статиите взети от API | Данните, които предоставя API-то трябва да бъдат достъпвани от клиентско приложение | 1 |
| FR\_07 | Клиентското приложение трябва да предоставя възможност за прочит на дадена статия | Приложението трябва да предоставя възможност за прочит на дадена статия, след като е получена от сървъра | 2 |
| FR\_08 | Клиентското приложение трябва да предоставя възможност за даване на обратна връзка | Клиентското приложение трябва да има начин, по който да предоставя възможност за обратна връзка | 3 |
| FR\_09 | Клиентското приложение трябва да предоставя възможност за преглед на определа категория | Приложението трябва да може да филтрира данните, като показва статиите от дадена категория | 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.*

* формата на датите предоставяни от API интерфейсите трябва да бъде консистентен и стандартизиран
* при настъпване на грешка, потребителите трябва да получават информативно съобщение

### Производителност

Системата е предвидено да се използва от потребители със средномощни персонални компютри. Това налага въвеждането на изисквания за бързодействие на работата на системата, като максималното време, което трябва да коства на API интерфейс да върне резултат да трае не повече от 3 секунди (при средна връзка с интернет от 50 Mbit/s).

### Наличност

Системата трябва да бъде достъпна по всяко време на денонощието и нейното правилно функциониране да не зависи от натоварвания в пикови часове и голям брой посетители.

### Сигурност

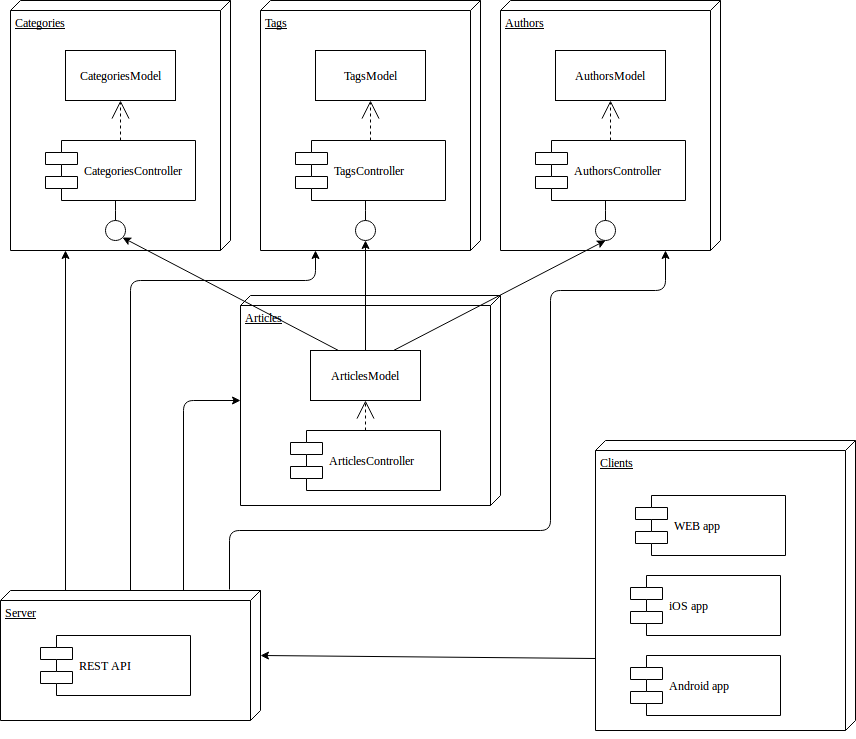
Системата трябва да използва криптирана HTTPS/SSL връзка за трансфер на данни. Ограничавайки възможността за използването на некриптирана HTTP връзка повишава сигурността на системата.

# СИСТЕМНА АРХИТЕКТУРА

## Архитектурно проектиране

*Опишете модулната структура на системата и взаимовръзките между отделните модули от гледна точка на функционалността ѝ. Идентифицирайте подсистемите от високо ниво заедно с техните роли и отговорности. Опишете как подсистемите си взаимодействат. Създайте диаграма, представяща модулната архитектура и я коментирайте.*

Системата се състои от две основни части, които можем да разглеждаме по отделно. Едната от тях представлява сървърно приложение, което представлява REST API, което съхранява данните на системата и предоставя интерфейси, чрез които те да бъдат достъпвани и предоставяне за обработка от външни източници. Това са външни клиенти, които могат да бъдат разработвани от трети лица, но могат да достъпват данните съхранявани от нас посредством дефенираните публични интерфейси. Тези приложения могат да варират в реализацията си и да представляват интернет или мобилни приложения разработени посредством различни технологии и работещи на различни платформи. Поради описаното до момента, архитектурата на системата може да се обособи като клиент-сървър архитектура с един основен сървър и множество клиенти. На фиг. 4.1.1 е показана диаграма, която показва основните модули на системата и тяхното взаимодействие.

**

фиг. 4.1.1 - диаграма на модулната архитектура

Както се вижда на фиг. 4.1.1, сървърът се състои от 4 основни модула. Всеки от тях предоставя възможност за работа с определен тип данни, за който е предназначен - статии, категории, тагове и автори. Всеки модул се състои от 2 по-основни компонента . контролер и модел. Моделът представлява ентити, сериализиран формат за обектите от съответния тип. От своя страна контролерите съдържат бизнес логиката и предоставят възможност за работа със съответните модели. Също така на фигурата можем да видим, че модулът за статиите има явна комуникация с останалите 3. Това е така, защото всяка статия е част от дадена категория, написана е от специфичен автор и има определени тагове.

На схемата се вижда и начинът за комуникация между сървърът и клиентските приложения. Това се случва единствено чрез REST API интерфейсите на сървърът и клиентските приложения имат достъп единствено до публичните интерфейси, които предоставя сървърът. От друга страна, клиентските приложения не са ограничени в това как да обработват и представят данните.

## 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

*Обосновете взетите архитектурни решения в Секция 2.1.*

# ДАННОВО ПРОЕКТИРАНЕ

## Описание на данните

Системата предоставя достъп до съдържание от новини. Това обуслява новините като основният елемент от нашите данни. Тъй като по идея системата предоставя достъп за извличане на данните под различни форми - например филтрирани по специфична категория или даден таг, то това обуслява нуждата от основните структури, които трябва да съхранява нашата база от данни. За целта, в нашата система ще се съхраняват данни за новините, за различните категории и различните тагове. Тези 3 структури са организирани под формата на таблици в базата данни, заедно със съответните връзки помежду си.

## Речник на данните

Системата се състои от няколко основни типа данни. Тук са дадени повече детайли за тяхното предназначение и информацията, която съхранява всеки от тях:

* Категории - представляват разделение между видовете статии съхранявани в системата. Съдържат в себе си уникален идентификатор и име. Обвързани са със статиите с релация 1-към-1.
* Тагове - представляват специфичните характеристики на различните видове статии. Съдържат в себе си уникален идентификатор и име. Обвързани са със статиите с релация много-към-много.
* Автори - съдържат данни за автора на дадена статия. Обвързан със статиите с релация 1-към-1.
* Статии - представляват основният елемент от данни в нашата система, който съдържа данни за различните статии. В това се включва уникален идентификатор, заглавие, кратко описание, линк до статията, категория, тагове, автор, брой прегледи и коментари и дата на публикация на статията.

# КОМПОНЕНТНО ПРОЕКТИРАНЕ

*Опишете по систематичен начин, ако е релевантно, съществените компоненти на системата. Приложете алгоритми и/или псевдокод.*

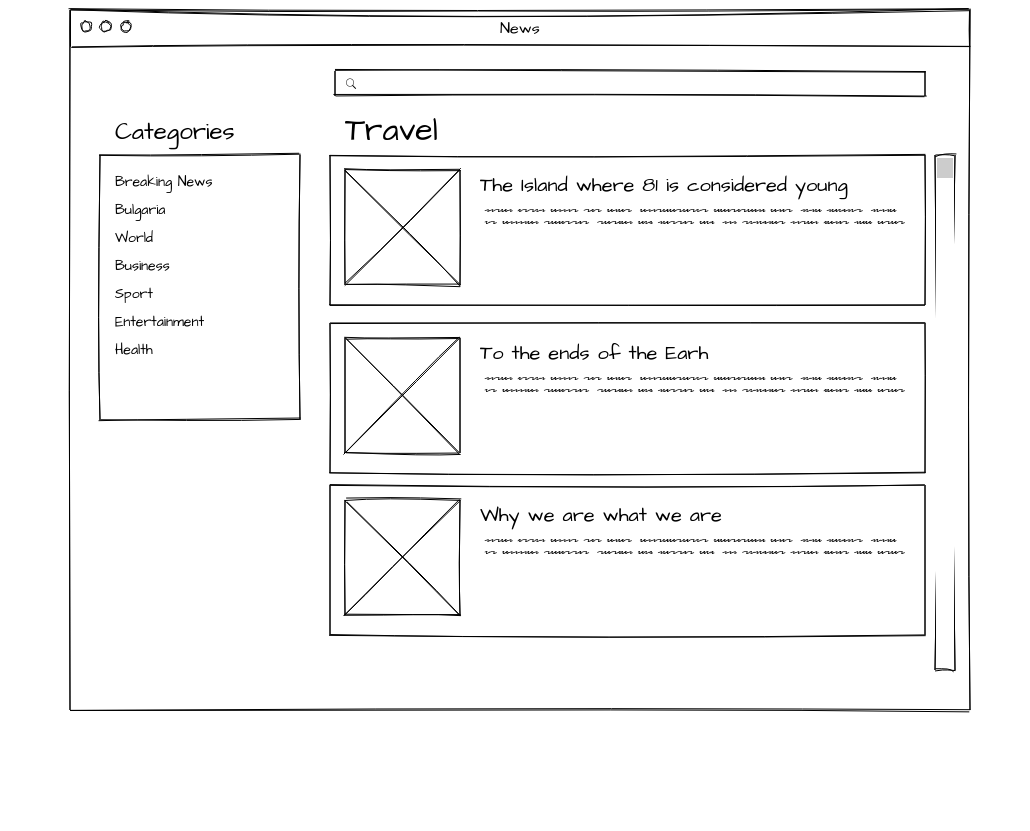
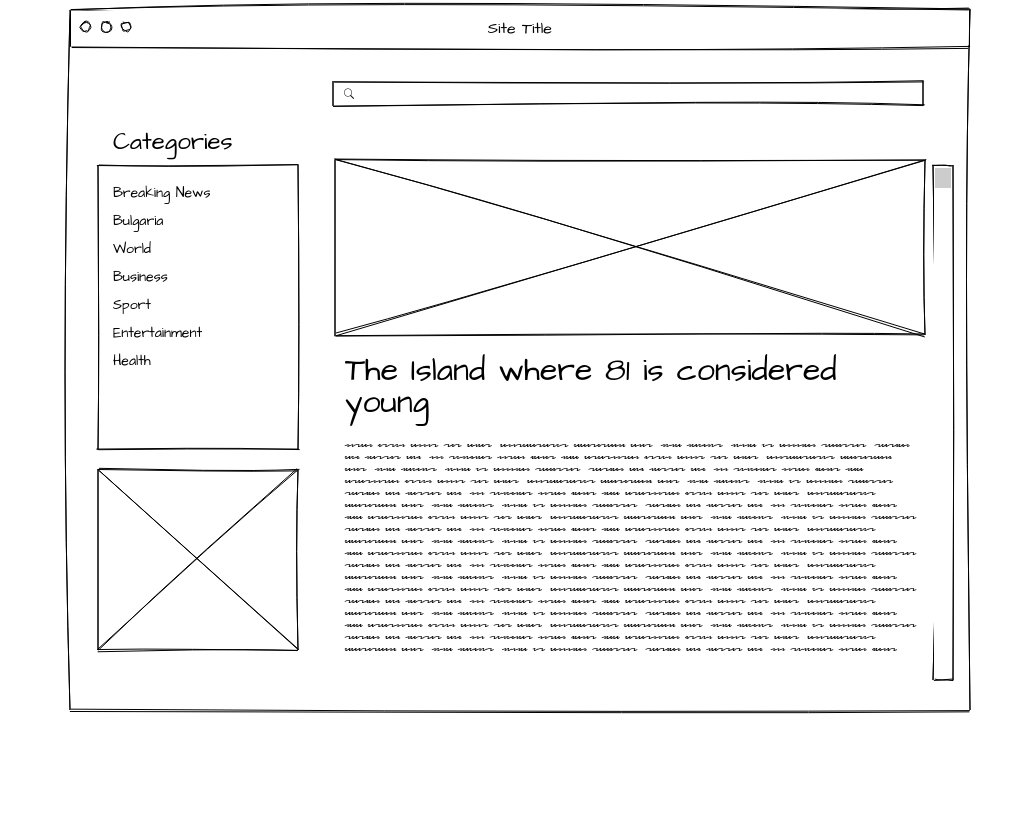
# ПРОЕКТИРАНЕ НА ПОТРЕБИТЕЛСКИЯ ИНТЕРФЕЙС

## Обобщение на потребителския интерфейс

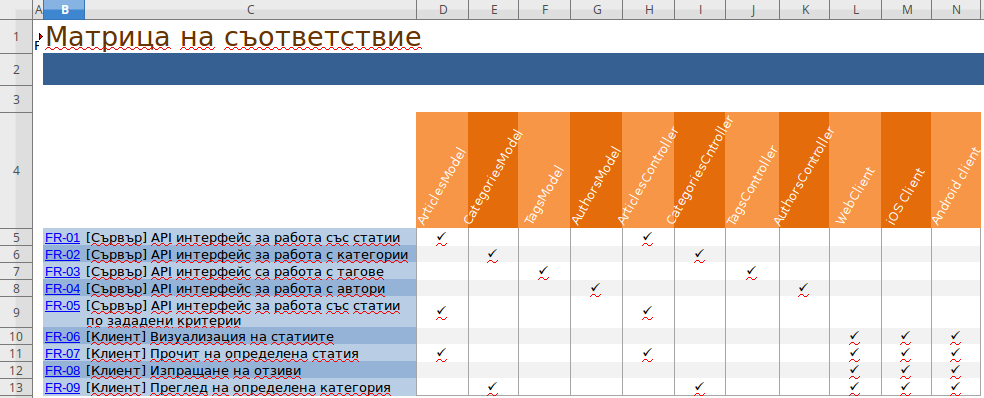
Нашите потребители могат да използват системата посредством клиентските приложения, които могат да консумират REST API интерфейса ни. По този начин те могат да извличат необходимите данни и да ги представят на потребителите посредством приятен и изчистен интерфейс. В рамките на този проект ще бъде изградено интернет приложение, което да извлича данни от REST сървъра и да ги предоставя на потребителите.

## Екранни изображения

На фиг. 7.2.1 се вижда прототип на клиентското приложение, което показва различните категории новини и списък със статиите от специфична категория. Съответно на фиг. 7.2 виждаме друг основен екран на клиентското приложение, който представлява прототип на екрана, когато е избрана дадена статия за прочит.

фиг. 7.2.1.фиг. 7.2

# МАТРИЦА НА ИЗИСКВАНИЯТА

**

фиг. 8.1 - матрица на съответствията между изискванията към системата

и компонентите и данновите структури

# ПРИЛОЖЕНИЯ

*Тази секция не е задължителна и се използва при необходимост.*

*Приложенията се включват директно или се реферират. Те подпомагат разбирането на системното проектиране, изложено в настоящия документ.*