Smart nation

*Software Requirements Specification*

**Утверждение документа**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Фамилия** | **Дата** | **Подпись** |
| Подготовлен | Белоха | 16/09/2017 |  |
| Подготовлен | Коваленко | 16/09/2017 |  |
| Подготовлен | Давидюк | 16/09/2017 |  |
| Проверен | Штогрина | 18/09/2017 |  |
| Контроль качества |  |  |  |
| Утверждён |  |  |  |

**Список доступа:**

В открытом доступе.

**История изменений**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Версия** | **Дата** | **Автор** | **Описание** | **Раздел** |
|  |  |  |  | **FUN** |

**Информация о документе**

|  |  |
| --- | --- |
| **Дата создания** | 16/09/2017 |
| **Имя файла** | **SN\_Technical\_doc** |
| **Местоположение** | Kiev, Ukraine |
| **Кол-во страниц** |  |

## 

## 

## **Введение**

Данный документ описывает функционал Веб приложения для обмена печатными книгами между людьми по всему миру.Нужно обеспечить пользователям возможности: выкладывать свои книги на сайт, искать доступные книги с учетом местоположения пользователя, отправлять заявки на доступные у других пользователей книги. Также необходимо обеспечить защиту от злоумышленников путем ограничения максимального количества книг которые можно взять при определенных условиях.

### **Ссылки**

Представленный документ ссылается на следующие документы.

●

### **Сокращения**

● Соц. сети - социальные сети.

### **Соглашение о терминах**

* Ресурсы - книги, газеты, статьи и документы, которые могут быть использованы для доступа к информации.
* Личный аккаунт - (далее аккаунт) профиль пользователя для доступа в систему.
* Личный кабинет - (далее кабинет) кабинет пользователя где хранятся его личные данные.
* Личная информация - (далее информация) личные данные о пользователе, предоставление пользователем.
* Злоумышленники - пользователи сайта, действия которых направлены на аккумуляцию книг в одних руках.
* Facebook - американская соц. сеть.
* Опционально - не обязательно.
* Владелец ресурса - (далее владелец) пользователь который первым внес ресурс в систему.

### **Структура документа**

Первый раздел описывает правила и методологии разработки.   
Второй раздел описывает функциональные и нефункциональные требования.   
Третий раздел всесторонне описывает варианты использования, с точки зрения того как модель структурирована на группы и какие варианты использования участников в модели.

# **Раздел 1: Разработка**

## Введение

В этой части описан:

* Команда
* Сроки
* Архитектура
* Процесс разработки
* Методология разработки
* Используемые языки программирования
* Используемые технологии
* Используемое ПО
* Дополнительные API
* Стандарты написания кода
* Соглашение о наименовании

## Архитектура

Клиент - Сервер - БД

## Методология разработки

Разработка ведется по методологии **Continuous Delivery**.

## Используемые языки программирования

## Используемые технологии

## Используемые ПО

## Дополнительное API

# **Раздел 2: Требования**

## **Обзор**

### **Введение**

Этот раздел описывает требования (функциональные и нефункциональные) системы.

### **Содержание**

Этот раздел содержит следующие темы

|  |  |
| --- | --- |
| Тема | См. стр. |
| Типы требований |  |
| Список функциональных требований |  |
| Список не-функциональных требований |  |

# **Типы требований**

## **Определения**

**Требование** определяется как "состояние или возможность, которым система должна соответствовать".

**Функциональные требования** определяют действия, которые система должна обеспечивать возможность выполнять, не принимая физических ограничений во внимание. Они часто лучше всего описаны в модели прецедентов (use-case model) и сценариев использования (use cases). Функциональные требования таким образом, определяют входное и выходное поведение системы.

Требования, которые не являются функциональными иногда называют **не-функциональными требованиями**. Многие требования не являются функциональными, и описывают только атрибуты системы или признаки окружения системы.

## **ФУНПП+ (FURPS+)**

Есть много различных видов требований. Один из способов их категоризации

описывается как ФУНПП+ (FURPS+) модель, используя акроним, чтобы описать

основные категории и подкатегории требований, как показано ниже.

● Функциональность (Functionality),

● Удобство использования (Usability),

● Надежность (Reliability),

● Производительность (Perfomance),

● Поддерживаемость (Supportability)

**"+"** помогает Вам также не забыть включать такие требования, как

● конструктивные ограничения,

● требования к интерфейсам,

● физические требования.

## **Функциональность (FUN)**

Функциональные требования могут включать в себя:

● наборы функций,

● возможности,

● безопасность.

## **Удобство использования (USA)**

Требования удобства использования могут включать такие подкатегории, как:

● человеческий фактор,

● эстетика,

● последовательность пользовательского интерфейса,

● он-лайн и контекстно-зависимая справка,

● мастера и агенты,

● пользовательская документация,

● учебные материалы.

## **Надежность (REL)**

Требования к надежности, которые будут рассмотрены:

● доступность (в процентах от имеющегося времени, часов эксплуатации, техническому обслуживанию доступа, ...)

● частота / степень интенсивности отказов,

● восстанавливаемость,

● предсказуемость,

● точность,

● среднее время наработки на отказ (MTBF).

## **Производительность (PER)**

Требования к производительности накладывают условия на функциональные требования.Например, для указанного действия, они могут указать параметры производительности для:

● пропускной способности (например, транзакций в секунду),

● время отклика,

● время восстановления,

● использование ресурсов (память, диск, процессор, ...).

## **Поддерживаемость (SUP)**

Требования поддерживаемости могут включать:

● тестируемость,

● расширяемость,

● адаптивность,

● ремонтопригодность,

● совместимость,

● конфигурируемость,

● удобство обслуживания,

● устанавливаемость,

● локализацию (интернационализацию).

## **Конструктивные требования (DES)**

Конструктивные требования, которые часто называют ограничения дизайна, указывают или ограничивают конструкцию системы.

В этом разделе должны быть указаны любые ограничения проектирования в строящейся системе. Ограничения дизайна представляют собой конструктивные решения, которые введены и должны соблюдаться. Примеры включают в себя языки программирования, требования к процессу разработки программного обеспечения, предписанного использования инструментов разработки, архитектурные и конструктивные ограничения, покупные комплектующие, библиотеки классов и т.д.

## **Требования к интерфейсам (INT)**

Этот раздел определяет интерфейсы, которые должны поддерживаться приложением. Она должна содержать адекватную спецификацию, протоколы, порты и логические адреса и т.д., такую, что программа может быть разработана и проверена с учетом требований к интерфейсу.

Требования к интерфейсу можно разделить на:

● Пользовательский интерфейс (пользовательские интерфейсы, которые должны быть реализованы с помощью программного обеспечения)

● Аппаратный интерфейс (аппаратные интерфейсы, которые должны поддерживаться программным обеспечением, в том числе логическую структуру, физические адреса и ожидаемое поведение, и т.д.)

● Программный интерфейс (программные интерфейсы для других компонентов системы. Они могут быть приобретенными, компонентами повторно используемыми из другого приложения, компонентами разрабатываемыми для подсистемы, которая выходит за рамки этого проекта, но с которой это приложение должно взаимодействовать).

## **Физические требования(HAR)**

Физические требования определяют физические характеристики, которыми система должна обладать;например:

● материал

● форма

● размер

● вес

Этот тип требований может быть использован для представления аппаратных требований, таких как необходимая физическая конфигурация сети.

## **Применимые стандарты (STD)**

В этом разделе описываются любые применимые стандарты и конкретные разделов таких стандартов, которые применяются к описываемой системе. Например правовые, нормативные, отраслевые стандарты, стандарты качества,

совместимости, локализации, соответствия операционным системам и т.д.

# **Список функциональных требований**

## **Введение**

Функциональные требования определяют действия, которые система должна обеспечивать возможность выполнять, не принимая физических ограничений во внимание. Они часто лучше всего описаны в модели прецедентов (use-case model) и сценариев использования (use cases). Функциональные требования таким образом, определяют входное и выходное поведение системы.

## **Список функциональных требований**

В таблице ниже перечислены все функциональные требования:

|  |  |  |
| --- | --- | --- |
| **Идентификатор функционального требования** | **Описание** | См. страницу |
|  | Новости на сайте. | new |
| FUN-01 | Регистрация пользователя. |  |
| FUN-02 | Управление кабинетом. |  |
|  | Привязка facebook к аккаунту. |  |
|  | Добавление ресурса в систему. |  |
|  | Управление текущими ресурсами. |  |
|  | Состояние непригодный к использованию ресурс. |  |
| FUN-03 | Поиск ресурсов среди всех пользователей в системе. |  |
|  | Результат поиска ресурса в системе. |  |
|  | Карта результата поиска ресурсов. |  |
| FUN-04 | Заявка на заказ ресурса у подходящего пользователя. |  |
| FUN-05 | Связь между пользователями системы. |  |
|  | Передача ресурса. |  |
| FUN-06 | Система штрафов пользователя. |  |
|  | Система бонусов пользователя. |  |
|  | Система ограничений. |  |
| FUN-07 | Рейтинг пользователя. |  |
| FUN-08 | Возможность оставлять комментарии пользователям. |  |
| FUN-09 | Система рецензий книг. |  |
|  | Выбор категории ресурса. |  |
|  | Выбор типа ресурса. |  |
|  | Добавление фотографий ресурса. |  |
|  | Назначение цены ресурса. |  |

# 

### Регистрация пользователей.

Пользователь должен иметь возможность зарегистрироваться в веб-приложении.

Регистрация с помощью facebook, либо регистрация на сайте.

Регистрация на сайте происходит в 2 этапа:

* Ввод номера личного телефона, пароля и подтверждение пароля.
* Подтверждение номера личного телефона.

Регистрация с помощью facebook подразумевает, что после аутентификации в facebook, аккаунт автоматически привязывается к facebook-аккаунту.

### Управление кабинетом.

Пользователь должен иметь возможность управлять кабинетом, чтобы:

* Добавлять информацию.
* Изменять информацию.
* Удалять аккаунт.
* Восстанавливать аккаунт.
* Привязывать facebook к аккаунту.
* Добавлять часы телефонной связи с другими пользователями.
* Изменять часы телефонной связи с другими пользователями.

### Привязка facebook к аккаунту.

Пользователь привязывает facebook к аккаунту путем ввода логина и пароля в офф. соответствующее API от facebook.

### Добавление ресурса в систему.

Аутентифицированный пользователь должен иметь возможность добавлять ресурсы в систему.

Аутентифицированный пользователь переходит на форму добавление ресурса, вносит имя ресурса ( одновременно ему предлагается список уже имеющихся в системе имен книг или книг существующих в каталоге интернет-магазина Amazon), выбирает категорию, выбирает тип, добавляет фотографии (опционально), назначает цену (опционально).

### Управление текущими ресурсами.

Пользователь должен иметь возможность управлять ресурсами, которые находятся у него в текущий момент времени, а именно:

* Изменять статус ресурса (ресурс в чтении, в ожидании)
* Изменять состояние ресурса (ресурс утерян, отдан, непригодный к использованию)

### Состояние непригодный к использованию ресурс.

Пользователь должен иметь возможность изменить состояние ресурса на непригодный к использованию.

При выборе данного действия пользователь должен подтвердить непригодное для использование состояние ресурса путем предоставлений соответствующих фотографий.   
Администрация сайта либо подтверждает непригодное состояние к использованию, либо отклоняет его.

### Поиск ресурсов среди всех пользователей в системе.

Пользователи должны иметь возможность осуществить поиск определенного ресурса в системе по следующим критериям:

* Название
* Автор
* Год выпуска
* Категория
* Популярность
* Местоположение от которого следует отталкиваться. По-умолчанию поиск должен осуществляться от текущего местоположения.

Также пользователь должен иметь возможность комбинировать данные критерии поиска для более точного результата поиска.

### Результат поиска ресурса в системе.

Пользователь должен иметь возможность получить результат поиска в табличном формате в левой части пользовательского интерфейса :

* Название ресурса
* Имя автора
* Расстояние до ресурса
* Цена за ресурс (если цена указана)

Пользователь должен иметь возможность:

* Перейти на страницу ресурса, где сможет изучить содержание ресурса (если оно есть), посмотреть рецензии\отзывы (если они присутствуют)
* Перейти на страницу автора, где сможет увидеть ...
* Заказать ресурс

Также, пользователь должен получить возможность увидеть в правой части экрана карту, где будут указаны ресурсы.

В случае отсутствия результатов поиска берется следующий масштаб местоположения, в случае если результаты снова отсутствуют, то снова следующий масштаб местоположения и так до момента максимально возможного. В случае, если максимально возможный масштаб местоположения не дал результатов, то вместо таблицы выводиться сообщение, что поиск не дал результатов, а на карте верхним слоем показывается соответствующее сообщение.

### Карта результата поиска.

Пользователь должен иметь возможность увидеть результат поиска ресурсов на карте.   
  
Также, пользователь должен иметь возможность изменять отображение результата поиска на карте:

* Изменять масштаб карты, что приведет к измененному отображению ресурсов как на карте, так и в таблице.
* Перемещаться по карте, что приведет к измененному отображению ресурсов как на карте, так и в таблице.

По-умолчанию отображаются на карте и показываются в таблице ресурсы, находящихся в одном районе с пользователем.

### Заявка на заказ ресурса у подходящего пользователя.

Авторизованный пользователь должен иметь возможность заказать ресурс кликнув по нему в таблице результатов, либо на карте.

После клика пользователю заказавшему ресурс показываются контактные данные владельца ресурса, а владельцу отправляется уведомление на почту о новой заявке на передачу ресурса.

### Передача ресурса.

После того как пользователь оставил заявку на получение какого-то ресурса у него по

### Связь между пользователями системы.

Пользователи системы должны иметь возможность осуществлять связь между собой.   
У всех пользователей, автоматически показывается почта.   
  
При активной передачи ресурса у пользователя и у владельца показываются мобильные телефоны друг друга и показываются, причем, только им.

**Добавление фотографий ресурса(опционально)**

### Назначение цены ресурса (опционально, по умолчанию 0$)

Цена ресурса назначается владельцем в пределах от 0% до 50% от номинальной на данный момент стоимость данной книги на сайте интернет магазина Amazon.

В случае отсутствия данного названия книги в базе пользователь может назначить цену самостоятельно.

# **Список не-функциональных требований**

## **Введение**

Не-функциональные требования описывают только атрибуты системы или признаки окружения системы.

Каждому определенному не-функциональному требованию присваивается уникальный ключ "XXX-nn", где XXX-аббревиатура типа требования, nn-порядковый номер, определяющий не-функциональное требование.

## **Требования удобства использования (USA)**

В таблице ниже перечислены все требования удобства использования:

|  |  |
| --- | --- |
| **Идентификатор требования удобства** | **Описание** |
| USA-01 | Простота интерфейса |
| USA-02 | Хорошая навигация (Возможность перехода на любую страницу сайта в 3 клика.) |
| USA-03 | Справка и рекомендации по использованию веб-приложения |
| USA-04 |  |
| USA-05 |  |

## 

## **Надежность (REL)**

В таблице ниже перечислены все требования надежности:

|  |  |
| --- | --- |
| **Идентификатор требования надежности** | **Описание** |
| REL-01 | Гарантия работы серверов 99% |
| REL-02 | Бэкап данных |
| REL-03 |  |
|  |  |
|  |  |

## **Производительность (PER)**

В таблице ниже перечислены все требования производительности:

|  |  |
| --- | --- |
| **Идентификатор требования**  **производительности** | **Описание** |
| PER-01 | Скоростное интернет соединение(>10Мбит/сек) |
|  |  |
|  |  |
|  |  |
|  |  |

## **Поддерживаемость (SUP)**

В таблице ниже перечислены все требования поддерживаемости:

|  |  |
| --- | --- |
| **Идентификатор требования**  **поддерживаемости** | **Описание** |
| SUP-01 | Удобство Администрирования |
| SUP-02 | Расширяемость функционала |
| SUP-03 | Cоздание/восстановление бекапа. |
| SUP-04 | Масштабируемость: географическая; клиент-ориентированная; по мощности |
| SUP-05 |  |
| SUP-06 |  |
| SUP-07 |  |
|  |  |

**SUP-01**

Система имеет пользовательские интерфейсы администрирования системы.

## **Конструктивные требования (DES)**

В таблице ниже перечислены все требования к дизайну:

|  |  |
| --- | --- |
| **Идентификатор требования к дизайну** | **Описание** |
| DES-01 | Clean-Code |
| DES-02 | Unit Testing |
| DES-03 | Использование CSS, HTML5, JS, ASP.NET. |
| DES-04 |  |

## **Требования к интерфейсам (INT)**

В таблице ниже перечислены все требования к интерфейсам:

|  |  |
| --- | --- |
| **Идентификатор требования к интерфейсам** | **Описание** |
| INT-01 | Понятность и логичность пользовательского интерфейса. |
| INT-02 | Гармоничная цветовая палитра пользовательского интерфейса. |
| INT-03 |  |
| INT-04 |  |
| INT-05 |  |
| INT-06 |  |
| INT-07 |  |
| INT-08 |  |

## **Совместимость (COM)**

В таблице ниже перечислены все требования совместимости:

|  |  |
| --- | --- |
| **Идентификатор требования**  **совместимости** | **Описание** |
| COM-01 | Совместимость с браузерами которые поддерживают технологии HTML5, CSS3. |
| COM-02 |  |
| COM-03 |  |
| COM-04 |  |

## **Безопасность (SEC)**

В таблице ниже перечислены все требования безопасности:

|  |  |
| --- | --- |
| **Идентификатор требования**  **безопасности** | **Описание** |
| SEC-01 | Разграничения прав доступа пользователей. |
| SEC-02 | Сокрытие точного местоположения пользователей. |
| SEC-03 | Верификация номера телефона и электронной почты |
| SEC-04 |  |

**Применимые стандарты (STD)**

В таблице ниже перечислены все применимые стандарты:

|  |  |
| --- | --- |
| **Идентификатор требования стандартов** | **Описание** |
| STD-01 |  |
| STD-02 |  |
| STD-03 |  |
| STD-04 |  |

# 

# 

# **Раздел 3: Варианты использования**

Например: Программа может быть использована как студентами в процессе обучения, так и профессиональными охранными организациями для обеспечения централизованного контроля удаленных устройств.