|  |  |
| --- | --- |
| УТВЕРЖДАЮ  Технический консультант  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.Д.Каташевцев  \_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г. | УТВЕРЖДАЮ  студентка гр. АСУб-13-1 ИРНИТУ  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ю.В. Комогорцева  \_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г. |

|  |
| --- |
| ИРНИТУ, институт ИТиАД, студентка группы ИСТб-19-2 |

наименование организации – разработчика ТЗ на АС

|  |
| --- |
| «Площадка для нетворкинга начинающих IT специалистов для повышения уровня компетенций в командной разработке» |

наименование вида АС

|  |
| --- |
| «Биржа опыта» |

сокращенное наименование АС

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

На \_21\_ листах

Действует с 18.11.2022 г.

|  |
| --- |
| СОГЛАСОВАНО  Научный руководитель  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Бахвалова З.А.  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г. |

**Оглавление**

[1 Общие сведения 3](#_Toc124012631)

[1.1 Полное наименование системы 3](#_Toc124012632)

[1.2 Краткое наименование системы 3](#_Toc124012633)

[2 Назначение и цели создания системы 3](#_Toc124012634)

[2.1 Назначение системы 3](#_Toc124012635)

[2.2 Цели создания системы 3](#_Toc124012636)

[3. Характеристика объектов автоматизации 4](#_Toc124012637)

[3.1. Описание целевой аудитории 6](#_Toc124012638)

[4. Требования к системе 7](#_Toc124012639)

[4.1. Требования к системе в целом 7](#_Toc124012640)

[4.1.1. Требования к структуре 7](#_Toc124012641)

[4.1.2. Требования к функционированию системы 8](#_Toc124012642)

[4.1.3. Требования к численности и квалификации персонала системы и режиму его работы 9](#_Toc124012643)

[4.1.4. Требования к надежности 9](#_Toc124012644)

[4.1.5. Требования к структуре сайта 10](#_Toc124012645)

[4.1.5. Требования к эргономике и технической эстетике 10](#_Toc124012646)

[4.1.6. Требования к верстке 10](#_Toc124012647)

[4.1.7. Требования к защите информации 11](#_Toc124012648)

[4.1.8. Требования по сохранности информации при авариях 11](#_Toc124012649)

[4.1.9. Требования по стандартизации и унификации 11](#_Toc124012650)

[4.1.10. Требования к безопасности 12](#_Toc124012651)

[4.2. Требования к функциям, выполняемым системой 12](#_Toc124012652)

[4.2.1. Подсистема Проекты 12](#_Toc124012653)

[4.2.2. Подсистема Хакатоны 13](#_Toc124012654)

[4.2.3. Подсистема Мероприятия 13](#_Toc124012655)

[4.2.4 Подсистема Личный кабинет 13](#_Toc124012656)

[4.2.5 Подсистема Административная панель 14](#_Toc124012657)

[4.3. Требования к ролям и правам доступа 14](#_Toc124012658)

[4.4. Требования к видам обеспечения 17](#_Toc124012659)

[4.4.1. Требования к информационному обеспечению 17](#_Toc124012660)

[4.4.2. Требования к лингвистическому обеспечению 18](#_Toc124012661)

[4.4.3. Требования к техническому обеспечению 18](#_Toc124012662)

[4.4.4. Требования к организационному обеспечению 18](#_Toc124012663)

[4.4.5. Требования к патентной чистоте 18](#_Toc124012664)

[5. Состав и содержание работ по созданию системы 19](#_Toc124012665)

[6. Требования к документированию 20](#_Toc124012666)

# 1 Общие сведения

## 1.1 Полное наименование системы

Полное наименование: Площадка нетворкинга начинающих IT специалистов для повышения уровня компетенций в командной разработке.

## 1.2 Краткое наименование системы

Краткое наименование: Система, площадка.

# 2 Назначение и цели создания системы

## 2.1 Назначение системы

Система предназначена для увеличения информированности начинающих IT специалистов о возможностях получения первичного опыта. Основным назначением системы является предоставление информации о проектах и хакатонах, а также объединение людей в команды. Система должна предоставлять возможность вступать в команды для получения опыта разработки. Так же система должна обеспечивать возможность хранения информации о пользователе для лучшего взаимодействия между пользователями. Система должна мотивировать пользователей на участие с помощью соревнований и системы достижений.

В рамках проекта автоматизируется деятельность в следующих бизнес-процессах:

* + поиск проекта для получения навыков с помощью работы в команде;
  + поиск хакатона для получения навыков разработки через «соревновательный дух»;
  + поиск команды для совместной работы;
  + проведение мероприятий направленных на повышение компетенций командной деятельности.

## 2.2 Цели создания системы

Глобальная цель системы: повысить уровень компетенций в командной разработке с помощью нетворкинга.

Система создается с целью **сопровождения процесса повышения уровня компетенций в командной разработке посредством нетворкинга.**

В результате создания системы должны быть улучшены значения следующих показателей:

* время, затрачиваемое на приобретение полезных связей в IT сфере;
* время поиска команды/участников в команду;
* время поиска проектов;
* время поиска хакатонов;
* время поиска мероприятий;
* время осваивания специалистов в новом коллективе;
* время принятия решения при принятии на работу;
* время поиска работы.

# 3. Характеристика объектов автоматизации

Словарь, необходимый для понимания предметной области:

Пет-проект — это личный проект, который разработчик делает в свободное время.

Хакатон — мероприятие для разработчиков, во время которого специалисты из разных областей разработки программного обеспечения (программисты, дизайнеры, менеджеры) сообща решают какую-либо проблему на время.

Нетворкинг — это расширение сети знакомств для решения профессиональных и личных задач с их помощью.

Процессы, которые планируется автоматизировать в данный момент, выглядят следующим образом:

*Поиск пет-проекта*:

1. необходимо найти статьи с рекомендациями о пет-проектах для своей области IT и ознакомиться с ними;
2. прочитать форумы или обсуждения в социальных сетях;
3. подобрать идею для пет-проекта исследуя потребности знакомых;
4. выбрать из готовых предложенных или генерировать самому идею.

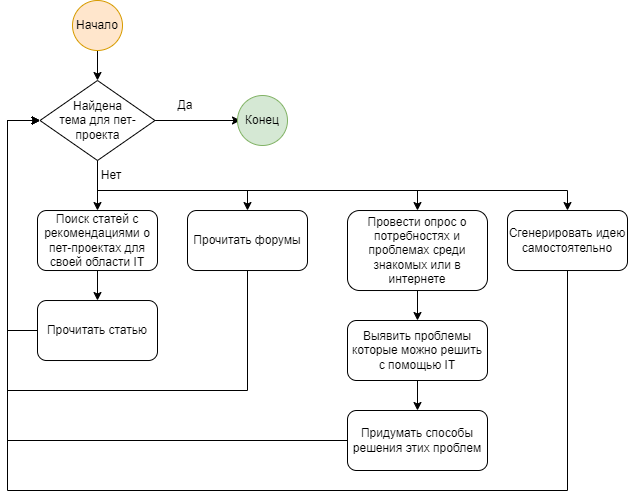


Рисунок 1 – Алгоритм поиска пет-проекта

*Поиск хакатона:*

1. найти ресурсы с хакатонами (иногда это группы в социальных сетях или чаты компаний, занимающихся организацией хакатонов);
2. поиск хакатона с помощью нетворкинга;
3. отслеживать подходящий по навыкам и интересам хакатон.

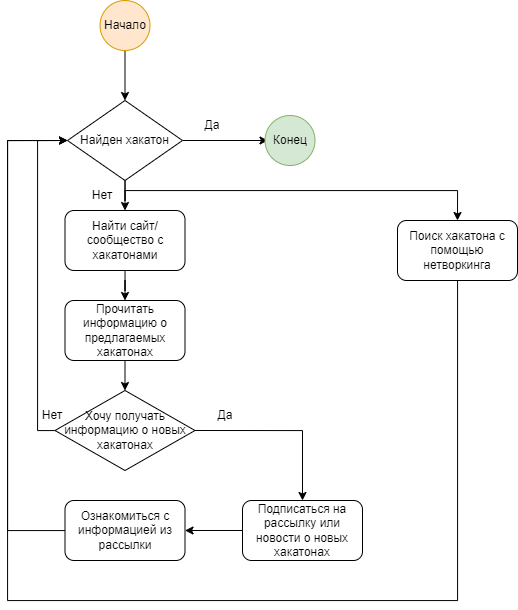


Рисунок 2 – Алгоритм поиска хакатона

*Поиск команды для проекта или хакатона:*

1. поиск через платформу, на которой проводится хакатон;
2. поиск через социальные сети (объявления, чаты);
3. поиск с помощью нетворкинга (знакомые, мероприятия, конференции).

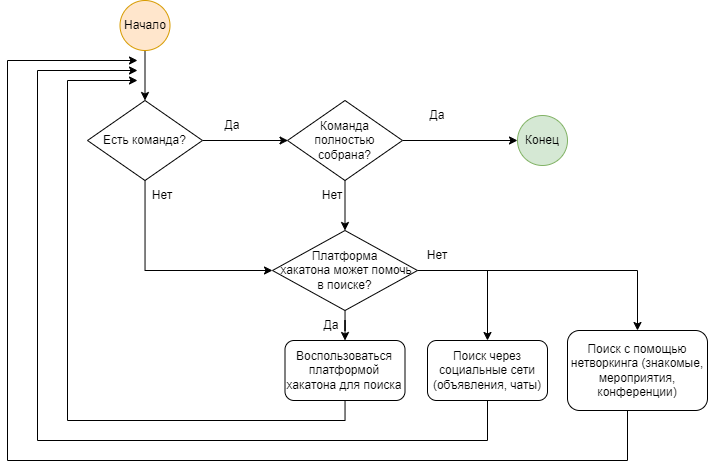


Рисунок 3 – Алгоритм поиска команды для проекта или хакатона

К сожалению, каждый из процессов занимает немало времени, выполняется много ненужных действий. В основе всего лежат статьи, соц. сети и сайты с хакатонами. Источники информации для каждого процесса разные, что является существенным недостатком.

Часто, несмотря на желание человека получить опыт, дальше ознакомления информации он не заходит. Чтобы решить эту проблему, необходимо учитывать систему мотивации. Мотивацию необходимо предусмотреть с помощью конкурсов и челленджей на платформе, а также системы достижений, правила выдачи которых необходимо определить на этапе проектирования.

## 3.1. Описание целевой аудитории

Общее описание:

* мужчины и женщины;
* от 10 до 40 лет;
* в процессе обучения или только начали карьеру в IT;
* стажер, джуниор, мидл.

Ключевые ценности: интересные задачи, профессиональный рост, современные технологии и отношения в команде.

Психологический портрет: интроверт, спокойный, флегматичный, технический, прагматичный склад ума.

Таблица 1 – Матрица ценностей

|  |  |  |
| --- | --- | --- |
| **Категория** | **Сценарий** | **Ценность** |
| Человек без опыта работы, участия в хакатонах и проектах | Подбирает себе нужный проект или хакатон | Получил первичный опыт |
| Человек с небольшим опытом, проактивный или имеющий идею для стартапа | Создает свой проект, набирает команду, реализовывает проект,  отмечает результат | Получил опыт и проект, возможно, прибыль |
| Человек с опытом, ищет интересные проекты для собственного развития | Подбирает нужный проект и команду | Получил удовольствие и опыт |

# 4. Требования к системе

## 4.1. Требования к системе в целом

### 4.1.1. Требования к структуре

В Системе предлагается выделить следующие функциональные подсистемы:

* подсистема хакатонов, которая предназначена для реализации процесса поиска/подбора хакатона, а также поиска команды или участника. Подсистема будет собирать информацию с помощью агрегации с других сайтов информации о хакатонах;
* подсистема проектов, которая должна предоставлять возможность подбора идей пет-проектов и сообщников для реализации пет-проектов. Подсистема подразумевает хранение информации о результате участия в проекте каждого участника;
* подсистема мероприятий, которая включает в себя создание конкурсов, проведение мероприятий и отслеживание успехов пользователей для мотивации получения опыта;
* подсистема личного кабинета, которая предназначена для формирования информации о пользователе, его опыте и достижениях, для подбора пет-проектов и хакатонов. Подсистема включает в себя так же механику достижений, а именно назначение достижений;
* подсистема администрирования, которая помогает управлять проектами, агрегатором и контента проекта в целом. Подсистема должна быть доступна только для администратора, и нужна для модерации поданных заявок, учета статистики платформы.

Система должна быть web-приложением. Благодаря этому, возможно обеспечить доступ из различных устройств из любой точки мира. Система должна состоять из трех приложений в виде сервера, агрегатора и клиента. Клиент должен содержать программу-клиент, которая предоставляет для пользователя интерфейс, посылает запросы на сервер и получает ответы на запросы. В то же время, сервер должен хранить данные и содержит программу, которая отвечает за логику обработки данных из поступивших запросов. Агрегатор должен взаимодействовать с сайтами-источниками и запускаться сервером.

Источниками данных для подсистемы агрегатора хакатонов должны быть:

* <https://leader-id.ru/>
* <https://hackathons.pro/>
* <https://www.хакатоны.рф/>
* <https://codenrock.com/>
* <https://pgenesis.ru/events>
* <https://devpost.com/>
* <https://хакатоны.рус/>

Данный перечень сможет редактироваться и пополняться после реализации подсистемы администрирования, а именно, управления агрегатором. В данном разделе подсистемы необходимо предоставлять возможность пользователю просматривать список источников, редактировать и добавлять новые источники.

### 4.1.2. Требования к функционированию системы

Система должна поддерживать следующие режимы функционирования:

* основной режим, в котором подсистемы выполняют все свои основные функции;
* профилактический режим, в котором одна или все подсистемы не выполняют своих функций.

В *основном режиме функционирования* площадка должна обеспечивать: доступность для пользователя в режиме – 24 часа в день, 7 дней в неделю (24х7) и выполнение своих функций.

В *профилактическом режиме* система должна обеспечивать возможность проведения: технического обслуживания и модернизацию аппаратно-программного комплекса.

Общее время проведения профилактических работ не должно превышать 2 дней.

Обеспечение приспособляемости системы к изменениям должно выполняться за счет:

* своевременности администрирования;
* модернизации процессов системы;
* модификации процедур доступа и представления данных конечным пользователям.

### 4.1.3. Требования к численности и квалификации персонала системы и режиму его работы

Для обеспечения работы площадки необходим администратор проекта. На начальных этапах развития площадки достаточно одного человека.

Администратор проекта на всем протяжении функционирования системы обеспечивает контроль, модерацию и управление системой, заполняет контентом и управляет агрегатором.

К квалификации администратора проекта, предъявляются следующие требования: знание предметной области - IT, понимание технологий агрегации информации, владение технологиями, которыми реализована система.

Администратор должен работать в следующих режимах: может заниматься модерацией соответствие с основным рабочим графиком, по желанию и в дополнительное свободное время.

### 4.1.4. Требования к надежности

4.1.4.1. Состав показателей надежности для системы в целом

Надежность должна обеспечиваться за счет:

* применения технических средств, системного и базового программного обеспечения, соответствующих классу решаемых задач;
* своевременного выполнения процессов администрирования Системы.

Система должна соответствовать следующим параметрам:

* среднее время восстановления 48 часов - определяется как сумма всех времен восстановления за заданный календарный период, поделенные на продолжительность этого периода;
* коэффициент готовности W - определяется как результат отношения средней наработки на отказ к сумме средней наработки на отказ и среднего времени восстановления;
* время наработки на отказ E часов - определяется как результат отношения суммарной наработки Системы к среднему числу отказов за время наработки.

#### 4.1.4.2. Требования к надежности технических средств и программного обеспечения

Надежность программного обеспечения подсистем должна обеспечиваться за счет:

* надежности общесистемного ПО и ПО, разрабатываемого Разработчиком;
* проведением комплекса мероприятий отладки, поиска и исключения ошибок.
* ведением журналов системных сообщений и ошибок по подсистемам для последующего анализа и изменения конфигурации.

### 4.1.5. Требования к структуре сайта

Структура сайта и основные компоненты определены и доступны по следующей ссылке: <https://octopus.do/nwi79o50tnl> .

Действия обычного пользователя начинаются с главной страницы, где содержатся краткие сведения о проекте. Далее с помощью навигационного меню пользователь может перейти на страницу одной из подсистем (хакатоны, проекты, конкурсы) или зарегистрироваться/авторизоваться.

Действия администратора проекта начинаются с главной страницей со статистикой, и с помощью навигационного меню администратор может перейти на страницы с управлением проектами, хакатонами, конкурсами.

### 4.1.5. Требования к эргономике и технической эстетике

Площадка должна обеспечивать удобный для конечного пользователя интерфейс, отвечающий следующим требованиям.

В части внешнего оформления:

* интерфейсы подсистем должен быть типизированы;
* должно быть обеспечено наличие локализованного (русскоязычного) интерфейса пользователя.

В части диалога с пользователем при возникновении ошибок в работе подсистемы должно выводиться сообщение с наименованием ошибки и с рекомендациями по её устранению на русском языке.

### 4.1.6. Требования к верстке

Система должна отвечать требованиям **кроссбраузерности**, а именно, поддерживаться во всех популярных браузерах в последних версиях:

* Internet Explorer версии 10 и выше, платформа — операционные системы семейства Windows;
* Mozilla Firefox версии 28 и выше, платформа — Windows версии XP и выше и Mac OS X версии 10.8 и выше;
* Safari версии 6.1 и выше, платформа — Mac OS X версии 10.8 и выше;
* Google Chrome версии 21 и выше, платформа — Windows версии XP и выше и Mac OS X версии 10.8 и выше;
* Opera версии 15 и выше, платформа — Windows версии XP и выше и Mac OS X версии 10.8 и выше;
* браузеры мобильных устройств iOS 7 и выше;
* браузеры мобильных устройств Android 5 и выше.

При верстке должны учитываться особенности браузеров, их ограничения внешнего вида и поведения объектов. Для обеспечения правильного отображения элементов в разных браузерах может применяться грациозная деградация.

Система должна разрабатываться в соответствии с современным методом верстки **БЭМ** – «блок – элемент – модификатор». Необходимо соблюдать единые правила верстки, которые помогают быстро разрабатывать интерфейсы, гибко их настраивать и легко модифицировать.

Верстка в системе должна быть **адаптивной**, а именно корректно отображаться на любом устройстве со следующими расширениями: 2560x1440; 3840x2160; 1024x600; 1024x768; 1152x864; 1280x800; 1280x1024; 1440x900; 1920x1080; 1680x1050.

Остальные требования:

* при размерах более 375px не должно возникать горизонтальной прокрутки страницы (за исключением отдельных блоков верстки, где подобная реализация предусмотрена дизайн-макетами);
* все изображения должны сопровождаться описанием при проблемах загрузки;
* логотип на внутренних страницах должен вести на главную;
* у каждой страницы должен быть свое уникальное поле title;
* изображения должны масштабироваться в зависимости от размера окна;
* картинки должны быть максимально сжаты, должны использоваться современные форматы изображений;
* весь важный функционал сайта должен быть доступен без js-скриптов;
* анимация не должна быть сложной.

### 4.1.7. Требования к защите информации

Обеспечение информационной безопасности платформы должно удовлетворять следующим требованиям:

* защита Системы должна обеспечиваться комплексом программно-технических средств и поддерживающих их организационных мер;
* защита Системы должна обеспечиваться на всех технологических этапах обработки информации и во всех режимах функционирования, в том числе при проведении ремонтных и регламентных работ;
* программно-технические средства защиты не должны существенно ухудшать основные функциональные характеристики Системы (надежность, быстродействие, возможность изменения конфигурации);
* разграничение прав доступа пользователей и администраторов Системы должно строиться по принципу «что не разрешено, то запрещено».

### 4.1.8. Требования по сохранности информации при авариях

В Системе должно быть обеспечено резервное копирование данных.

### 4.1.9. Требования по стандартизации и унификации

Разработка системы должна осуществляться с использованием стандартных методологий функционального моделирования: IDEF0, DFD в рамках рекомендаций по стандартизации Р50.1.028-2001 «Информационные технологии поддержки жизненного цикла продукции. Методология функционального моделирования».

Для работы с БД должен использоваться язык запросов SQL в рамках стандарта ANSI SQL-92.

В системе должны использоваться (при необходимости) общероссийские классификаторы и единые классификаторы и словари для различных видов алфавитно-цифровой и текстовой информации.

### 4.1.10. Требования к безопасности

Система должна обеспечивать:

* предотвращение несанкционированного доступа к информации и (или) передачи ее лицам, не имеющим права на доступ к информации;
* пароли от учетных записей должны храниться в зашифрованном виде.

## 4.2. Требования к функциям, выполняемым системой

Система должна включать 5 подсистем, определенных в пункте [4.1.1](#_4.1.1._Требования_к). В данном разделе описана детализация подсистем с описанием функциональности.

Форма представления выходной информации для всех функций определена структурой сайта.

### 4.2.1. Подсистема Проекты

Данная подсистема должна автоматизировать процесс поиска команды или идеи для пет-проекта. Функции, которые должны быть обязательно реализованы в первой версии:

* просмотр списка проектов с помощью фильтрации по тегам, по типу проектов;
* поиск проектов с помощью поисковой строки;
* просмотр информации о проекте: название; описание; изображения; тип проекта (уникальный/учебный); список команд, которые его выполняют/выполнили;
* отправление заявки на создание проекта администратору;
* отправление заявки на вступление в команду для выполнения конкретного проекта;
* рассмотрение заявки на вступление в команду создателем команды или любым из участников (при создании команды необходимо выбрать роль участника, который будет рассматривать заявки);
* фиксация результатов деятельности команды по проекту в виде прикрепления ссылки на готовое решение и написания отзыва о проделанной работе.

Функции, которые необходимо реализовать в следующих версиях:

* оценка результата деятельности команды другими участниками с помощью выставления оценки «нравится/не нравится»;
* рулетка проектов – альтернативный инструмент подбора проектов.

### 4.2.2. Подсистема Хакатоны

Данная подсистема должна автоматизировать процесс поиска хакатона и команды или участников в команду. Функции, которые должны быть обязательно реализованы в первой версии:

* просмотр списка хакатонов с помощью фильтрации по тегам, по датам начала регистрации, по размеру команды;
* просмотр информации о хакатоне: описание, даты, описание приза;
* переход на другой ресурс для записи на хакатон;
* агрегация хакатонов со сторонних ресурсов с помощью библиотек Python для парсинга веб-сайтов.

### 4.2.3. Подсистема Мероприятия

Данная подсистема должна мотивировать участников на выполнение каких-либо требований с целью получения приза. Мероприятия создаются администратором или могут быть предложены участниками. Функции, которые должны быть обязательно реализованы в первой версии:

* просмотр списка мероприятий;
* участие в мероприятии (функции аналогичны участию в проекте из п. [4.2.1](#_4.2.1_Подсистема_Проекты));
* просмотр результатов мероприятия (отображение победителей на странице мероприятия).

Функции, которые необходимо реализовать в следующих версиях:

* доски со списком лидеров по достижениям;
* проведение мероприятия - хакатона с помощью системы;
* оценка результатов команд, экспертами с помощью выставления оценок.

### 4.2.4 Подсистема Личный кабинет

Данная подсистема должна хранить информацию о пользователях и с ее помощью пользователь будет получать доступ ко всему с ним связанному. Функции, которые должны быть обязательно реализованы в первой версии:

* регистрация с помощью email, пароля и логина пользователя;
* авторизация с помощью email и пароля;
* восстановление пароля с помощью отправления письма на почтовый адрес;
* заполнение профиля после регистрации: информация о пользователе, контактная информация;
* просмотр личного кабинета: информация о профиле, список достижений пользователя;
* просмотр команд, в которых состоит участник, в личном кабинете;
* просмотр проектов, которые создавал участник, в личном кабинете;
* переход к странице проекта из списка проектов личного кабинета;
* переход к странице команды из списка команд личного кабинета;
* подписка на рассылку.

Функции, которые необходимо реализовать в следующих версиях:

* уведомления пользователя;
* возможность регистрации через другие ресурсы;
* интеграция данных о пользователе с другими ресурсами.

### 4.2.5 Подсистема Административная панель

Данная подсистема должна быть панелью управления проектами, агрегатором и контентом Системы. Подсистема должна быть доступна только для администратора. Функции, которые должны быть обязательно реализованы в первой версии:

* просмотр списка поданных на рассмотрение заявок для создания проектов;
* рассмотрение заявки на создание проекта в административной панели, в результате проект приобретает статус (создан, не создан);
* создание хакатона вручную;
* просмотр статистики площадки, а именно посещаемости и активности;
* создание мероприятия администратором;
* завершение мероприятия, в том числе оценка результатов команд экспертами и публикация результатов мероприятия на странице мероприятия.

Функции, которые необходимо реализовать в следующих версиях:

* управление источниками для агрегатора хакатонов;
* управление частотой сбора информации агрегатора;
* рассмотрение заявок на создание мероприятия от участников;
* архивация проектов из-за низкого интереса.

## 4.3. Требования к ролям и правам доступа

В списке ниже приведено разграничение прав доступа между 3 пользователями: гость, участник и администратор. Каждый пользователь обладает правами предыдущего, и расположены они по возрастанию от меньшей группы к большей.

Таблица 2 – Роли и функции

|  |  |  |
| --- | --- | --- |
| **Роль** | **Подсистема** | **Функция** |
| Гость | Проекты | просмотр списка проектов с помощью фильтрации по тегам, по типу проектов |
| поиск проектов с помощью поисковой строки |
| просмотр информации о проекте: название; описание; изображения; тип проекта(уникальный/учебный); список команд, которые его выполняют/выполнили |
| рулетка проектов – альтернативный инструмент подбора проектов |
| Хакатоны | просмотр списка хакатонов с помощью фильтрации по тегам, по датам начала регистрации, по размеру команды |
| просмотр информации о хакатоне: описание, даты, описание приза |

Продолжение таблицы 2 – Роли и функции

|  |  |  |
| --- | --- | --- |
| **Роль** | **Подсистема** | **Функция** |
| Гость | Хакатоны | переход на другой ресурс для записи на хакатон |
| Мероприятия | просмотр списка мероприятий |
| просмотр результатов мероприятия (отображение победителей на странице мероприятия). |
| доски со списком лидеров по достижениям |
| Личный кабинет | регистрация с помощью email, пароля и логина пользователя |
| авторизация с помощью email и пароля |
| возможность регистрации через другие ресурсы; |
| Участник | Проекты | отправление заявки на создание проекта администратору |
| отправление заявки на вступление в команду для выполнения конкретного проекта |
| рассмотрение заявки на вступление в команду создателем команды или любым из участников (при создании команды необходимо выбрать роль участника, который будет рассматривать заявки) |
| фиксация результатов деятельности команды по проекту в виде прикрепления ссылки на готовое решение и написания отзыва о проделанной работе |
| оценка результата деятельности команды другими участниками с помощью выставления оценки «нравится/не нравится» |
| Мероприятия | участие в мероприятии (функции аналогичны участию в проекте из п. [4.2.1](#_4.2.1_Подсистема_Проекты)) |
| оценка результатов команд, экспертами с помощью выставления оценок |
| Личный кабинет | восстановление пароля с помощью отправления письма на почтовый адрес; |
| просмотр проектов, которые создавал участник, в личном кабинете |
| заполнение профиля после регистрации: информация о пользователе, контактная информация; |

Продолжение таблицы 2 – Роли и функции

|  |  |  |
| --- | --- | --- |
| **Роль** | **Подсистема** | **Функция** |
| Участник | Личный кабинет | просмотр личного кабинета: информация о профиле, список достижений пользователя |
| просмотр команд, в которых состоит участник, в личном кабинете |
| переход к странице проекта из списка проектов личного кабинета |
| переход к странице команды из списка команд личного кабинета |
| подписка на рассылку |
| уведомления пользователя |
| интеграция данных о пользователе с другими ресурсами |
| Администратор | Административная панель | проведение мероприятия - хакатона с помощью системы |
| просмотр списка поданных на рассмотрение заявок для создания проектов |
| рассмотрение заявки на создание проекта в административной панели, в результате проект приобретает статус (создан, не создан) |
| создание хакатона вручную |
| просмотр статистики площадки, а именно посещаемости и активности |
| создание мероприятия администратором |
| завершение мероприятия, в том числе оценка результатов команд экспертами и публикация результатов мероприятия на странице мероприятия |
| управление частотой сбора информации агрегатора |
| рассмотрение заявок на создание мероприятия от участников |
| архивация проектов из-за низкого интереса |
| управление источниками для агрегатора хакатонов |

## 4.4. Требования к видам обеспечения

### 4.4.1. Требования к информационному обеспечению

#### 4.4.1.1. Требования к составу, структуре и способам организации данных в системе

В системе должна храниться информация о следующих объектах: проект, хакатон, мероприятие, пользователь, команда. Детализация информации об объектах показана в таблице 2.

Таблица 3 – Структура данных

|  |  |
| --- | --- |
| **Объект** | **Данные** |
| Проект | Название, тэги, тип (учебный/уникальный), картинки, описание |
| Хакатон | Название, тэги, даты начала и окончания, дата завершения регистрации, картинки, описание, описание призов, ссылка на ресурс для регистрации |
| Мероприятие | Название, тэги, даты начала и окончания, дата завершения регистрации, картинки, описание, описание призов, список экспертов, списки команд, оценки экспертов для каждого решения |
| Пользователь | ФИО, контактные данные (email), список достижений, опыт, пароль, фотография |
| Команда | Создатель команды, список ролей в команде, пользователи с их ролью в команде, результат участия команды в проекте/хакатоне/мероприятие (ссылка, отзыв), описание команды, название команды |

Логическая и физическая модель данных должна быть определена Разработчиком на стадии «Проектирование. Разработка технического проекта».

#### 4.4.1.2. Требования к информационной совместимости со смежными системами

Источники для агрегации информации о хакатонах определены в пункте [4.1.1](#_4.1.1._Требования_к).

Состав данных для осуществления информационной агрегации по каждому источнику должен быть определен Разработчиком на стадии «Проектирование. Разработка технического проекта».

#### 4.4.1.3. Требования по применению систем управления базами данных

Для реализации подсистемы хранения данных должна использоваться свободная объектно-реляционная система управления базами данных PostgreSQL - версия 14.

#### 4.4.1.4. Требования к структуре процесса сбора, обработки, передачи данных в системе и представлению данных

Система при вводе в действие должна содержать тэги, которые команда разработчиков должна самостоятельно сформировать, потому что они разбираются в данной предметной области.

Так же, первые несколько учебных проектов необходимо создать в системе перед вводом в действие. Предполагается, что эти проекты так же заполнит команда разработки.

Остальные процессы сбора, обработки и передачи данных в системе определятся на этапе «Проектирование».

#### 4.4.1.5. Требования к защите данных от разрушений при авариях и сбоях в электропитании системы

Информация в базе данных системы должна сохраняться при возникновении аварийных ситуаций, связанных со сбоями электропитания на стороне арендованного сервера.

Резервное копирование данных должно осуществляться на регулярной основе, в объёмах, достаточных для восстановления информации в подсистеме хранения данных.

### 4.4.2. Требования к лингвистическому обеспечению

При реализации системы должны применяться следующие языки высокого уровня: SQL, JavaScript, TypeScript, Python.

Для описания предметной области (объекта автоматизации) должна использоваться нотация IDEF0.

### 4.4.3. Требования к техническому обеспечению

Основные требования к хостингу:

* система должна быть расположена на виртуальном сервере VPS;
* объём дискового пространства – от 1GB;
* сервер должен быть с операционной системой Ubuntu;
* сервер должен быть с защитой от DDoS и блокиратором спама, а также с системой резервного копирования данных для восстановления в случае непредвиденных ситуаций и ошибок при обновлении.

### 4.4.4. Требования к организационному обеспечению

Основными пользователями системы являются обычные посетители сайта.

К защите от ошибочных действий пользователей должно быть разработано полное и доступное руководство пользователя, а также реализована система подсказок.

### 4.4.5. Требования к патентной чистоте

По всем техническим и программным средствам, применяемым в системе, должны соблюдаться условия лицензионных соглашений и обеспечиваться патентная чистота.

# 5. Состав и содержание работ по созданию системы

Система должна быть реализована за 7 месяцев, в этот период необходимо выполнить анализ, проектирование, реализацию и ввод в действие. Некоторые процессы необходимо выполнять параллельно.

Таблица 4 – Стадии работ по созданию системы

|  |  |  |
| --- | --- | --- |
| **Стадия** | **Результат** | **Сроки** |
| Анализ предметной области, формирование требований и разработка технического задания. | Собрана и обработана информация о:   * + предметной области получения первичного опыта IT специалистов;   + целевой аудитории и ее потребностях;   + сущности пет-проекты;   + сущности хакатоны.   Определены источники данных.  Техническое задание и приложения к нему | 01.10.2022 – 15.12.2022 |
| Проектирование. Создание технического проекта. | Архитектура системы, модели данных, сценарии использования системы, спроектированные интерфейсы пользователя | 01.12.2022 – 30.12.2022 |
| Реализация. | Разработка системы и ее документирование | 01.12.2022 – 01.04.2023 |
| Ввод в действие. | Работающий в интернете сайт | 15.02.2023-01.05.2023 |
| Сопровождение. | Система работает стабильно, без сбоев. Выявленные недостатки устранены, постоянная модерация администратором. | 15.02.2023 |

Для создания системы необходимо придерживаться итеративной модели разработки. Стадия проектирования и реализации, обязательно должны выполняться в каждой итерации для определённой цели. В разделе [4.2](#_4.2._Требования_к) определены функции, которые необходимо реализовать в каждой подсистеме. Список функций разделен на первую и последующие версии. В таблице 4 показаны сроки двух итераций проектирования и реализации системы. Разбиение на итерации необходимо для более раннего ввода в действие системы с минимальным функционалом. На второй реализации необходимо реализовать оставшиеся функции.

Таблица 5 – Итерации проектирование и реализации

|  |  |  |  |
| --- | --- | --- | --- |
| Итерация проектирования и реализации | Цель | Длительность | Даты |
| 1 | Реализация функций первой версии | 2,5 месяца | 01.12.2022 - 15.02.2022 |
| 2 | Реализация остальных функций | 1,5 месяца | 15.02.2023 - 01.04.2023 |

# 6. Требования к документированию

Документирование разработчиком производится на основе требований:

* ГОСТ Р 59795— 2021 (АС. Требования к содержанию документов);
* ГОСТР 59793— 2021 (АС. Стадии создания);
* ГОСТ 34.201-89 (Информационная технология. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированных систем).

А также, документирование регулируется перечнем документов, приведенных в таблице ниже.

Таблица 6 – Необходимые документы на этапах разработки

|  |  |
| --- | --- |
| **Стадия** | **Документ** |
| Анализ предметной области, формирование требований и разработка технического задания | Техническое задание |
| Проектирование. Создание технического проекта. | Технический проект |
| Схема функциональной структуры |
| Общее описание системы |
| Руководство пользователя |
| Разработка рабочей документации | Состав выходных данных (сообщений) |
| Каталог базы данных |
| Программа и методика испытаний |
| Спецификация |
| Описание программы |
| Код программы |