

# Формирование требований

## Структура АИС(3 семинар)

🕒 Created	@September 15, 2022 12:20 PM
🕒 Last Edited Time	@September 22, 2022 2:37 AM
▼ Type	
▼ Status	
👤 Created By	
👤 Last Edited By	
👥 Stakeholders	

### *Анализ требований*

## Что нужно сделать:

Проанализировать и записать требования к АИС

Обсудить, выявить наиболее важные функциональные и нефункциональные требования

Фиксируем какие показатели хотим достигнуть и как их будем достигать

Сформулировать и записать атрибуты качества АИС

# 1. Анализ требований

## Требований к АИС

- Масштабируемость - способность системы поддерживать как единичных пользователей так и множества
- 

## Важные функциональные и не функциональные требования

### Функциональные требования

#### 1.1.1. Ведение учета доступных ресурсов

Система должна обеспечить ведение реестра всех сотрудников, занятых в производственном процессе, для всех сотрудников ведется учет текущей роли, грейда, основных навыках и умениях

#### 1.1.2. Ведение учета потребных ресурсов

Система должна обеспечить возможность ведения «вакансий» то есть, позиций с закрепленными требуемыми ролью, грейдом и навыками, но не закрытые физически

#### 1.1.3. Ведение реестра проектов

Система должна позволять вести учет проектов (активностей) для каждой из которых указан владелец, срок начала и окончания

#### 1.1.4. Учет занятости ресурса

Система должна обеспечить возможность закрепление ресурса за проектом, такой ресурс не может быть использован в других проектах до срока завершения

проекта

### **1.1.5. Возможности поиска в системе**

Система должна обеспечить поиск в реестрах, примерами поисковых запросов могут быть: поиск сотрудников, освобождающихся с проектов в ближайшие N дней (месяцев), поиск незакрытых вакансий по критериям и так далее

### **1.1.6. Поддержка работы рекрутеров**

Система должна обеспечить возможность заведение сотрудников в статусе кандидатов. Кандидаты не доступны при поиске для выделения на проекты, но доступны для сотрудников, занимающихся интервьюированием и процессами приема

### **1.1.7. Портал поддержки**

Система должна предоставлять портал технической поддержки, на котором ведется учет инцидентов и заявок, касающихся работы системы. С этим порталом работают сотрудники поддержки

### **1.1.8. Отчетность**

Система должна обеспечить возможности получения статистических отчетов по работе системы

## **Нефункциональные требования**

1.2.1. Система должна использовать современные web технологии для предоставления современного простого, интуитивно понятного интерфейса, позволяя пользователям выполнять их работу максимально эффективным образом

1.2.2. Система должна поддерживать различные разрешения экрана (включая экраны мобильных телефонов и планшетов) без потери функциональных возможностей интерфейса

1.2.3. Система должна использовать современную, масштабируемую платформу работы с данными, позволяющую надежно хранить и обрабатывать большие объемы данных

1.2.4. Предполагается, что система будет удовлетворять ожиданиям современных пользователей. Например, время загрузки экрана любого интерфейса должно быть почти мгновенным (например, менее 2 секунд) а отчеты, включающие обработку большого объема информации, будут выполняться за адекватное время не вызывающее чувство разочарования у пользователей (точные критерии будут выработаны в процессе проектирования)

1.2.5. Система должна демонстрировать высокий уровень устойчивости и обеспечивать минимальный риск простоя в случае неполадок и сбоев

1.2.6. Система должна быть разработана и поставлена таким образом, чтобы обеспечить последующие возможности масштабирования

1.2.7. Весь трафик авторизованных пользователей должен передаваться только через зашифрованные каналы

1.2.8. Данные системы должны быть защищены от несанкционированного доступа

## Показатели

- Шифрование данных - использование современных алгоритмов шифрования. Разработка отдельного сервиса - проху-сервера с надежным хранением ключей.
- Производительность (загрузка интерфейса менее 2 секунд) - использование надежных и эффективных фреймворков. Применение современных технологий программирования (параллелизм, асинхронность, реактивность)
- Закрытость системы - Ограничение доступа для регистрации. Регистрация по специальному ключу.
- Эффективное и надежное хранение данных - Выбор оптимальной для системы СУБД.
- Web-платформа - Использование современных фреймворков (vue js или react).
- Модульность - Система представляет собой набор независимых компонент - модулей.
- Масштабируемость - проектирование системы с расчетом на горизонтальное масштабирование. Применение современных технологий программирования (параллелизм, асинхронность, реактивность)

- Устойчивость - Отказ одного из модулей не должен прерывать работу всей системы. В случае возникновения сбоев, время на перезапуск и восстановление должно быть не более 4 часов.

## **Атрибуты качества**

### **Внешние**

- Правильность
- Точность
- Гибкость
- Надежность
- Устойчивость
- Удобство использования
- Эффективность
- Безопасность

### **Внутренние**

- Гибкость
- Портруемость
- Поддержка
- Экономическая эффективность
- Защищенность
- Тестируемость
- Повторное использование
- Модульность

## **2. Архитектурная схема решения**

### **Обязанности для команд:**

## **Бизнес-аналитики (2-3 человека)**

Аналитика и планирование проекта

Контроль за процессом разработки

Проектирование моделей и диаграмм для системы

Детальное описание требований и бизнес-процессов

Разработка и проверка документации

Отчетность о проекте

## **Backend-разработчики (4-7 человек)**

Анализ и выбор стека технологий (СУБД, платформа, язык программирования, фреймворк, библиотеки и т.д.)

Разработка API

Написание кода по проектным моделям и диаграммам

Обеспечение шифрования

Разработка или внедрение модуля авторизации

Оптимизация функций

Разработка документации кода и API

Тестирование (unit-тесты)

## **Дизайнеры (1-3 человека)**

Выбор инструмента для разработки дизайна и разработка дизайна

## **Frontend-разработчики (2-4 человека)**

Анализ и выбор стека технологий (язык программирования, фреймворк, библиотеки)

Написание интерфейса по дизайнерским макетам, полученных от дизайнеров

Тестирование (unit-тесты)

## **DevOps-инженеры (3-5 человек)**

Настройка среды и выбор технологий сборки, управления и развертки всей системы

Проведение интеграционных тестов

Выпуск версий системы

- ☐ На проведение тестов можно выделить отдельную команду, которая будет заниматься исключительно тестированием отдельных модулей и всей системы в целом.

## **Функциональные части**