Отчет по курсовой работе

Этап 1.

Задание

1. Согласовать с преподавателем предметную область, для которой будет разрабатываться информационная система.
2. Составить подробное текстовое описание предметной области.
3. Сформулировать, зачем нужна информационная система для представленной предметной области, какие задачи она позволит решить.
4. Составить функциональные/нефункциональные требования к разрабатываемой информационной системе.
5. Построить модели основных прецедентов (прецеденты согласуются с преподавателем), составить их описание.
6. Предложить архитектуру будущей системы. При составлении архитектуры необходимо учитывать, что все этапы курсовой работы необходимо будет демонстрировать на сервере helios. Согласовать с преподавателем технологии и фреймворки, которые будут использоваться при реализации системы. Для реализации системы можно использовать:
   a. Frontend: React, Angular, Vue, Next JS, JSF, Spring MVC (Thymeleaf или другой шаблонизатор).
   b. Backend: основанный на Jakarta EE или Spring MVC
   c. БД: PostgreSQL
   7.Составить отчет.

Предметная область

Сайт с гайдами по игре Factorio

Текстовое описание предметной области

Factorio — это симулятор строительства и управления заводами, который требует от игроков навыков проектирования автоматизированных систем, добычи ресурсов и управления процессами производства. В связи с высокой сложностью игрового процесса, новички и опытные игроки часто обращаются к сторонним ресурсам для поиска информации, которая поможет им эффективно управлять их игровыми проектами.

Сайт будет служить платформой для публикации гайдов, новостей и других материалов по игре Factorio. Это поможет игрокам находить информацию о механиках игры, способах оптимизации процессов и общаться с другими игроками через комментарии и обсуждения.

Основные функции сайта

1. Публикация гайдов и новостей по игре.
2. Фильтрация контента по категориям (новости, гайды, обновления, модификации).
3. Оценка и комментарии пользователей на опубликованные статьи.
4. Возможность авторизации и создания пользовательских профилей.

5. Поддержка форумов или обсуждений по разным аспектам игры.
6. Поиск и сортировка контента (по дате, рейтингу, популярности).
7. Интеграция с Steam API для добавления достижений в личный профиль.
8. Отображение в личном профиле мини-карты текущего сохранения с помощью модификации игры.
9. Подбор игроков для сетевой игры по введённым параметрам.

Зачем нужна информационная система

1. Создание ресурса для предоставления актуальной информации и поддержки игрового сообщества.
2. Упрощение поиска и распространения информации о стратегиях, механиках и модах.
3. Обеспечение площадки для взаимодействия игроков через комментарии и форумы.
4. Возможность предоставления автоматизированных обновлений по новостям и гайдам.

Какие задачи решает система

1. Упрощение для игроков поиска гайдов, новостей, информации об обновлениях и модификациях.
2. Создание единой базы данных и площадки для взаимодействия игроков.
3. Возможность пользователям делиться своими стратегиями и создавать контент.
4. Обеспечение удобного пользовательского интерфейса для просмотра, сортировки и фильтрации контента.
5. Поддержка авторизации для более интерактивного взаимодействия между пользователями (рейтинги, комментарии).

Функциональные и нефункциональные требования

Функциональные требования

1. Авторизация и регистрация: Пользователи могут регистрироваться, входить на сайт, изменять профиль.
2. Управление контентом: Администраторы могут создавать, редактировать и удалять новости и гайды.
3. Фильтрация и сортировка: Возможность фильтровать гайды и статьи по категориям (новости, стратегии, обновления и т.д.), сортировать по рейтингу или дате.
4. Оценка и комментарии: Пользователи могут оценивать контент (поставить лайк/дизлайк), оставлять комментарии.
5. Поиск: Возможность поиска гайдов по ключевым словам.
6. Лента новостей: Главная страница сайта отображает последние новости и гайды.
7. Форумы: Возможность обсуждения различных тем игры на форуме.
8. Подбор игроков: Система должна подбирать игроков по введённым пользователем параметрам (количество часов в игре, предпочитаемые модификации, наличие различных соцсетей, возможное время для игры).

Нефункциональные требования

1. Производительность: Время отклика на запросы пользователя до 2 секунд на загрузку страницы.

2. Масштабируемость: Поддержка растущего количества пользователей и контента без значительного ухудшения производительности.
3. Безопасность: Система должна поддерживать шифрование персональных данных пользователей для обеспечения безопасности с помощью алгоритма AES.
4. Доступность: Система должна быть доступна 99.9% времени.
5. Поддержка мобильных устройств: Адаптация интерфейса под мобильные устройства.
6. Нагрузка: Система должна поддерживать до 100 активных пользователей без потери производительности.
7. Совместимость: Система должна быть совместима с популярными браузерами (Chrome 118.0, Firefox 119.0, Safari 17.0, Microsoft Edge 118.0, Yandex 23.9 или более поздние версии).

Модели основных прецедентов

Прецедент 1: Пользователь создаёт аккаунт

```
Система: FactorioGuide.ru
Основное действующее лицо: Пользователь
Цель: Создание аккаунта
Триггер: Пользователь пытается создать аккаунт
Результат: Система создаёт личный аккаунт
Основной сценарий:
1. Пользователь переходит на страницу регистрации.
2. Вводит необходимые данные.
3. Нажимает на кнопку "Зарегистрироваться".
4. Система проверяет корректность данных.
5. Система создаёт нового пользователя.
6. Пользователь получает уведомление об успешной регистрации.
```

Прецедент 2: Пользователь публикует гайд

```
Система: FactorioGuide.ru
Основное действующее лицо: Авторизованный пользователь
Цель: Публикация гайда
Триггер: Пользователь пытается создать новый гайд
Результат: Система опубликовала на сайте новый гайд
Основной сценарий:
1. Пользователь заходит в свой профиль.
2. Переходит в раздел создания гайда.
3. Вводит необходимую информацию.
4. Нажимает на кнопку "Опубликовать".
5. Система проверяет данные.
6. Система публикует гайд на сайте.
```

Прецедент 3: Пользователь комментирует гайд

Система: FactorioGuide.ru
Основное действующее лицо: Авторизованный пользователь
Цель: Написать комментарий под гайдом
Триггер: Пользователь пытается написать комментарий под гайдом
Результат: Система оставила новый комментарий под гайдом
Основной сценарий:
1. Пользователь заходит на страницу гайда.
2. Пользователь вводит текст комментария.
3. Система выполняет автоматическую модерацию комментария
4. Система сохраняет комментарий.
5. Комментарий отображается под гайдом.

Прецедент 4: Пользователь ищет гайд или новость по ключевым словам

Система: FactorioGuide.ru
Основное действующее лицо: Пользователь
Цель: **Найти** необходимый контент
Триггер: Пользователь пытается **найти** необходимый контент на сайте
Результат: Система выдаёт пользователю список найденного контента
Основной сценарий:
1. Пользователь заходит на сайт и вводит ключевое слово в поисковую строку.
2. Система обрабатывает **запрос** и выполняет поиск **по** заголовкам и содержанию гайдов и новостей.
3. Система отображает результаты, соответствующие введенному ключевому слову.
4. Пользователь просматривает результаты и выбирает нужный материал.

Прецедент 5: Пользователь ставит оценку гайду или новости

Система: FactorioGuide.ru
Основное действующее лицо: Авторизованный пользователь
Цель: Поставить оценку под гайдом **или** новостью.
Триггер: Пользователь пытается поставить оценку.
Результат: Система оставляет оценку **по** гайдом **или** новостью.
Основной сценарий:
1. Пользователь находит интересующий его гайд **или** новость.
2. Нажимает на кнопку «Лайк» **или** «Дизлайк».
3. Система фиксирует оценку и обновляет общее количество лайков и дизлайков.
4. Система обновляет рейтинг материала, и оценка отображается **для** всех пользователей.

Прецедент 6: Пользователь использует форум

```
Система: FactorioGuide.ru
Основное действующее лицо: Авторизованный пользователь
Цель: Создать раздел или написать сообщение на форуме.
Триггер: Пользователь хочет использовать форум.
Результат: Система создаёт раздел или пишет сообщение на форуме.
Основной сценарий:
1. Пользователь заходит в раздел «Форум».
2. Пользователь выбирает опцию «Создать тему» и переходит на страницу создания темы.
3. На странице создания темы пользователь заполняет заголовок и текст сообщения.
4. Пользователь нажимает на кнопку «Опубликовать».
5. Система проверяет корректность введённых данных и публикует тему в соответствующем разделе форума.
6. Тема становится доступной для других пользователей, которые могут её просматривать и комментировать.
Дополнительный сценарий:
1. Пользователь заходит на страницу интересующей его темы на форуме.
2. В начале страницы с темой пользователь находит форму для ответа.
3. Пользователь заполняет текст ответа и нажимает на кнопку «Отправить».
4. Система проверяет корректность данных и добавляет комментарий под основной темой.
5. Ответ отображается на странице темы, доступный для просмотра и обсуждения другими пользователями.
```

Прецедент 7: Пользователь подбирает игрока

```
Система: FactorioGuide.ru
Основное действующее лицо: Авторизованный пользователь
Цель: Найти игрока для сетевой игры.
Триггер: Пользователь хочет найти игрока для игры по сети.
Результат: Система выводит список пользователей, ищущих совместной игры.
Основной сценарий:
1. Пользователь заходит в раздел «Подбор игроков»
2. Заполняет параметры поиска
3. Система выполняет поиск и выводит список подходящих пользователей.
4. Пользователь выбирает пользователя и отправляет приглашение.
5. Другой пользователь подтверждает приглашение.
Дополнительный сценарий:
5. Другой пользователь отклоняет приглашение.
Исключение:
1. Если система не находит подходящих игроков, она отображает сообщение и предлагает настроить параметры поиска заново или оставить запрос на подбор игроков
```

Архитектура системы

Frontend
React

Backend
Spring MVC

Database
PostgreSQL

Размещение
Сервер helios