ФГАОУ ВО «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»

Кафедра вычислительной техники

КУРСОВАЯ РАБОТА

Программирование интернет-приложений

Саржевский Иван Анатольевич

Лабушев Тимофей Михайлович

Группа P3202

Санкт-Петербург

2018 г.

Sumaju nikki

Браузерная многопользовательская текстовая игра
жанра Zero-Player Game

Содержание

1 Введение

Предметом разработки является платформа для многопользовательской текстовой игры жанра *Zero-Player Game (ZPG)*, использующая браузерный интерфейс и сохраняющая прогресс игрока на игровом сервере.

В основе жанра лежит концепция протекания игрового процесса без вмешательства со стороны пользователя. При этом игроку предоставляются возможности ускорения своего игрового прогресса и взаимодействия с другими пользователями.

Предполагается, что взаимодействие с платформой будет представлять собой не длинные сессии, свойственные традиционным играм, а короткие, но частые посещения, характерные для социальных сетей.

1.1 Цель создания

Платформа разрабатывается с целью предоставления творческим молодым людям возможности самовыражения в игровой форме. Платформа предоставляет набор базовых игровых механик, в то время как содержательная часть создается самими пользователями, путём предоставления возможности формировать предложения для контента

1.2 Целевая аудитория

Творческие молодые люди, в возрасте 18-25 лет, знакомые с современными медиа, активные пользователи интернета. Не имеют времени/желания проводить много времени за игрой в компьютерные игры, ищут более подходящего под свой стиль жизни формата.

2 Прецеденты использования

Регистрация

Цель: Завести аккаунт в системе, создать персонажа

Сценарий:

1. Незарегестрированный пользователь открывает любую страницу платформы.
2. Система перенаправляет пользователя на главную страницу с формой регистрации.
3. Пользователь вводит адрес электронной почты и пароль, отправляет форму.
4. Система создает аккаунт и перенаправляет пользователя на страницу создания персонажа.
5. Пользователь указывает имя и пол персонажа, опционально загружает картинку, отправляет форму.
6. Система авторизует пользователя.

Расширения:

- На шаге 3 пользователю предоставляется возможность регистрации используя аккаунт социальной сети.
- Ошибка на шаге 4 (адрес электронной почты уже используется, пароль содержит менее восьми символов) предотвращает переход к следущему шагу.

Авторизация

Цель: Войти в систему

Сценарий:

1. Незарегестрированный пользователь открывает любую страницу платформы.
2. Система перенаправляет пользователя на главную страницу с формой входа.
3. Пользователь вводит адрес электронной почты и пароль, отправляет форму.
4. Система авторизует пользователя и перенаправляет его на страницу персонажа.

Расширения:

- На шаге 3 пользователю предоставляется возможность авторизоваться, используя аккаунт социальной сети.
- На шаге 4 возникновение ошибки (адрес электронной почты уже используется, пароль содержит менее восьми символов) предотвращает переход к следущему шагу.

Наблюдение за игровым процессом

Цель: Отследить развитие персонажа, прочесть интересные записи в дневнике

Сценарий:

1. Авторизованный пользователь открывает страницу персонажа.
2. Система предоставляет обзор характеристик персонажа, его текущее занятие и цели для повышения уровня.
3. За один клик пользователь имеет возможность перейти к дневнику персонажа, энциклопедии известных персонажу существ, карте мира.

Расширения:

- Пользователь может увидеть краткую информацию о состоянии персонажа (его здоровье и текущее занятие) в шапке любой страницы игрового портала.

Участие в игровом процессе

Цель: Оказать влияние на развитие персонажа

Сценарий:

1. Авторизованный пользователь открывает меню *сов* с помощью кнопки, доступной в шапке любой страницы игрового портала.
2. Пользователь выбирает из списка доступных сов одну.
3. Система добавляет сову во внутреннюю очередь эффектов хода, уменьшает количество доступных сов выбранного типа на одну, показывает пользователю уведомление об успешном начале полета птицы.
4. По завершению хода в дневнике персонажа появляется заметка о действии совы.

Расширения:

- Число возможных сов одного типа ограничено, что должно быть понятно пользователю из интерфейса системы.

Взаимодействие с другими игроками

Цель: Повлиять на отношения персонажа с другими учениками, усилив таким образом влияние на персонажа сов

Сценарий:

1. Авторизованный пользователь выбирает *чернильную сову* из списка доступных.
2. Система предоставляет пользователю форму с поиском персонажа-получателя по имени, в которой доступно автодополнение и список последних адресатов.
3. Пользователь выбирает получателя.
4. Система случайно выбирает *клуб*, для которого создается запись о письме. При посещении персонажем-получателем клуба, его отношения с персонажем пользователя изменяются.
5. В дневники обоих персонажей добавляется запись об изменении отношений.

Расширения:

- Действие невозможно, если у пользователя отсутсвует необходимый тип совы.

3 Требования к аппаратно-программному обеспечению

3.1 Требования к серверному обеспечению

Система, которая обеспечивает выполнение программных продуктов сервера приложений и хранение данных платформы, должна отвечать следующим требованиям:

- Поддержка операционной системой бинарного интерфейса приложений (ABI) Linux
- Наличие сервера баз данных PostgreSQL версии 9.6 и выше

3.2 Требования к клиентскому обеспечению

Браузерный интерфейс разрабатывается с учетом следующих требований к программному обеспечению на стороне пользователя:

- веб-браузер Google Chrome версии 67 и выше или Mozilla Firefox версии 61 и выше с включенным интерпретатором сценариев JavaScript,
- отсутствие запрета веб-страницам платформы доступа к внешним ресурсам, а именно изображениям, шрифтам, таблицам стилей CSS и сценариям JavaScript, в том числе блокировщиками рекламы.

4 Требования к архитектуре системы

4.1 Глоссарий

Уровень back-end — серверное приложение, с которым взаимодействует пользовательский интерфейс игры. Уровень back-end обеспечивает хранение данных, расчет игровых процессов, взаимодействие между пользователями.

Уровень front-end — браузерное приложение, которое реализует пользовательский интерфейс игры. Уровень front-end включает в себя HTML-страницы, сценарии JavaScript, таблицы стилей CSS.

4.2 Уровень back-end

1. Серверное приложение должно разрабатываться на платформе *JVM* с использованием фреймворков *Akka* и *Play*.

2. Взаимодействие между уровнями front-end и back-end должно осуществляться посредством REST API; возможно использование протокола WebSockets для реализации обновлений в режиме реального времени.

3. Серверное приложение должно реализовывать аутентификацию пользователей с поддержкой входа через социальные сети (OAuth), используя библиотеку *play-silhouette*. Пароли пользователей должны храниться как криптографический хэш.

4. Серверное приложение должно публиковать push-уведомления об игровых событиях, используя сервис *Firebase Cloud Messaging*.

5. Серверное приложение должно отправлять пользователям еженедельное новостное сообщение электронной почты, используя *JavaMail API*.

6. Серверное приложение должно предоставлять API для доступа к очереди пользовательских предложений, а также их принятию, отклонению и редактированию (см. раздел 4.4).

4.3 Уровень front-end

1. Клиентское приложение должно разрабатываться с использованием фреймворка Vue.js.

2. Веб-интерфейс должен быть адаптирован для отображения в трех режимах: *десктопном* (ширина экрана больше 1107px), *планшетном* (больше 889px) и *мобильном* (меньше 889px).

3. Веб-интерфейс должен предоставлять пользователю возможность подписаться на push-уведомления.

4. Веб-интерфейс должен распознаваться в мобильной ОС Android как приложение, используя *Web App Manifest*.

4.4 Telegram-бот

Модерация добавляемого пользователями контента должна осуществляться при помощи Telegram-бота.

1. Бот должен работать в заранее определенном чате, в котором находятся администраторы платформы.
2. Предложения должны выноситься ботом по одному на обсуждение в чат. Администраторы должны иметь возможность принятия, отклонения, редактирования предложения.

5 Архитектура системы