

Техническое задание:

Цель проекта – создание здоровой соревновательной атмосферы для увеличения продуктивности и производительности, повышения мотивации и вовлеченности сотрудников компании, улучшение качества выпускаемого ПО

Формулировка задачи: разработка системы геймификации. Проект - веб-сайт, стилизованный под пиксельную игру. У каждого участника команды есть свой подоконник с цветочками в горшках в личном кабинете. Каждый из цветов отвечает за метрику или набор метрик. Чтобы цветочки не завяли, сотруднику надо выполнять задачи, которые выгружаются из трекера задач. За выполнение задач начисляются баллы, они определяются с помощью ИИ, который персонализирует задачи под уровень пользователя

Есть вторая страница на сайте с общим большим плодовым деревом. В конце месяца у дерева появляются плоды, а на плодах имя каждого участника и его средний процент вклада за месяц. Лучшие сотрудники получают мерч компании или другие поощрения

Описание источников:

- Внутренняя база данных сотрудников, руководителей и набранных процентов вклада по датам
- Jira – трекер задач, в котором отслеживается процесс их выполнения

Описание логики реализации:

1. Название задачи: геймификация взаимодействия в IT-командах

Структура Сайта

- Главная страница
 - Кнопка входа в ЛК
 - Личный кабинет пользователя: его профиль, «подоконник» и его рейтинг внутри команды
 - Страница с общим деревом
 - Политика конфиденциальности
 - Пользовательское соглашение

2. Функциональные требования:

- Сотрудник компании авторизуется через Jira и попадает в ЛК на сайте
 - Проверить роль сотрудника:
 - Если роль = менеджер, то отображать следующие цветы-метрики:
 1. производительность команды
 2. уровень активности сотрудников
 3. процент выполнения планов спринта
 4. уровень удовлетворенности подчиненных
 - Если роль = разработчик, то отображать следующие цветы-метрики:
 1. участие в код-ревью
 2. частота релизов и развертываний
 3. вклад в производительность команды (в процентах)
 4. количество решенных задач за день (в сторипоинтах)
 - Если роль = аналитик, то отображать следующие цветы-метрики:
 1. участие в тестировании
 2. количество данных, обработанных за неделю работы
 3. вклад в производительность команды (в процентах)
 4. количество решенных задач за день (в сторипоинтах)
 - Коллективные метрики для команды, работающей над одним проектом:
 1. Time to Market (Время до выхода на рынок)
 2. Burn-down Chart (График выполнения задач)
 3. Work in Progress (WIP)
 4. Velocity (Скорость команды)
 - Когда сотрудник выполняет задачи, отвечающие за определенную метрику, баллы за метрику добавляются и этот цветочек расцветает. Если задача не выполнена – цветочек с метрикой увядает, баллы не добавляются
 - Каждый сотрудник может отметить свое впечатление от рабочего дня, общения с менеджером и командой и задач по десятибалльной шкале.

Из этих баллов находится среднее значение по проголосовавшим людям в команде и идет в счет метрики «уровень удовлетворенности подчиненных» для менеджера этой команды

- В конце рабочего дня баллы каждого сотрудника суммируются. Вычисляется его процент вклада в проект по формуле: $\text{проц}_{\text{вклада}} = \frac{\text{сумма_баллов_сотр}}{\text{сумма_баллов_всех_сотр}} * 100 \%$. У всех в ЛК виден рейтинг, в котором по убыванию процента вклада расположены все сотрудники, зарегистрированные на сайте
- В рейтинге так же видно у какого менеджера наиболее довольные сотрудники (метрика «уровень удовлетворенности подчиненных»)
- С утра баллы обнуляются, подгружаются новые данные из Jira. Рейтинг так же обнуляется. Процент вклада для каждого сотрудника за каждый рабочий день вносится в базу данных
- В конце месяца на общем дереве отображается средний процент вклада для каждого сотрудника за этот месяц
- Также на общем дереве в конце месяца отображаются результаты по коллективным метрикам в баллах для всех команд и проектов, работающих в компании. Образуется рейтинг команд по убыванию набранных баллов

Применение ИИ:

- ИИ анализирует уровень каждого сотрудника, его грейд (интерн, джуниор, миддл и тд), стаж работы в компании и процент успешного набора баллов за различные метрики в прошлом и персонализирует игру под пользователя: с ростом этих характеристик баллы набирать становится сложнее

3. Нефункциональные требования

- Простота в использовании, интуитивно понятный интерфейс с минимальным количеством кнопок
- Минималистичный дизайн в стиле «пиксель-арт» в одной цветовой гамме
- На сайте должна быть реализована адаптивная верстка, в частности:
 - Отзывчивая верстка – верстка HTML-страниц, обеспечивающая отображение сайта на различных устройствах, подключенных к Интернету и динамически подстраивающаяся под заданные размеры окна

- Резиновая верстка – верстка, обеспечивающая автоматическое изменение ширины блоков сайта в зависимости от ширины рабочего поля браузера
- Отзывчивая верстка сайта должна поддерживать корректное отображение в следующих браузерах:
 - Internet Explorer 10.0 и выше
 - Opera Browser последней версии
 - Mozilla Firefox последней версии
 - Google Chrome последней версии
 - На мобильных устройствах под управлением ОС Android, iOS
- Страницы сайта должны корректно отображаться без полосы горизонтальной прокрутки и автоматически масштабироваться для следующих разрешений экрана (по ширине):
 - Мобильная версия: 320 – 767px (резиновая верстка)
 - Планшетная версия: 768 - 959px (резиновая верстка)
 - Десктопная малая версия: 960 – 1279px (резиновая верстка)
 - Десктопная большая версия: 1280px и более (фиксированная верстка)
 - На экранах с шириной разрешения более 1280px рабочая область страницы должна иметь фиксированную ширину, по горизонтали выравниваться по центру экрана
- Производительность. Система должна легко и быстро работать на любом устройстве
- Все компоненты CMS должны физически размещаться на одном сервере и работать с одной базой данных
- Защищенность данных от злоумышленников
- Сайт должен быть доступен для просмотра посетителями только из локальной сети и недоступен для остальных пользователей сети Интернет
- Устойчивость. Если на сайте произойдет сбой, набранные за день баллы не должны быть утеряны, поэтому, как только сотрудник получает балл, этот балл вносится в базу данных
- Обеспечить авторизацию на сайте через Jira
- Обеспечить обмен данными с Jira
- Интеграция с нейросетью для персонализации сложности игры
- Необходима внутренняя база данных, в которую вносятся проценты вклада в проект для каждого сотрудника

Связи таблиц в реляционной базе данных:

Employees – таблица с разработчиками и аналитиками. Включен номер проекта, над которым работают сотрудники, и логин в Jira. Аналогичная таблица **managers** для менеджеров

Results – таблица, в которую вносится баллы каждого сотрудника за день, процент вклада в проект для каждой даты и баллы удовлетворенности сотрудников работой для менеджеров. По данным из этой таблицы вычисляется средний процент за месяц

