

Find your hostel

(Котов Максим, Морозов Сергей, Атаманова Анна)

Этапы работы

1. Детальный сбор информации о функционале системы.
 - a. Выделение функционала первой версии.
2. Анализ предварительных сроков реализации.
3. Подбор разработчиков.
4. Разработка дизайна приложений и административно-информационного портала
5. Разработка серверной архитектуры
6. Разработка клиентской архитектуры для обеих мобильных платформ(iOS и Android)
7. Разработка API
8. Анализ сроков реализации
9. Реализация серверной части
 - a. Развертывание тестового сервера в облаке
 - b. Настройка мониторинга
 - c. Покрытие кода тестами
10. Реализация мобильных клиентов
 - a. Настройка систем бета-тестирования
 - b. Настройка мобильной аналитики
 - c. Покрытие кода тестами
11. Реализация административно-информационной панели
12. Тестирование и отладка
13. Настройка систем резервного копирования для БД
14. Релиз
 - a. Развертывание сервера в облаке (Digital Ocean напр.)
 - b. Загрузка приложений в магазины.
15. Поддержка.
16. Анализ статистики и улучшение.

Риски

При разработке нужно учесть следующее:

1. Потеря разработчиков

- a. Необходимо заставлять разработчиков писать чистый, документированный код, который способны понять другие члены команды. В этом помогут соглашения о стиле разработки и просмотр кода другими членами команды.
 - b. Обеспечивать высокую мотивацию сотрудников с помощью денежных вознаграждений за выполнение с опережением графика, а так же поддержанием командного духа.
- 2. Застои в разработке
 - a. Учесть застой в разработке в следствие невыполнения заявленных гарантий доступности облачным провайдером и заложить в план предполагаемое время простоя.
 - b. Не допускать простоя разработчиков, следить за зависимостями задач и их выполнением, особенно если таковые находятся на критических путях в сетевом графике.
 - c. Перераспределять задачи между разработчиками.
- 3. Утечки конфиденциальной информации пользователей.
 - a. Обеспечить защиту передачи данных по сети с помощью SSL.
 - b. Ограничить доступ к серверу по SSH.
- 4. Учесть возможные изменения требований заказчика
 - a. Вести разработку версионно, с наращиванием функционала.
 - b. В каждой версии определять 1-2 недельные отрезки и функционал, который должен быть разработан в конкретном отрезке.
 - c. Поддерживать постоянный контакт с заказчиком.

Системные требования

- 1. Для корректного использования системы пользователям необходим доступ в интернет, иначе будут доступны только закешированные данные.
- 2. iOS 8.0 или выше; Android 4.0.2 или выше.
- 3. Пользователям с ролью staff необходим браузер Internet Explorer 8 или выше, браузер Google Chrome 30 или выше, браузер Mozilla Firefox 20 или выше. Корректная работа с другими браузерами не гарантируется.
- 4. Для работы функции *find the nearest hostels* в устройстве, на котором установлено приложение, должен быть GPS или ГЛОНАСС адаптер.
- 5. Экран устройства, на котором установлено приложение должен воспринимать как минимум трехпальцевое нажатие.

Бизнес требования

1. Роли пользователя
 - a. tourist. Осуществляет взаимодействия клиентов с приложением
 - b. staff. Осуществляет взаимодействие хостела с приложением
2. Обширная база хостелов и интересных мест рядом
3. Инструменты для сбора статистики и аналитики системы
 - a. активные пользователи
 - b. статистика потребления ресурсов
4. Привлечение хостелов
 - a. Наличие специальных кадров для контроля ситуаций в мире хостелов и связи с ними.
5. Привлечение клиентов
 - a. Реклама

Техническая часть

В системе можно выделить 3 части:

- Сервер
- Мобильные клиенты
- Административно-информационный портал

Серверная часть:

- БД - PostgreSQL
- Фреймворк разработки - Ruby on Rails
- Развертывание в Digital Ocean
- Доступ по SSH
- Дополнительный сервер для бэкапа БД
- RESTful API
- Защита обмена данными по сети с помощью SSL

Мобильный клиент iOS:

- Версия 8.0 и выше
- Адаптивная верстка и под iPhone и под iPad

- Язык разработки - Obj-C
- Оптимизация под arm64
- Использование RestKit для взаимодействия с сервером, отображения данных в локальные сущности и их кэширования
- Покрытие приложения событиями и их логирование с помощью Flurry
- Fabric для распространения бета-версий приложения
- Crashlytics(часть Fabric) для сбора информации о падениях приложения

Мобильный клиент Android:

- Версия 4.0.2 и выше
- Верстка, поддерживающая большинство классов экранов смартфонов
- Язык разработки - Java
- Restlet для взаимодействия с сервером, отображения данных в локальные сущности и их кэширования
- Покрытие приложения событиями и их логирование с помощью Flurry
- Fabric для распространения бета-версий приложения
- Crashlytics(часть Fabric) для сбора информации о падениях приложения

Административно-информационная панель:

- HTML + CSS + JS
- Верстка по мобильные устройства

Требования к производительности

1. Сервер + Административно-информационная панель
 - а. При заказе статистики пользователем с ролью staff время сбора информации с серверов не должно превышать 30 секунд для периода прошедший день, 2 минут для периода прошедшая неделя и 5 минут для периода прошедший месяц. Максимальное время ожидания для промежутка времени “весь период” не должно превосходить 10 минут.
2. Сервер + Приложение
 - а. Отклик сервера на запросы от приложения не более 2 секунд в часы пиковой нагрузки. Не более 0.5 секунд в среднем.