



Роли

- Project manager
- Product owner
- Developer 1
- Developer 2
- QA



За проекта

- Цел на проекта
- Очаквани резултати
- Планиране
- Контрол на изпълнение
- Оценка на рискове



Project manager

Като ръководител на проекта, основните ми отговорности ще се опират до планирането, организирането и ръководството по проекта, като същевременно ще следя за изпълнението на задачите в срок, по бюджет и в обхвата на изискванията.

Product owner ще има задачата да изготви бизнес модел, стратегии и предложения за печалба и идеи за популяризиране на идеята. За разработването на самата платформа ще бъдат ангажирани 2-ма разработчика, а 1 QA ще се грижи за качеството и сигурността на изпълнените задачи.



- Нашето приложение ще помага за намирането на паркинг.
- Целта ни е да направим приложение в което всеки да може да си намери паркомясто в близост до него.
- Заинтересованите страни са всички водачи на МПС.
- Очакваният резултат е хората по-лесно да си намират места за паркиране.
- Плана се приема предвид гледните точки и идеите на ключовите заинтересовани страни. По този начин ще се контролират сроковете, рисковете, проблемите с качеството,. По време на изпълнението ще се следи за завършването на работата, и по време на въпросното планиране, ще бъдат правени съпоставки между планирано изпълнение и действително изпълнение и планиран бюджет и действителни разходи.



Product owner

- Трябва да бъде финансирана нашета идея: Финансирането на приложението ще бъде от собствениците на всеки паркинг във всеки един град.
- Ще се сключи договор с паркинг-компаниите: В приложението ще предлагаме паркинга, а те ще ни дават процент от печалбата .



-
- Печалба: Приложението ще бъде безплатно за сваляне. Ще има реклами на самото приложение от където ще се трупа печалбата. По този начин хем ще се рекламира приложението ни, хем и ще печелим пари. Точно от това че е безплатно самото сваляне предполага голям брой сваляния от което също ще имаме печалба.



Developer 1

Въведение.

- Ами, за начало искам да кажа, че самият език за програмиране не е основна част от успешен проект, тук по-голямата част от „Успеха“ зависи от уменията на самия програмист.

Но все пак езиците за програмиране също са част от проекта и вижте за тях по-долу ...



Езици за програмиране.

- За това приложение трябва да има база данни, идеалният вариант е SQL
- В зависимост от (iOS / Android) мобилна разработка за iOS, перспективно да се използва езикът Swift. Ако приложението е на Android — Kotlin.
- За да разширим възможностите си, можем да добавим Java, C ++.



Developer 2

Обща архитектура

- Включват се бази данни от няколко източника:
- ОП „Паркиране и репатриране“
- инсталирани сензори на всеки паркинг включен в системата за всяко паркомясто.
- Google APIs
- Използвайки **MySQL Connector/C++** информацията може да се компилира спрямо локацията.
- Могат да се ползват и други езици за програмиране, за да има възможност за добавяне на други плъгини или възможности.



Обща архитектура

- GPS на мобилното устройство се използва за определяне на време и разстояние до паркомястото и се извеждат данни при избор на място.
- При съвпадане на местоположението на устройството с паркомястото се извежда бутон за плащане, при натискане на бутона излиза меню за избор на време за престой и начин на плащане – PayPal или с карта към трета страна за сигурно плащане (3D secure или друго).
- При плащане се запазва точното местоположение за определеното време.
- Подава се заявка към базата данни.



UI and viewpoint

Компоненти

- Съвместимост с версии на Android/iOS
- User-agreement с източниците на информация и със собствениците на паркинги
- Ресурси за поддръжка на софтуерния продукт, добавяне на допълни сензори или използване на вече съществуващи такива
- Създаване на ясна и последователна логика и правила при изграждане на продукта

Логин

- Влизане на приложението чрез телефонен номер или друг вид верификация, искане на разрешение за използване на мобилни данни и GPS
- След логин – изскачащ прозорец - предложение за откриване на паркинг места наблизо. При съгласяване се извеждат наличните в базата данни свободни места, спрямо координатите на устройството



Нива за разработка

1. Компиляция на наличните данни спрямо местоположение
2. Контролер
3. Оптимизация на заявките и показване на свободни места в реално време
4. Известие при съвпадение на местоположението на устройство с паркомясто, извеждане на меню за опции
5. Система за плащане
6. Връщане към основно меню/навигация след успешно плащане на паркомясто



QA

Намиране на паркинг

Заглавие: Намиране на най-близкия паркинг спрямо локацията на потребителят.

Описание: Намира се паркинг с натискането на UI елемент.

Предусловие: Потребителят трябва да е свързан с интернет.

Предположение: Използва се мобилно устройство с интернет функционалност.

Стъпки за тестване:

1. Кликаме на иконата на приложението
2. Потребителят натиска locate бутон.
3. Автоматично се задават направления към най-близък паркинг.



Запомняне на локация

Заглавие: Запомняне на локацията на паркиране след пристигане на дестинацията.

Описание: Запомня се локацията автоматично при спиране на превозното средство.

Предусловие: Потребителят трябва да е свързан с интернет.

Предположение: Потребителят е изпълнил предишната стъпка и е зададен курс.

Стъпки за тестване:

1. Излизаме от превозното средство.
2. След период от време влизаме в приложението.
3. Автоматично се показва точната локация на превозното средство.



Заплащане на паркомясто

Заглавие: Заплащане на паркомясто, спрямо прекарано време в паркинга.

Описание: Заплаща се прекараното време, като се online payment solution

Предусловие: Потребителят трябва да има поддържан начин за заплащане.

Предположение: Потребителят е наясно с начинът на плащане, преди използване.

Стъпки за тестване:

1. Преди излизане, се влиза в приложението и се отваря системата за заплащане.
2. Потребителят въвежда детайлите на предпочитаният от него метод.
3. Автоматично се показва дали заплащането е успешно.

