

Table of contents

A table of contents with pages numbers indicated for all sections / headings should be included.

1. Introduction

1.1 Overview

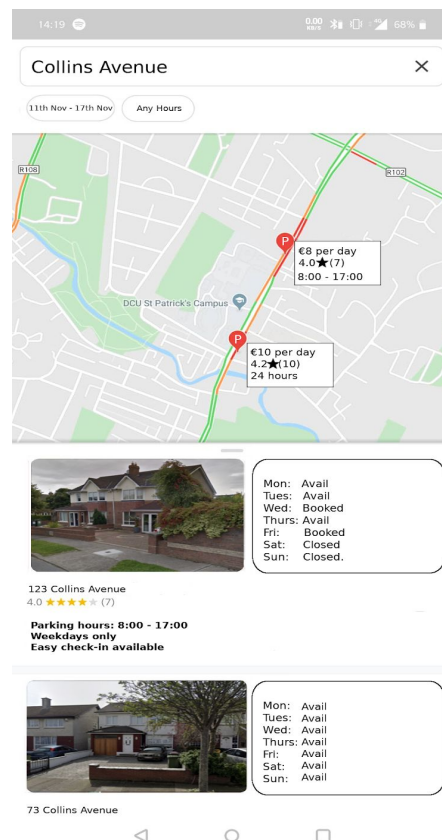
“AirBnb for parking”

The aim of our project will be to develop an android app that will allow homeowners to easily rent out their driveways as parking spaces to commuters with minimal effort. And likewise enable commuters to easily view and book parking spots with competitive rates.

The app will allow homeowners to list their driveway property as available to rent. They will be able to set their own rates on a daily, weekly or even monthly basis and can choose to give discounts for booking multiple days. The homeowner can choose to allow overnight parking or a time frame such as parking between 8am to 6pm regardless of the amount of days booked.

From the renters perspective the app will allow them to choose an area they wish to park nearby and show them all available spaces to rent that fit their requirements. They will be able to view the availability of the driveways to rent for their preferred dates and compare the rates for each property.

We will be utilising NFC to provide a hassle free way to check-in for both renters and homeowners. The homeowner can optionally register a property nfc tag that the renter can use in conjunction with the app to tap and alert the owner that they have arrived. Likewise when the renter is leaving they will tap and the homeowner will be notified through the app's push notification that their driveway is now free.



Mockup of the park space search UI

1.2 Glossary

- [Android](#): Mobile operating system
- [Android Studio](#): IDE built upon IntelliJ targeting android app development
- [NFC](#): Near field communication, a protocol allowing two electronic devices to pass data between each other when in close proximity.
- [Firebase](#): Realtime cloud database allowing us to query and sync information in real time.
- [Google maps api](#): Leveraging google maps api will allow us to implement the geographical aspect of our app.

2. General Description

2.1 Product / System Functions

The goal of the product is to allow homeowners to be able to lease their driveway and commuters to book those properties with ease. This will be accomplished through the android app by using the google maps api to display all properties available in a given area of interest to the commuter.

The homeowner will be able to set the hours of availability and pricing with minimal input through the android app.

The app will handle management of bookings, storing them on a firebase database instance, ensuring that both homeowner and commuter's have the latest snapshot of bookings for a given property.

The check-in functionality will be accomplished using nfc tags at the homeowners property to give notice of the commuters arrival and then to ensure they have left by the agreed time. The app will handle the writing of the unique property identifier to the card as well as the reading by the commuters phone to confirm the arrival and leaving of the property.

Analysis on historical bookings for a given area will be utilised to provide price listing advisement to homeowners to ensure they're staying competitive in their area. Likewise this will be provided to commuters to so they are aware they're getting a reasonable deal on the rental of the driveway.

The app will allow all users of the app to give a review in the form of a star based system following the booking and use of the parking space, which will then be displayed on property listings and commuter profiles.

2.2 User Characteristics and Objectives

Describes the features of the user community, including their expected expertise with software systems and the application domain. Explain the objectives and requirements for the system from the user's perspective. It may include a "wish list" of desirable characteristics, along with more feasible solutions that are in line with the business objectives.

2.3 Operational Scenarios

This section should describe a set of scenarios that illustrate, from the user's perspective, what will be experienced when utilizing the system under various situations.

In the article Inquiry-Based Requirements Analysis (IEEE Software, March 1994), scenarios are defined as follows:

In the broad sense, a scenario is simply a proposed specific use of the system. More specifically, a scenario is a description of one or more end-to-end transactions involving the required system and its environment. Scenarios can be documented in different ways, depending up on the level of detail needed. The simplest form is a use case, which consists merely of a short description with a number attached. More detailed forms are called scripts.

2.4 Constraints

1. **Time:** The project is on a deadline of 11th of May 2020 and as such the scope of the project must be achievable within this time frame. In light of this we will aim to deliver the projects features in stages, ensuring that we have a working prototype at the end of each sprint. Additional features will be built upon base functionality.
2. **Scope:** Due to the aforementioned time frame, we must be careful in deciding the exact scope of the project, what our expected outcome of the project will be. This will require constant assessment subject to the progress of the development of the apps base functionality and subsequent features, each will be ranked in order of importance.
3. **Performance:** The app must be able to process all user requests in a quick and efficient manner, whether that is adding a new property, searching for one, booking a park space or placing a review. This is a separate constraint from UX(User experience) or usability of the app, rather the throughput of the app.
4. **User Experience (UX):** The app must be usable by any user with a base understanding of android apps. Each element of the ui that can be interacted with must give a clear affordance of what it is responsible for and how it can be used. In particular the map view and listing of driveways to rent must be conveyed in a simple but informative manner.

TODO

5. Payment processing? ie: Paypal/Google wallet, security concerns?
6. NFC? ie: Mandatory?

3. Functional Requirements

This section lists the functional requirements in ranked order. Functional requirements describes the possible effects of a software system, in other words, *what* the system must accomplish. Other kinds of requirements (such as interface requirements, performance requirements, or reliability requirements) describe *how* the system accomplishes its functional requirements.

As an example, each functional requirement could be specified in a format similar to the following:

- **Description** - A full description of the requirement.
- **Criticality** - Describes how essential this requirement is to the overall system.
- **Technical issues** - Describes any design or implementation issues involved in satisfying this requirement.
- **Dependencies with other requirements** - Describes interactions with other requirements.
- **Others as appropriate**

4. System Architecture

This section describes a high-level overview of the anticipated system architecture showing the distribution functions across (potential) system modules. Architectural components that are reused or 3rd party should be highlighted.

5. High-Level Design

This section should set out the high-level design of the system. It should include one or more system models showing the relationship between system components and the systems and its environment. These might be object-models, DFD, etc.

6. Preliminary Schedule

This section provides an initial version of the project plan, including the major tasks to be accomplished, their interdependencies, and their tentative start/stop dates. The plan also includes information on hardware, software, and wetware resource requirements. The project plan should be accompanied by one or more PERT or GANTT charts.

7. Appendices

Specifies other useful information for understanding the requirements.