ParkFinder
Software Requirements Document
SE 3A04

Abdul Ahad   Salma Belal   Josh Chatten   Nathanael Jordan
akhteraa   belalsm   chattejj   jordanen

Robert Stuart
stuarr2

March 5, 2016

Contents

1   Introduction

1.1   Purpose

The purpose of this Design Document is to provide a description for the design of the Park Finder app. The description of the design will allow anyone who will be involved in the development of the system to proceed with an understanding of what is to be built and how it is expected to be built. This document provides a description of the system architecture, as well as diagrams that model the functionality of system, describe the key classes of the system, their interrelationship, and their responsibilities.

The intended readers of this document include all of the project's stakeholders. This includes the end-user, the software engineers, and the park authorities.

1.2   System Description

The software system being described in this document is called the ParkFinder app. This system will have datasets of information about parks from all over the world and will allow the client to use search methods in order to find parks based on the clients' desired attributes. The app is meant to be used anywhere in the world, provided an Android or iOS device with the app installed. This provides clients with an easier, faster, and more efficient way to look up parks and acquire information such as the location, facilities, activities, and rentals that the parks provide.

1.3   Overview

The remainder of this document will contain diagrams and information that will describe the details for the software system being built. This will include a use case diagram in Section 2, an analysis class diagram in Section 3, a description of the architectural design in Section 4, and CRC cards for all identified classes in Section 5.

2   Use Case Diagram

a) **Search for parks:** The user searches for parks. This is accomplished by consulting experts based on which park attributes were selected by the user.

b) **Browse park's listing** User browses a list of parks, this list can either be the result of a previous search action or a default list (all parks).

c) **Select park(s):** User selects a park or several parks from the list they were browsing, this displays additional park information to the user as well as the park(s) on a map if desired.

d) **Request nearest 5 parks:** User requests the five nearest parks to their current location.

e) **Swap or remove expert:** A developer attempts to swap or remove an expert from the system, the system requires authorization from a manager for the change to occur.
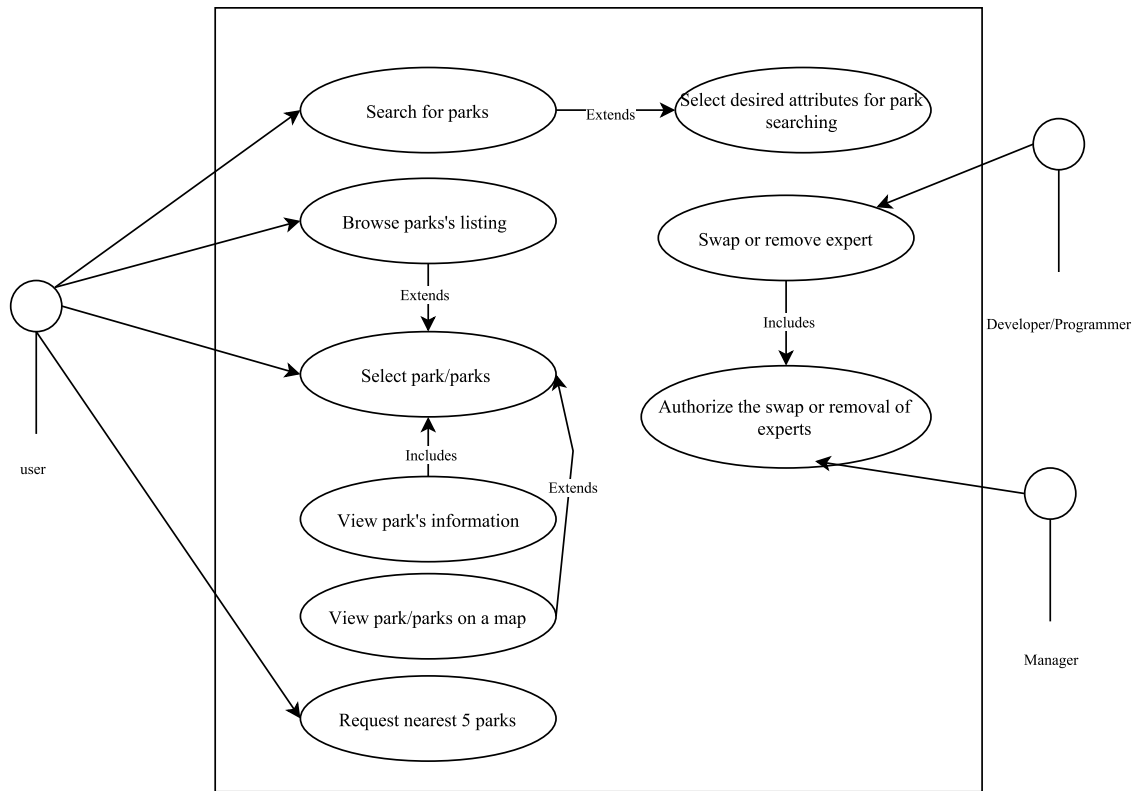
Figure 1: Use Case Diagram
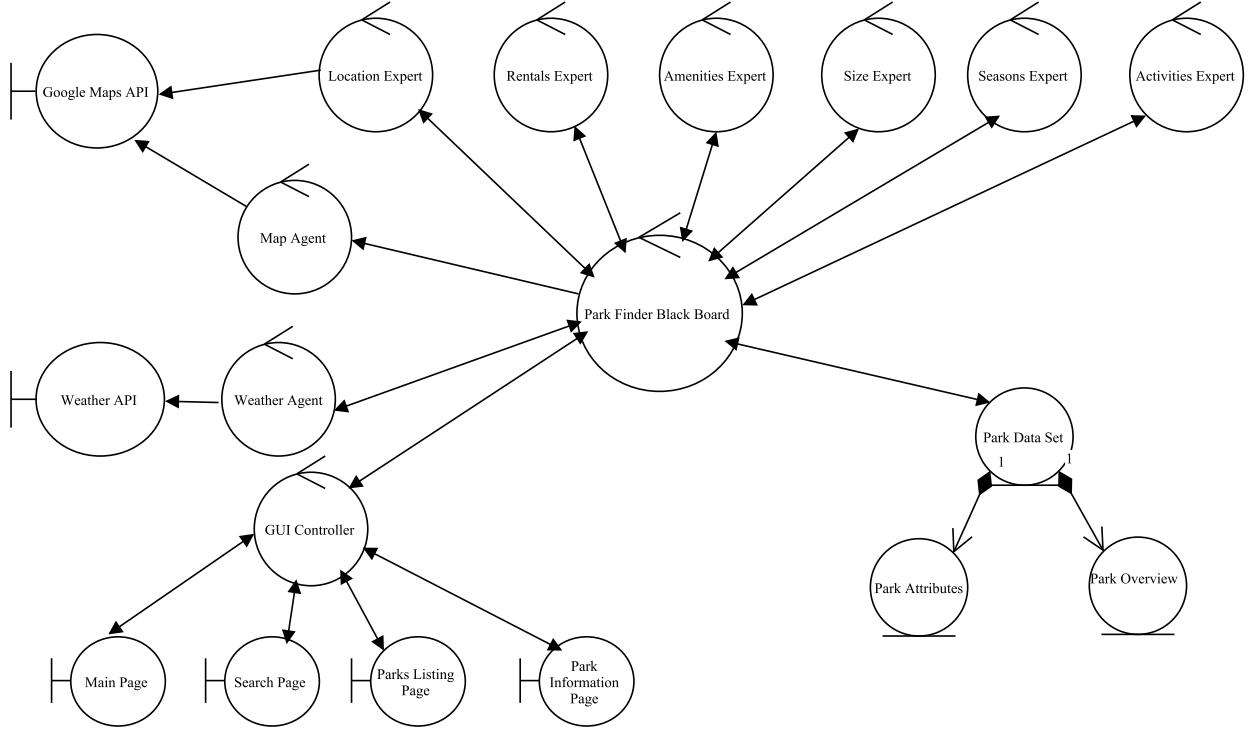
3   Analysis Class Diagram

Figure 2: Analysis Class Diagram

4   Architectural Design

4.1   System Architecture

The application uses a blackboard architecture which incorporates eight independent subsystems that all interact with the blackboard, the ParkFinder application. These independent subsystems are the Location Expert, Rentals Expert, Amenities Expert, Size Expert, Seasons Expert, Activities Expert, Map Agent and Weather Agent. The blackboard includes a data store for parks and also interacts with a controller.

By using the blackboard architecture style, a modularized and intuitive design is achieved by having independence between knowledge sources. This independence implies high cohesion and low coupling, allowing changes or updates to the knowledge sources with ease.

4.2   Subsystems

The system will be divided into several different subsystems that are shown in Figure 2. Each of these subsystems will handle different functionality of the overall system. The GUI and User subsystems will handle all interactions with the user of the application. The database will contain all of the information about each park in the system. The several

Expert subsystems will have similar functionality but handle different information. Each Expert subsystem will be provided search criteria and will return all parks which satisfy the criteria. The Weather Agent will provide the weather at any requested location. Lastly, ParkFinder will handle the information flow between all of the subsystems. ParkFinder will receive the user search criteria from GUI then provide that information and the database to the appropriate Expert(s) in order to identify the correct park.

5 Class Responsibility Collaboration (CRC) Cards

A   Division of Labour

Contributions	Name	Signature
x	Abdul Ahad	
x	Salma Belal	
x	Josh Chatten	
x	Nathanael Jordan	
x	Robert Stuart	

IMPORTANT NOTES

- Please document any non-standard notations that you may have used

  - *Rule of Thumb*: if you feel there is any doubt surrounding the meaning of your notations, document them

- Some diagrams may be difficult to fit into one page

  - It is OK if the text is small but please ensure that it is readable when printed
  - If you need to break a diagram onto multiple pages, please adopt a system of doing so and thoroughly explain how it can be reconnected from one page to the next; if you are unsure about this, please ask about it

- Please submit the latest version of Deliverable 1 with Deliverable 2

  - It does not have to be a freshly printed version; the latest marked version is OK

- If you do <u>NOT</u> have a Division of Labour sheet, your deliverable will <u>NOT</u> be marked