

ParkFinder
Software Requirements Document
SE 3A04

Abdul Ahad
akhterraa

Salma Belal
belalsm

Josh Chatten
chattejj

Nathanael Jordan
jordanen

Robert Stuart
stuarrr2

March 7, 2016

Contents

1	Introduction	2
1.1	Purpose	2
1.2	System Description	2
1.3	Overview	2
2	Use Case Diagram	2
3	Analysis Class Diagram	4
4	Architectural Design	4
4.1	System Architecture	4
4.2	Subsystems	5
5	Class Responsibility Collaboration (CRC) Cards	6
A	Division of Labour	12

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.

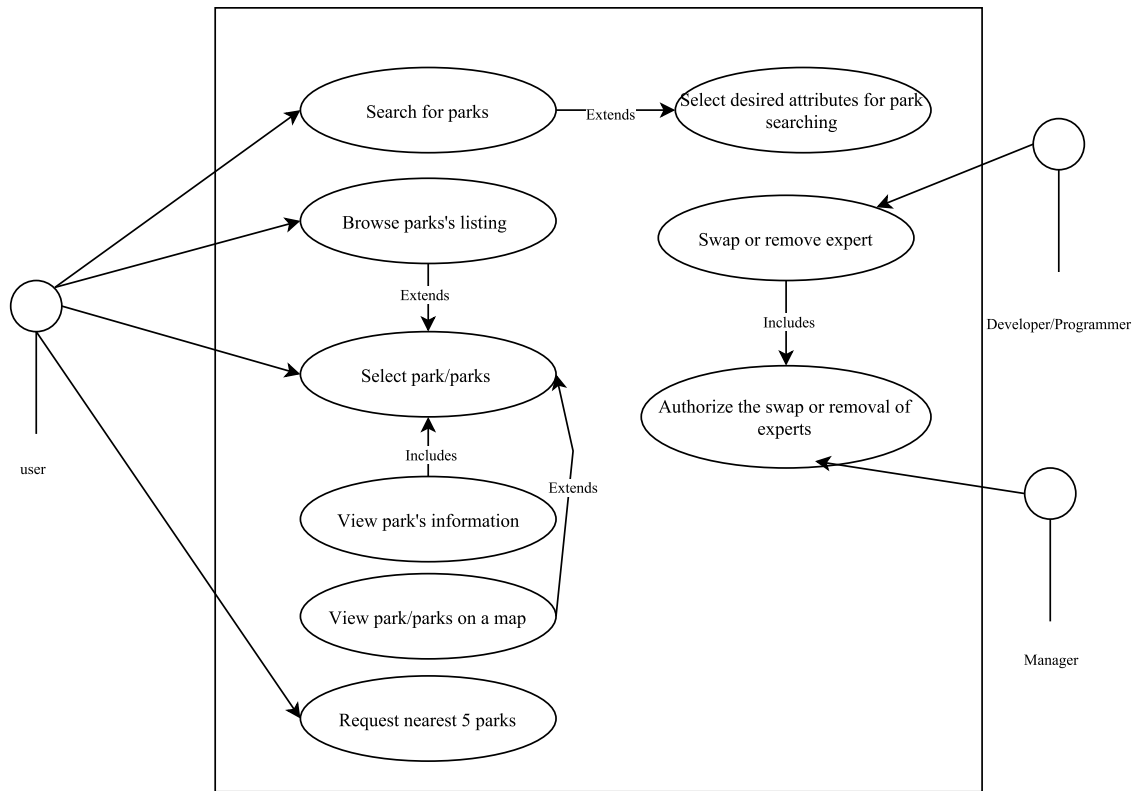


Figure 1: Use Case Diagram

3 Analysis Class Diagram

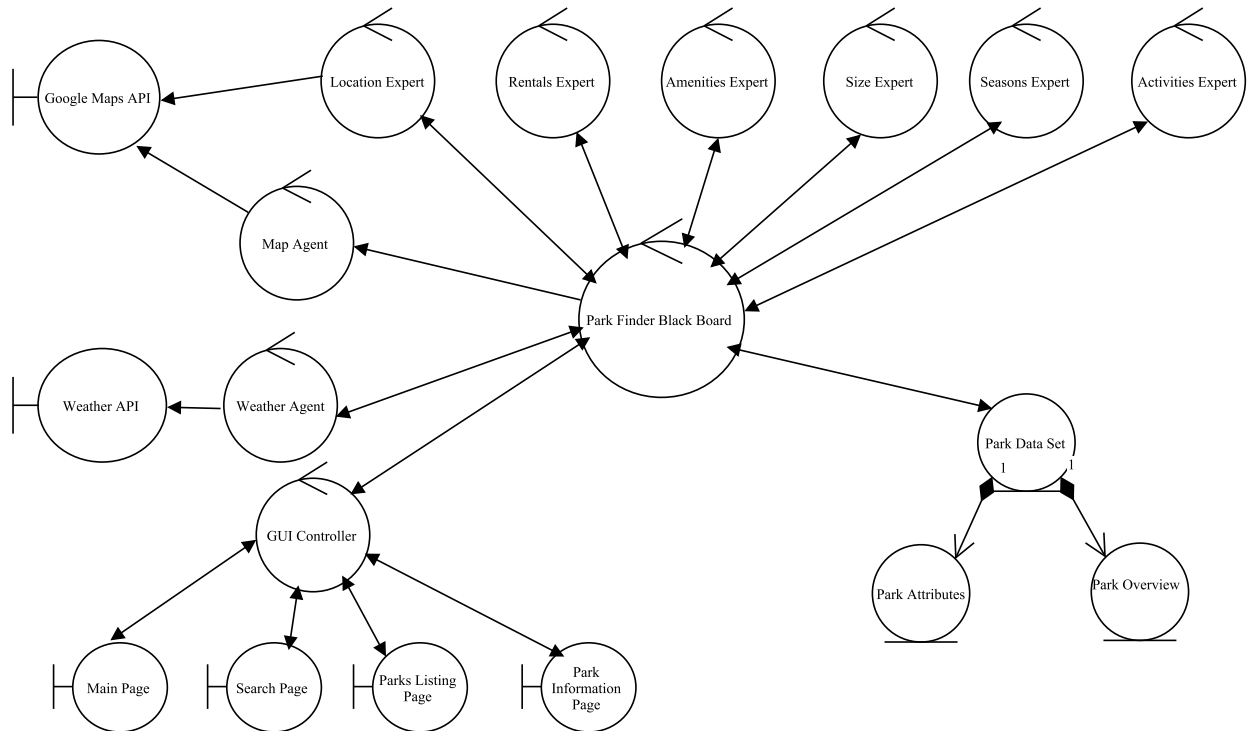


Figure 2: Analysis Class Diagram

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.

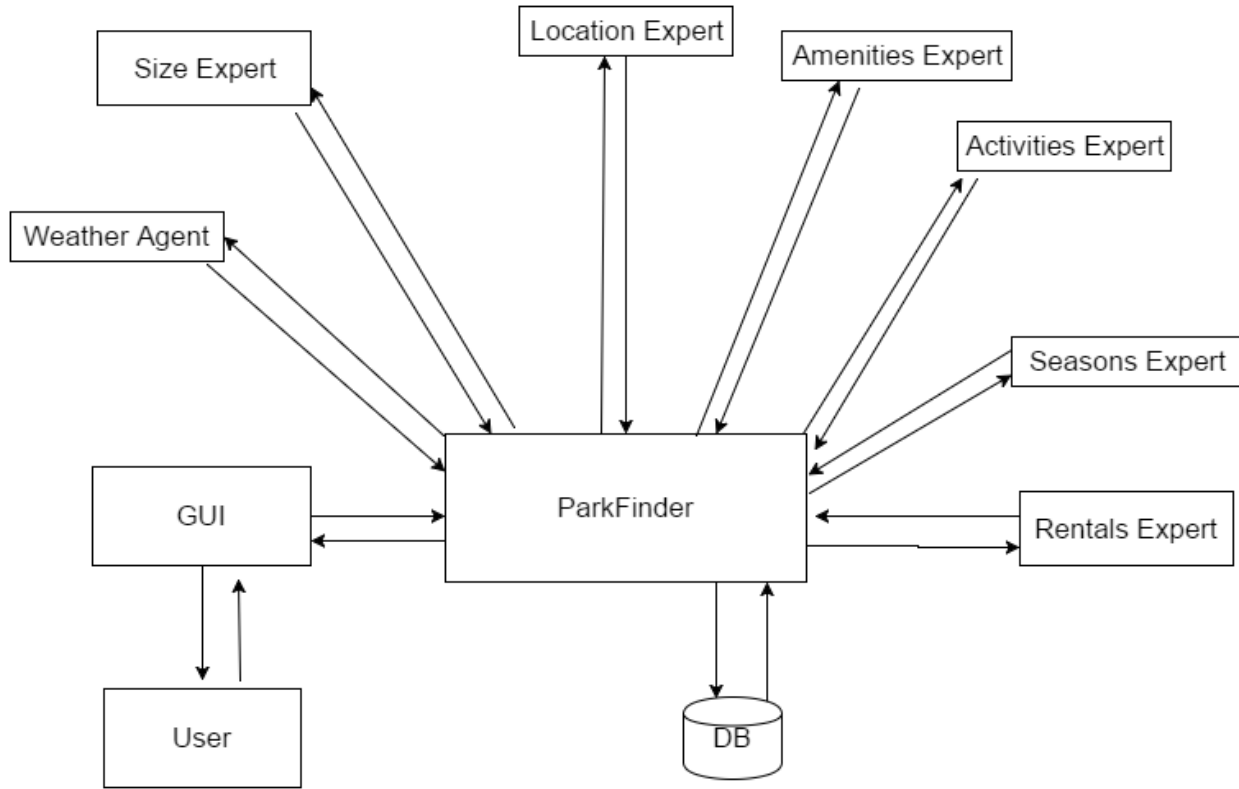


Figure 3: Structural Architecture Diagram

By using the blackboard architecture style, as seen above in Figure 3, 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

Class Name: Main Page	
Responsibility:	Collaborators:
Handles click events of "Start Search" Button	GUI Controller
Handles click events of View All Parks button	GUI Controller
Handles click events of Find Nearest 5 Parks button	GUI Controller

Class Name: Search Page	
Responsibility:	Collaborators:
Knows Search Criteria	
Knows GUI Controller	
Handles click events of selecting a search criterion	GUI Controller
Handles click events of the "Search" Button	GUI Controller
Handles click events of the "Back to main page" Button	GUI Controller

Class Name: Park Listings Page	
Responsibility:	Collaborators:
Knows GUI Controller	
Displays park names that are a result of the selected search criteria, or all available parks	GUI Controller
Handles click events for selecting a park	GUI Controller
Handles click events of the view the listed parks on a map Button	GUI Controller
Handles click events of the Back to search page button	GUI Controller

Class Name:Park Information Page	
Responsibility:	Collaborators:
Knows park name	GUI Controller
Displays park overview	GUI Controller
Displays park attributes	GUI Controller
Handles click events for the back to park listings page button	GUI Controller

Class Name:GUI Controller	
Responsibility:	Collaborators:
Knows Search Page	
Knows Main Page	
Knows Park Listings Page	
Knows Park Information Page	
Knows Park Finder Black Board	
Controls which pages are displayed to the user and when to switch from one page to another page	Search Page, Main Page, Park Listings Page, Park Information Page
Controls the parks being displayed in the park listings page	Park Listings Page, Park Finder Black Board
Controls the park information being displayed in the park	Park Information Page, Park Information Page Finder Black Board
Gives the users desired search criterion to the system	Search Page, Park Finder Black Board

Class Name:Park Data Set	
Responsibility:	Collaborators:
Knows Park	
Knows Park Attributes	
Knows Park Overview	
Knows Park Finder Black Board	
Owns Park Attributes	Park Attributes
Owns Park Overview	Park Overview
Allows Park Finder Black Board to read park data	Park Finder Black Board

Class Name: Park Overview	
Responsibility:	Collaborators:
Knows Park Data Set (Owner)	
Knows Park	
Knows Parks Overview (highlights of the park, the address, phone number, website, and operational dates)	
Each park overview belongs to the owner park	Park Data Set

Class Name:Park Attributes	
Responsibility:	Collaborators:
Knows Park Data Set (Owner)	
Knows Park	
Knows Parks attributes (amenities, size, seasons open, activities, and rentals)	
Each set of park attributes belongs to the owner park	Park Data Set

Class Name:Activities Expert	
Responsibility:	Collaborators:
Knows Activities	
Knows Park Finder Black Board	
Uses knowledge of activities to find parks with specified activities	Park Finder Black Board

Class Name:Amenities Expert	
Responsibility:	Collaborators:
Knows Amenities	
Knows Park Finder Black Board	
Uses knowledge of amenities to find parks with specified amenities	Park Finder Black Board

Class Name:Rentals Expert	
Responsibility:	Collaborators:
Knows Rentals	
Knows Park Finder Black Board	
Uses knowledge of rentals to find parks with specified rentals	Park Finder Black Board

Class Name:Size Expert	
Responsibility:	Collaborators:
Knows Sizes	
Knows Park Finder Black Board	
Uses knowledge of sizes to find parks with specified size	Park Finder Black Board

Class Name:Location Expert	
Responsibility:	Collaborators:
Knows Google Maps API	
Knows Locations	
Knows Park Finder Black Board	
Uses knowledge of locations to find nearest 5 parks	Park Finder Black Board, Google Maps API

Class Name:Seasons Expert	
Responsibility:	Collaborators:
Knows Seasons	
Knows Park Finder Black Board	
Uses knowledge of Seasons to find parks with specified operating season	Park Finder Black Board

Class Name: Weather Agent	
Responsibility:	Collaborators:
Knows Park Location	
Knows Weather API	
Knows Park Finder Black Board	
Uses knowledge of Park Location to get weather at that location	Park Finder Black Board, Weather API

Class Name: Park Finder Black Board	
Responsibility:	Collaborators:
Knows Weather Agent	
Knows Google Maps API	
Knows GUI Controller	
Knows Park Data Set	
Knows Location Expert	
Knows Rentals Expert	
Knows Amenities Expert	
Knows Size Expert	
Knows Seasons Expert	
Knows Activities Expert	
Knows the search criterion that the user requested and	All Experts, Search Page tells each expert its relevant search criteria
Knows the results of all search requests	All Experts, Park Data Set
Sends search results and park information to the GUI	GUI Controller, Park Data set
Accesses the weather agent and sends the retrieved information to the GUI	Weather Agent, GUI Controller
Sends park locations to the Map Agent	Map Agent, Park Data Set

Class Name:Map Agent	
Responsibility:	Collaborators:
Knows Park Location	
Knows Google Maps API	
Knows Park Finder Black Board	
Sends locations to the Google Maps API to display Parks On the Map	Park Finder Black Board, Google Maps API API

A Division of Labour

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