Software Requirements Specifications

Functional requirements

Searching

Public users should be able to search parking areas by providing basic details like the type of their car, Date, Time, duration, and also the city name which he wants to find. As a result, all the possible parking areas are displayed in form of a list and map. To book a particular parking area user can click on the know more option in the list and pin it in the map view. Detailed information about the selected parking should be visible.

Booking

Once the user has the parking area he should be able to book the parking area by providing details like the type of car, car number, duration, and days for which he wants to park. Once the data is submitted and successful payment is done the parking area is allocated for the user. Along with that the booked parking area details can be edited as well as canceled within the time-bound. User should be able to show their likeliness to the previously booked parking using ratings.

Subscription

Similar to the booking process but the user should be capable of selecting multiple days from the booking form to subscribe to the parking location. Also, the subscribed parking can be edited as well as canceled. Use should be able to rate the parking areas that he subscribed to.

New area registration

To register a parking area in the app, providers may use one registration form for enrolling their area into the system. The form takes data related to the area like the name of the parking, city, address, vehicle types, corresponding rates, location of the parking, etc. Once the submission is done the area will be verified.

Statistics

A provider should be able to analyze data to take decisions regarding the areas. The data can be like revenue generated or a rush of cars from the provided parking**.**

Manage provided Areas

All the parking areas that were added to the system should be displayed and these areas are editable so that the user can change the details like provided types of cars and corresponding rates, name of the service, etc. The area can be canceled too by giving the appropriate reason..

Slot tracing

The system in the parking area should be capable of keep tracking of all the parking slot along with checking whether it is empty or not.

Number plate recognition and send SMS

Once the car number is detected the new car coming into the parking should be notified with the detail of the slot number using SMS.

Nonfunctional requirements

Performance

The application should have high performance and low failure rates. The hardware and software should be able to transmit/receive data from databases with high baud rates, ranging from Mbps to Gbps.

Reliability

All the data related to users like profile information booked or subscribed parking, new parking area data, etc. should be persistence and not get lost.

Availability

The app should be accessible 24X7 so that users from different time zones can interact easily.

Basic understanding of mobiles

As the application is made for mobile devices, a user should have the basic knowledge to use it.

GUI should be interactive and user-friendly.

One of the basic needs for a better user experience is GUI. It should be not simple so the user gets bored but a decent presentation of data attracts the user the use it. Also, the features included in the app should be easily understandable by the user.

Technical requirements

System requirement

- Memory Space Required : Up to 600 MB
- A Webcam or External Camera
- Min. RAM : 1 GB

- **Operational environment**
  - **Operating system:** ios or android

Technologies

Python

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library

Python

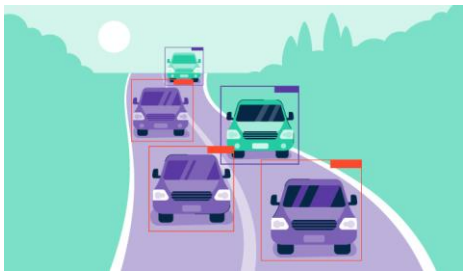
Open cv

OpenCV is the huge open-source library for computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today's systems. By using it, one can process images and videos to identify objects, faces, or even the handwriting of a human. This article focuses on detecting objects.

Open cv

Object Detection

Object Detection is a computer technology related to computer vision, image processing, and deep learning that deals with detecting instances of objects in images and videos.



Object detection

OCR

Optical character recognition or optical character reader (OCR) is the electronic or mechanical conversion of images of typed, handwritten, or printed text into machine-encoded text, whether from a scanned document, a photo of a document, a scene-photo (for example the text on signs and billboards in a landscape photo) or from subtitle text superimposed on an image (for example: from a television broadcast).

OCR

Firebase

Firebase provides the tools and infrastructure you need to develop, grow, and earn money from your app. This package supports web (browser), mobile web, and server (Node.js) clients.

Firebase Realtime Database - The Firebase Realtime Database lets you store and query user data, and makes it available between users in real time.

Cloud Firestore - Cloud Firestore is a flexible, scalable database for mobile, web, and server development from Firebase and Google Cloud Platform.

Firebase Storage - Firebase Storage lets you upload and store user-generated content, such as files, and images.

Firebase

React native

React Native is a popular open-source mobile application development framework created by Facebook. It uses JavaScript to create cross-platform mobile applications with true native capabilities. This means you can develop natively-rendered mobile apps for both iOS and Android simultaneously using a single codebase.

Benefits of Using React Native are…

1. Code Reusability

The biggest advantage of React Native is that developers don't need to create separate codes for different platforms (Android and iOS). In fact, around 90% of the code can be reused between the two platforms which helps increase development speed and efficiency considerably. As a result, you get faster time-to-market and require lesser maintenance efforts.

2. Native Look and Feel

React Native components map 1:1 with native development components. It combines the building blocks from the native user interface with its own JavaScript which gives the app a native-like appearance. Moreover, as the building blocks are the same for Android and iOS, the look and feel of the app are also similar across the two platforms.

3. Live Reload

The life reloads feature of React Native allows you to see and work with changes in real time. You can make fixes in the code while the app is loading and it will be reflected in the app with an automatic reload. You can also reload a particular area of change to save time on the compilation.

4. UI Focused

React Native uses the React JavaScript library to build app interfaces that are fast and responsive. It has great rendering abilities and uses a component-based approach which makes it easy to create both simple and complex UI designs.

- **Apps Built with React Native**

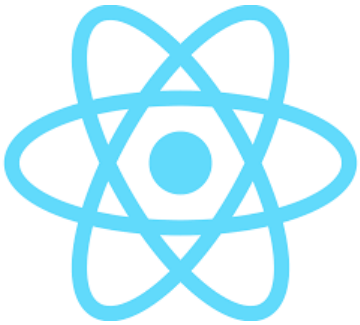
  Facebook

  Instagram

  Uber Eats

  Skype

  Pinterest



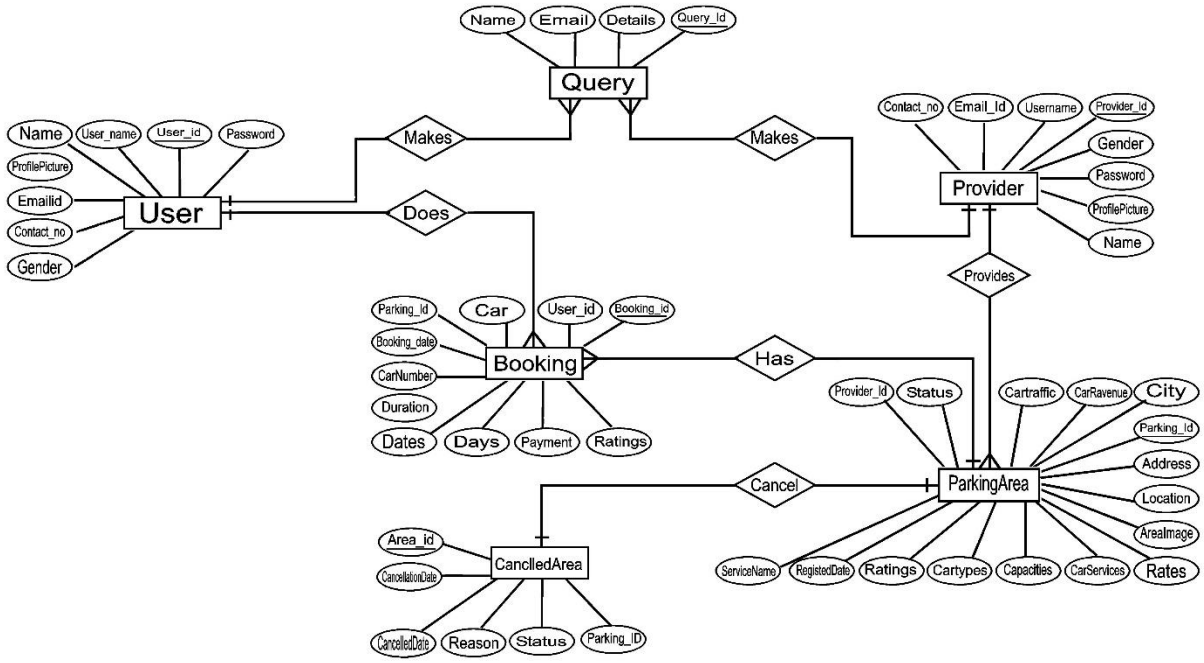
React native

Modeling and Design

For better visualization of implemented methodology, software design diagrams were constructed which provided a glimpse of the flow of control as well as the data in our project. The following diagrams proved useful information for a better understanding of this project.

- The ER diagram illustrates the entities that are involved in the application and also the type of data information that they are having.
- The Use case diagram tells about the user and their possible set of use cases. What different kinds of features the user can hold.
- The State Diagram indicates the possible transition between different states.
- The activity diagram will show all the activities carried out by the user to use the system efficiently and effectively for their own benefit.
- Data flow presents how the flow of the data in the application goes.
- The Sequence Diagram will show the simplest form of interaction between the user to the system.

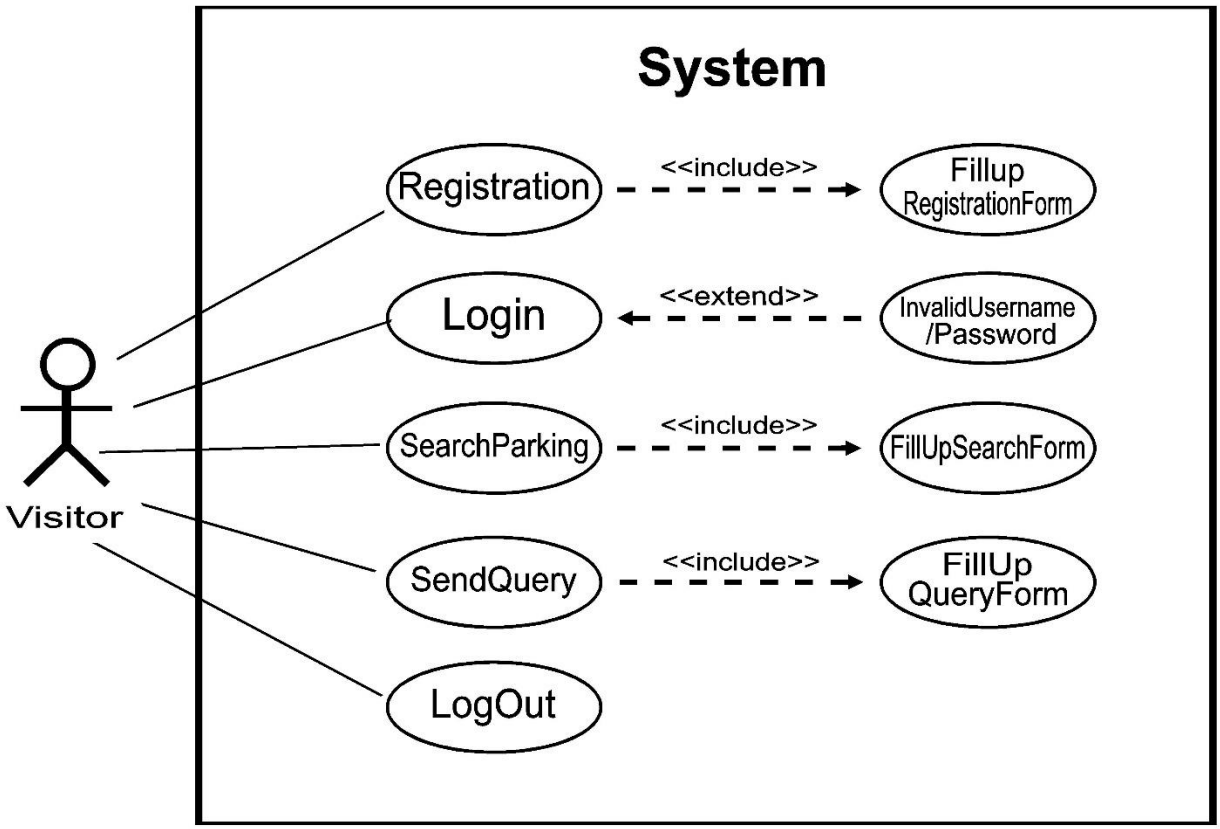
ER diagram



ER diagram

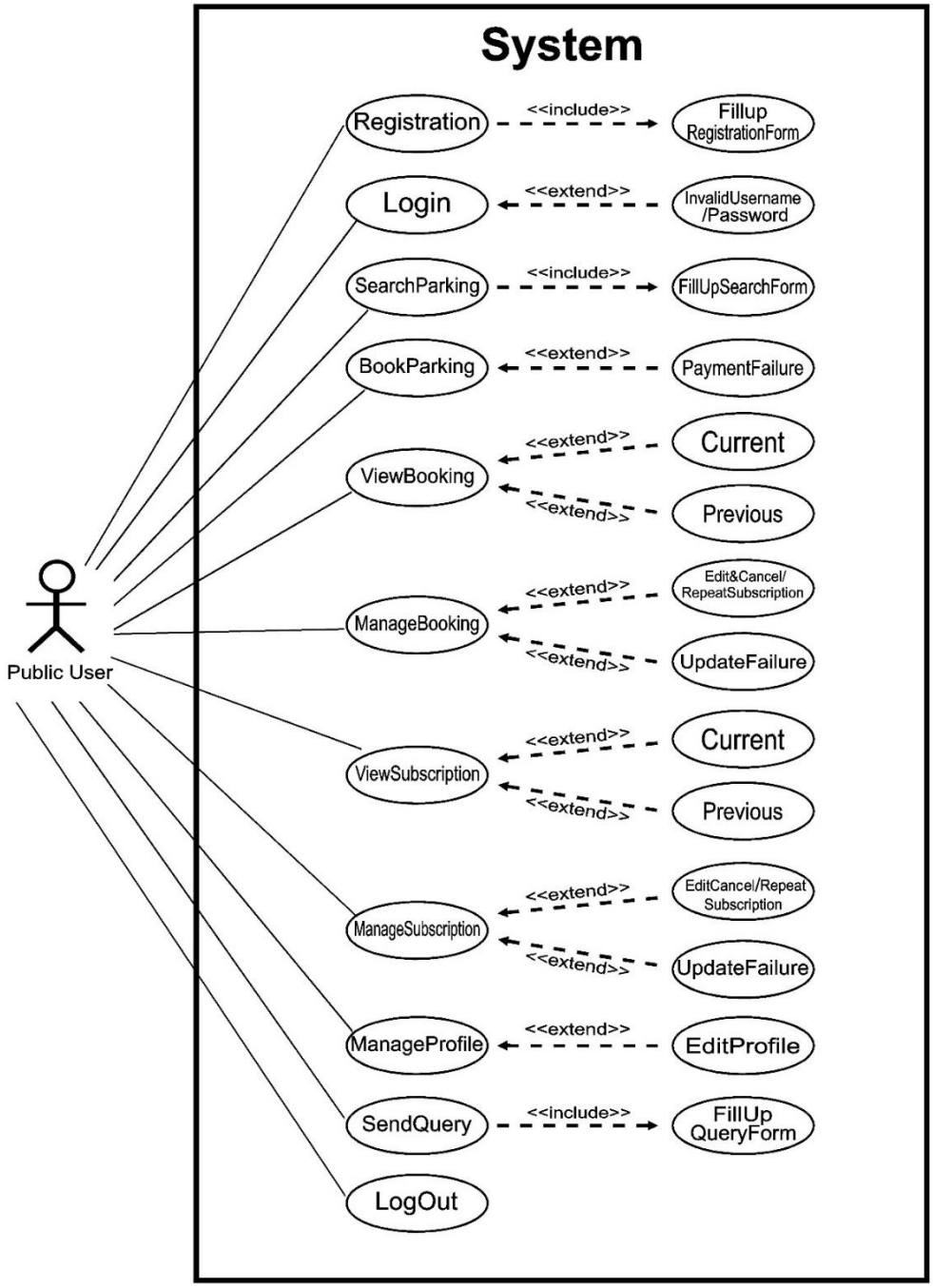
Use case diagram

Visitor



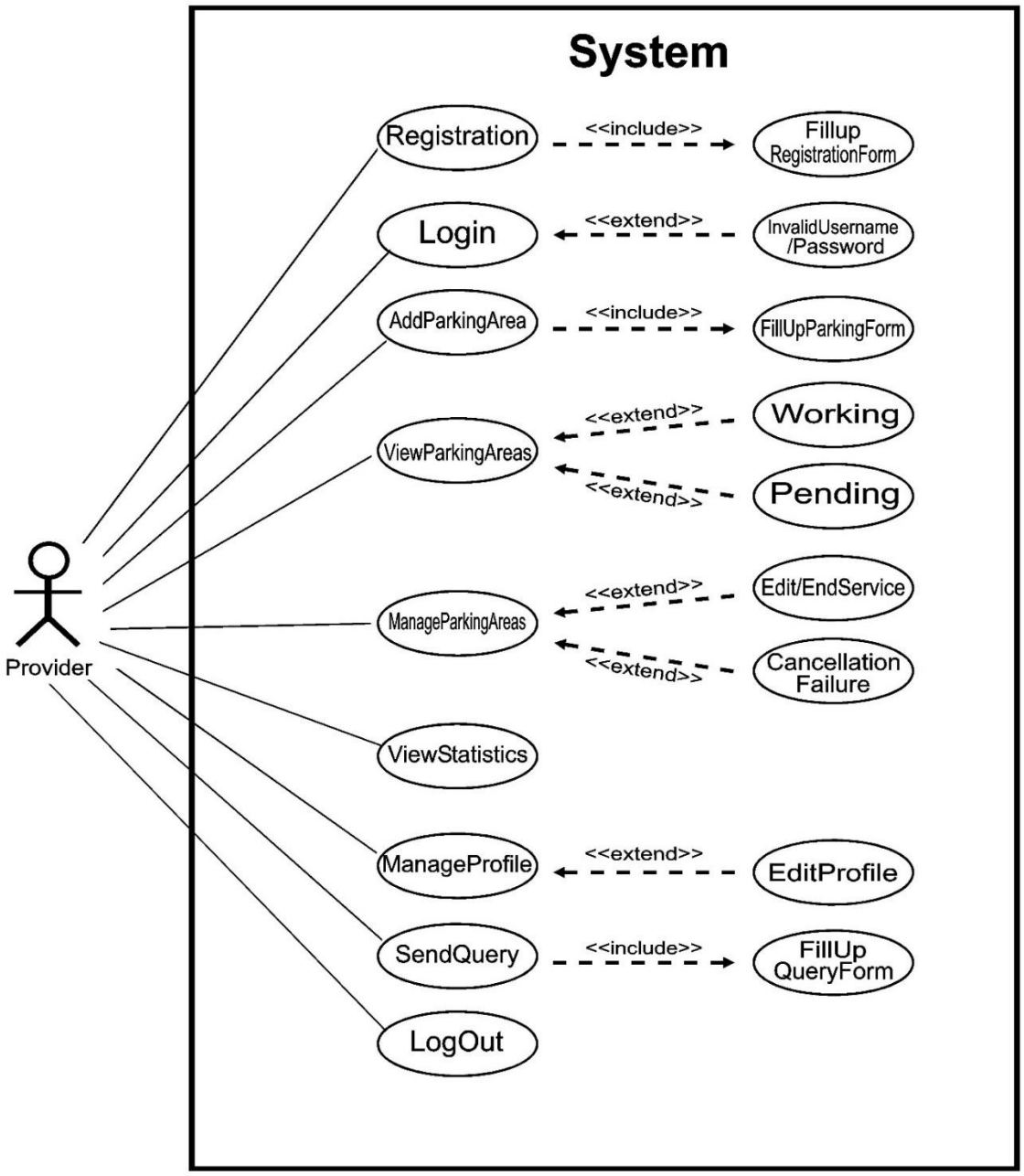
Usercasediagram_visitor

Public user



Usercasediagram_public user

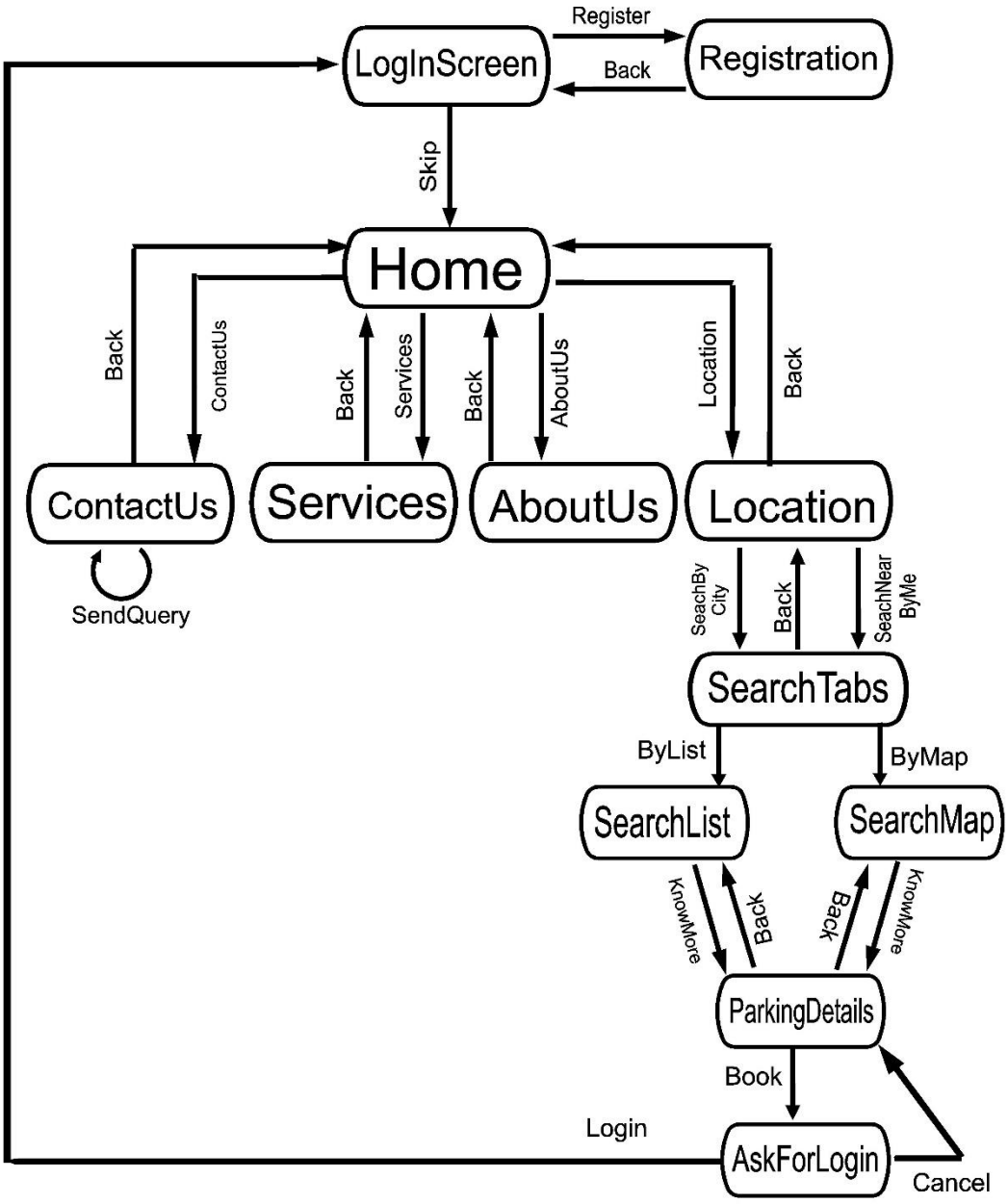
Provider



Usercasediagram_provider

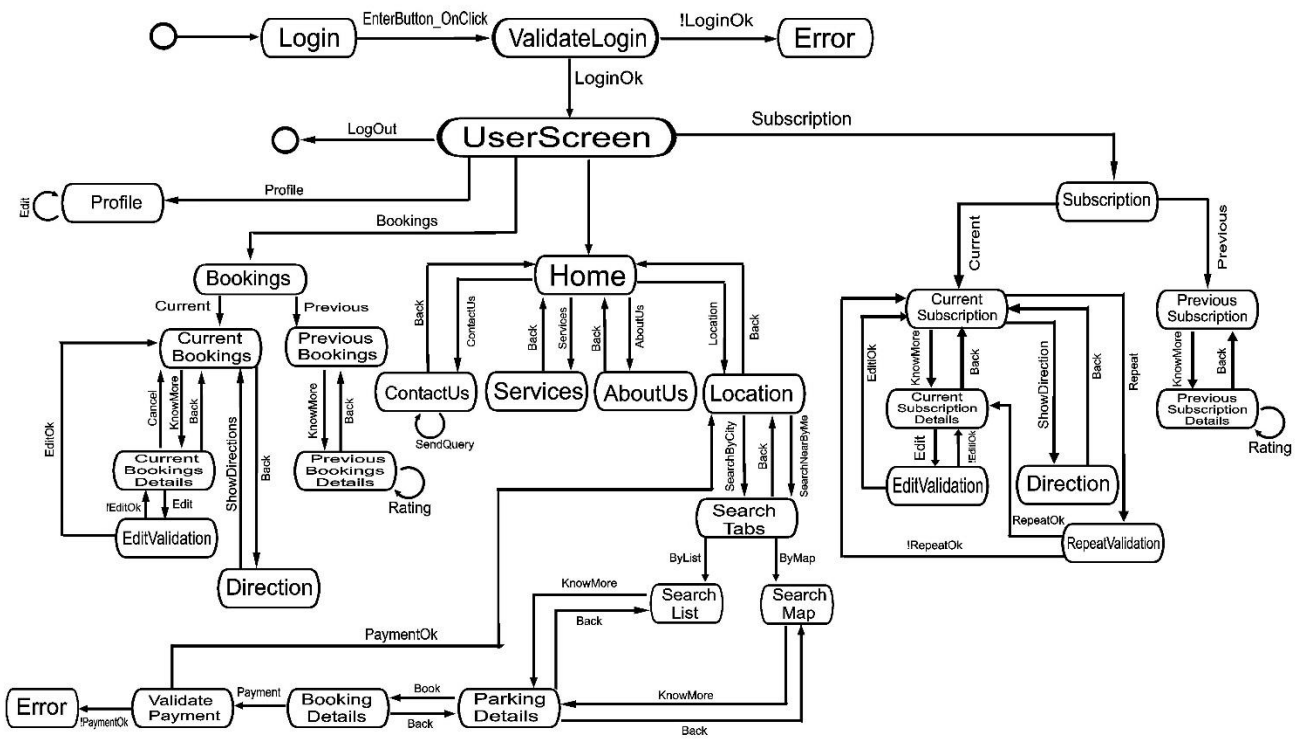
State Diagram

Visitor



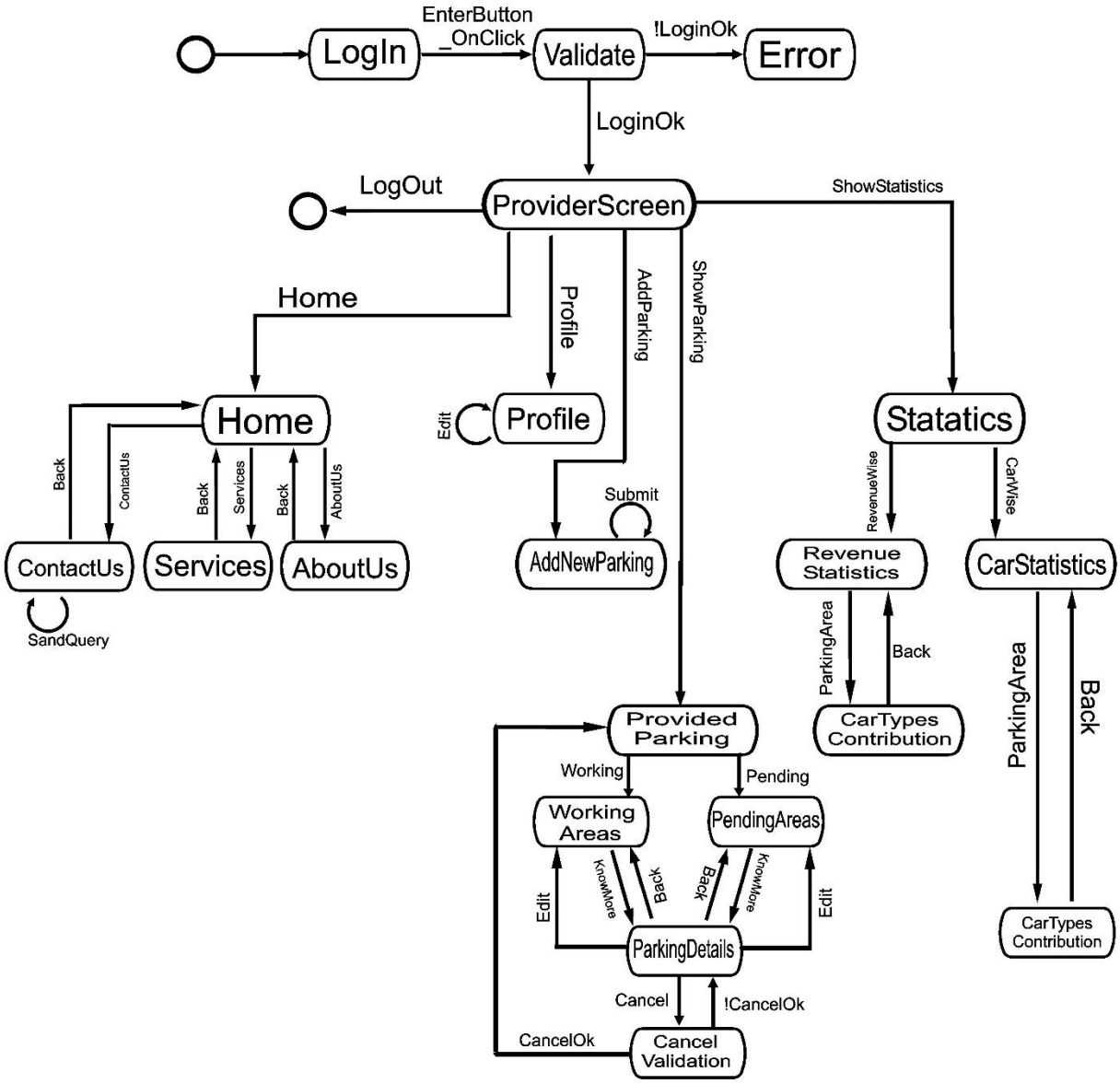
Statediagram_visitor

Public user



Statediagram_public user

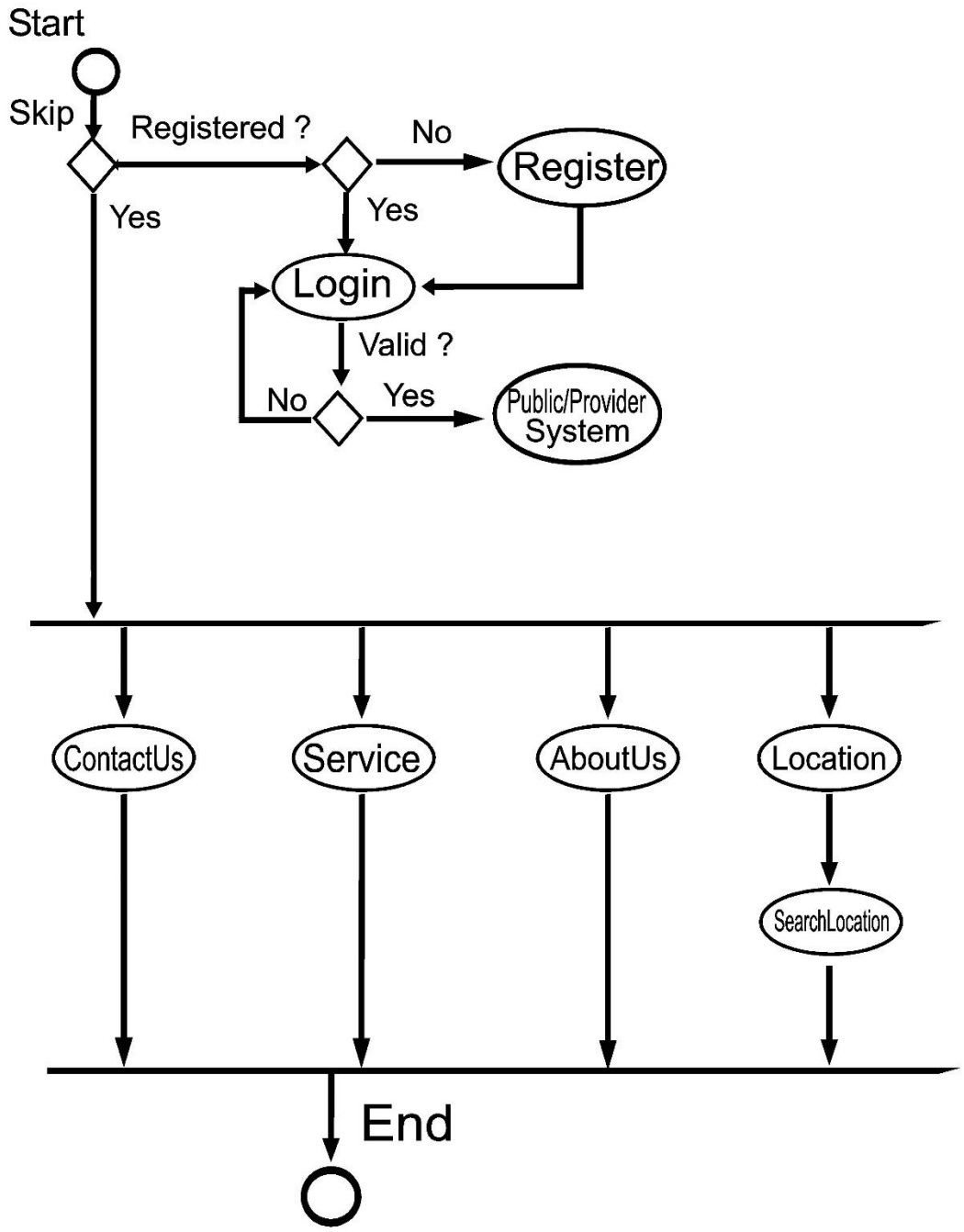
Provider



Statediagram_provider

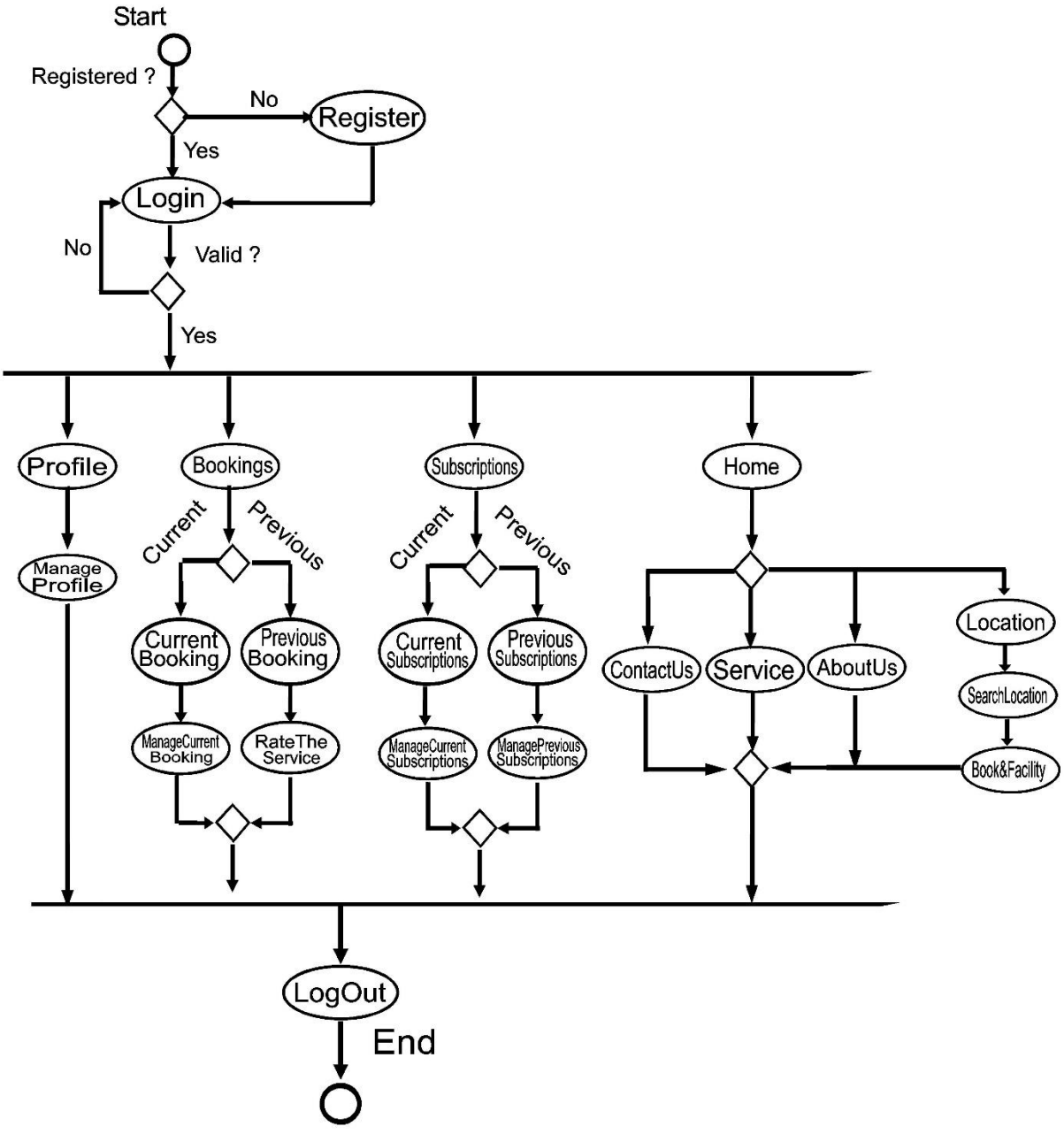
Activity Diagram

Visitor



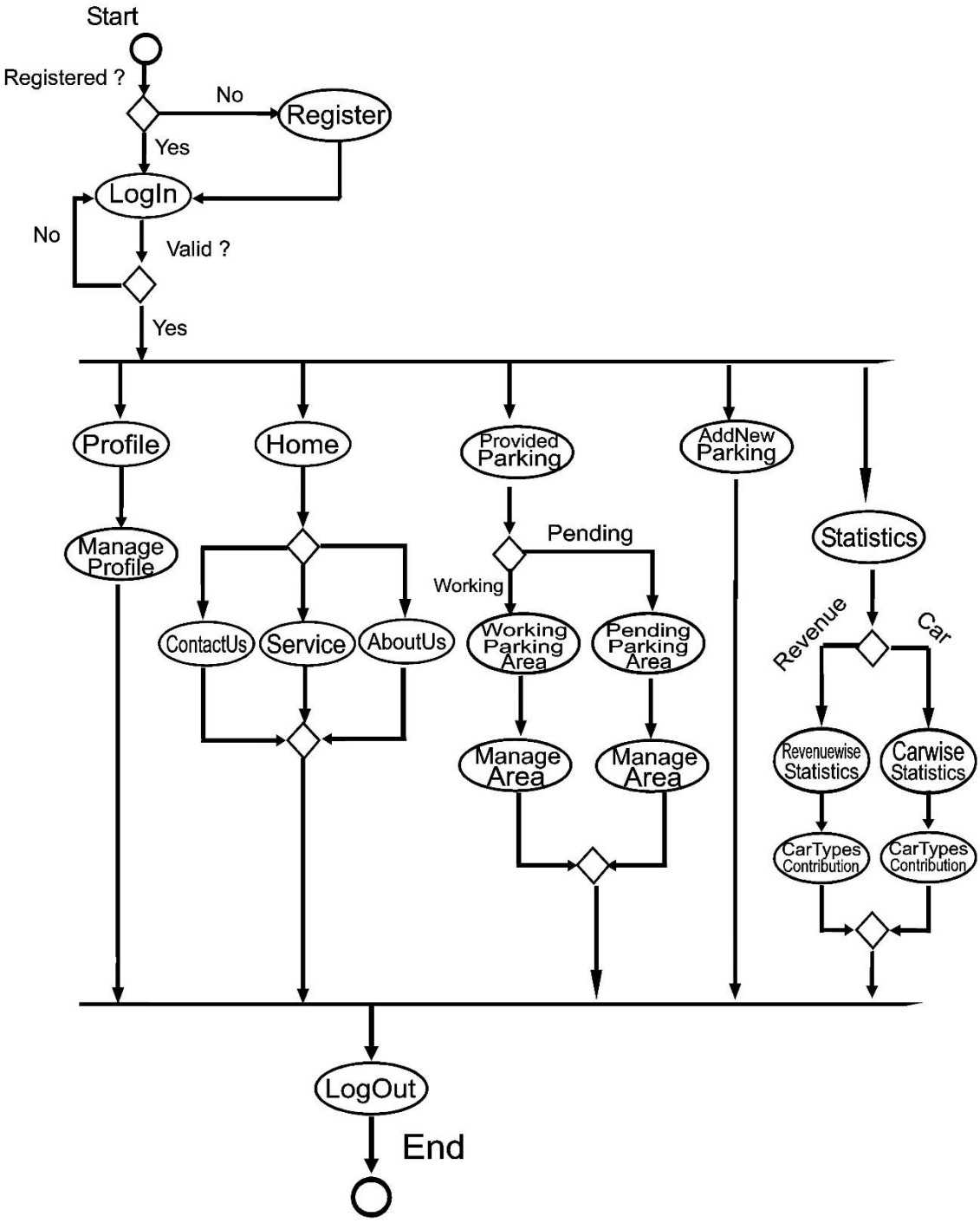
Activitydiagram_visitor

Public user



Activitydiagram_public user

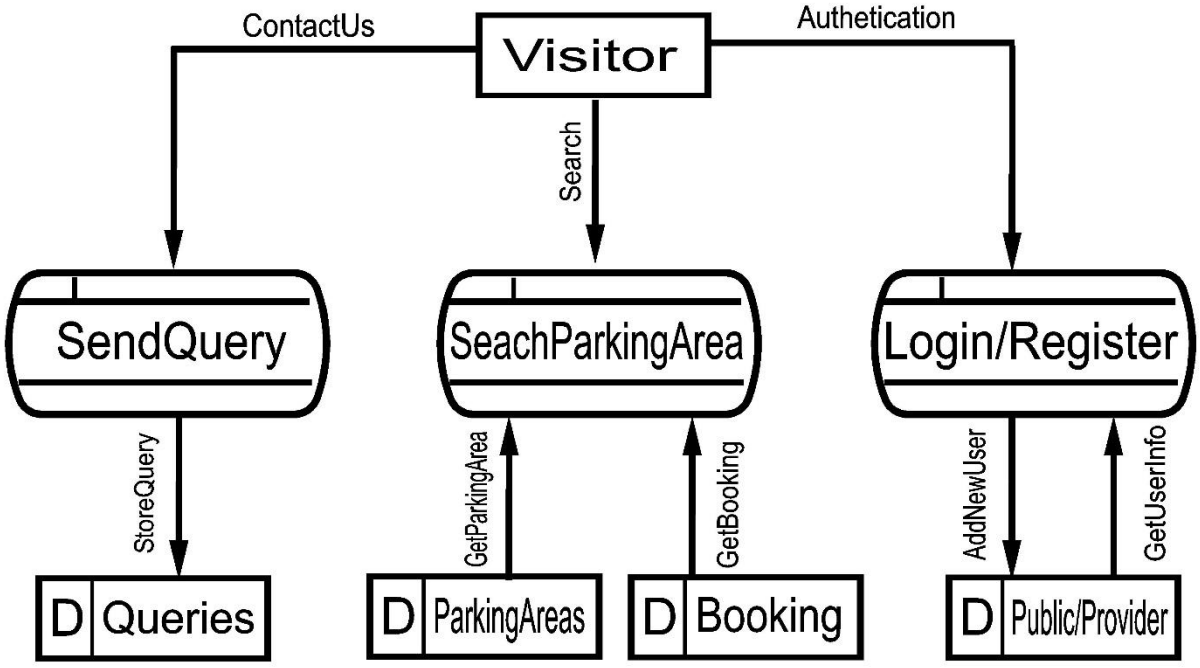
Provider



Activitydiagram_provider

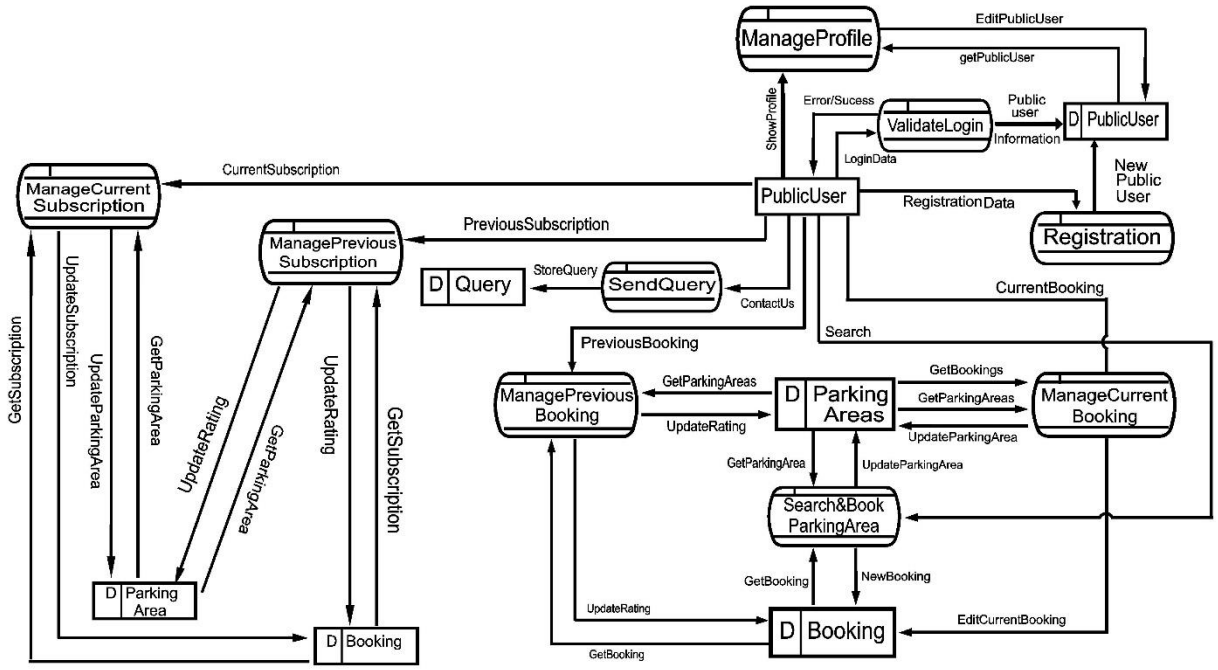
Data flow Diagram

Visitor



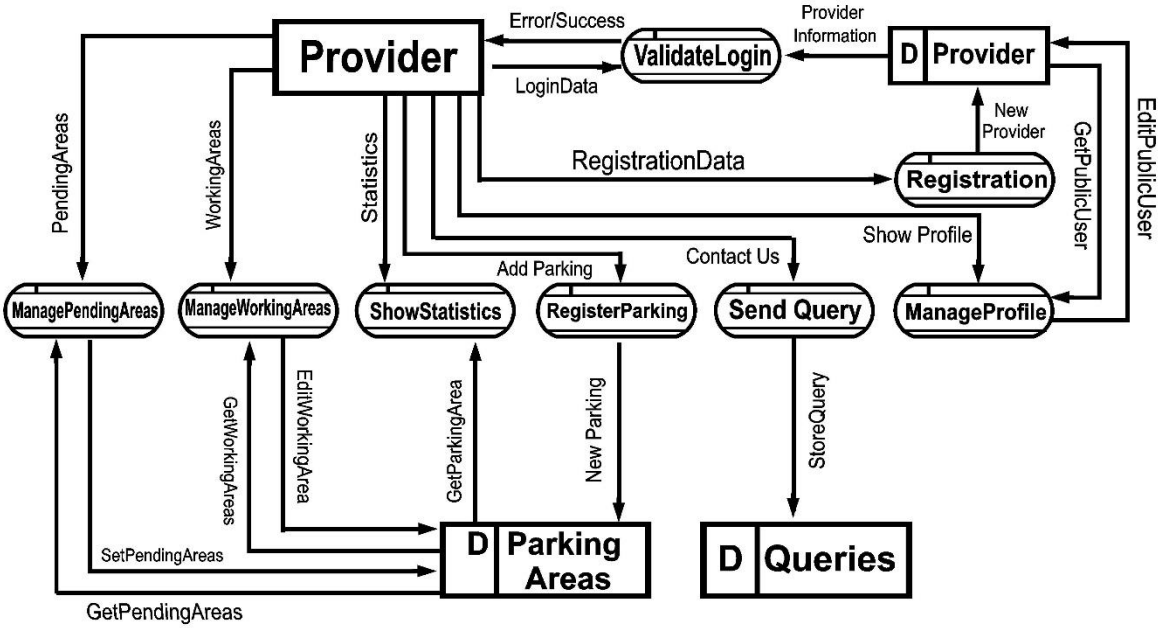
Dataflowdiagram_visitor

Public user



Dataflowdiagram_public user

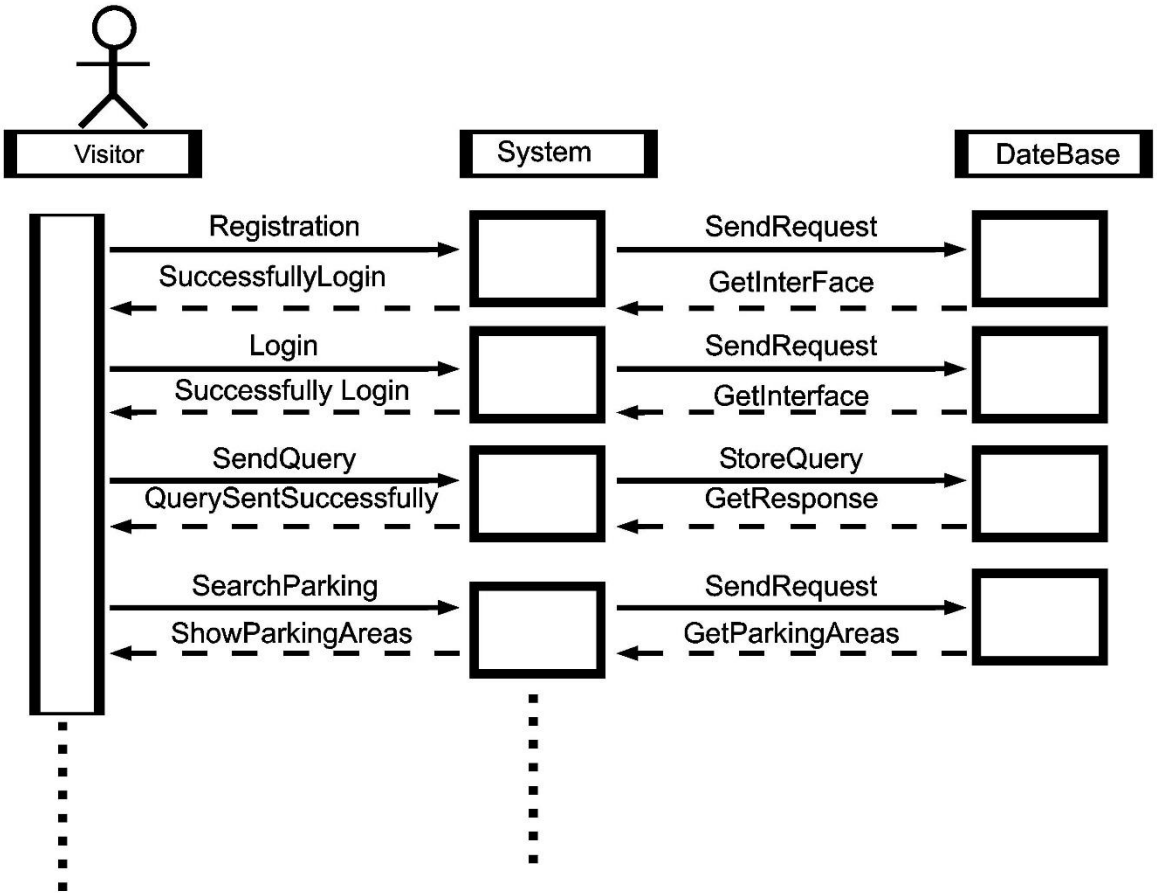
Provider



Dataflowdiagram_provider

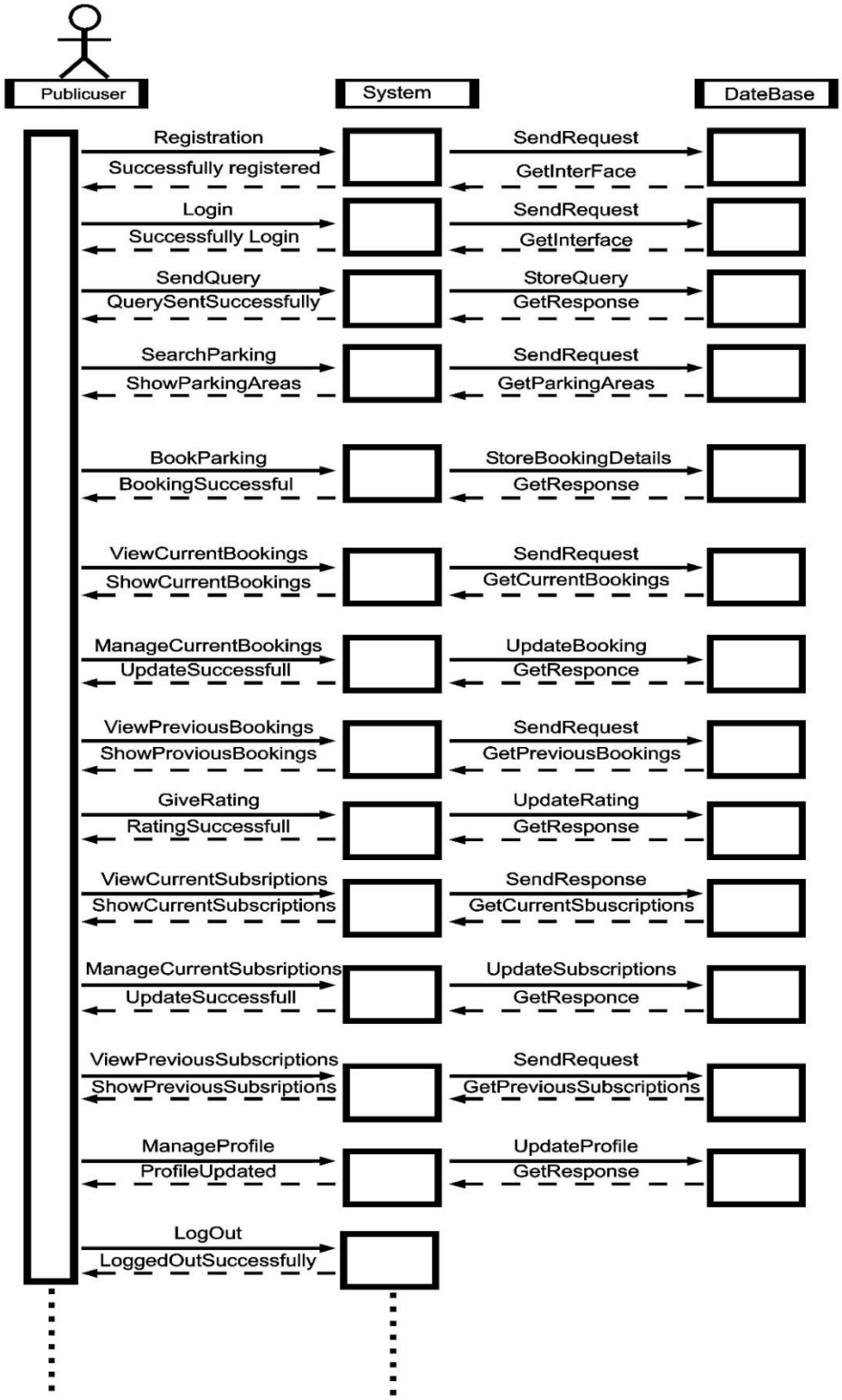
Sequence Diagram

Visitor



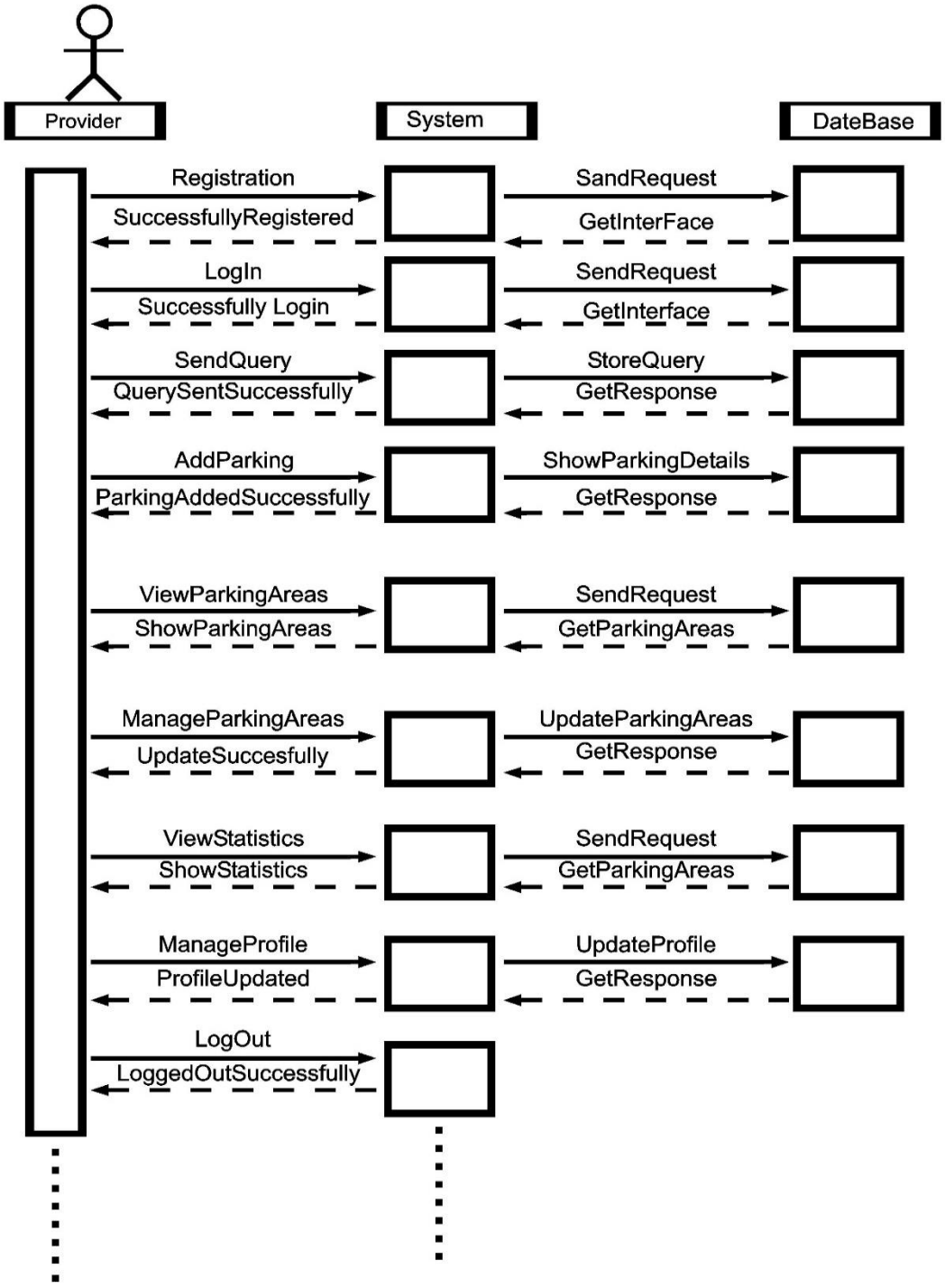
Sequencediagram_visitor

Public user



Sequencediagram_public user

Provider



Sequencediagram_provider