

Software Requirements Specification

For Locate-A-Socket

Version 1.2 Prepared by Joshua Kempster
Deakin University
07/01/2026

Table of Contents

- [1. Introduction](#)
 - [1.1 Document Purpose](#)
 - [1.2 Product Scope](#)
 - [1.3 Document Overview](#)
 - [1.4 Definitions, Acronyms, and Abbreviations](#)
- [2. Product Overview](#)
 - [2.1 Product Perspective](#)
 - [2.2 Product Functions](#)
 - [2.3 User Characteristics](#)
 - [2.4 Product Constraints](#)
 - [2.5 Assumptions and Dependencies](#)
- [3. Specific Requirements](#)
- [4. Supporting Information](#)

Revision History

Name	Date	Reason For Changes	Version
Joshua Kempster	17/01/2026	Initial Draft	0.1
Joshua Kempster	18/01/2026	Clean up, ready for submission	1.0
Joshua Kempster	19/01/2026	Shorten document to 4 pages	1.1
Joshua Kempster	19/01/2026	Add table of contents back in	1.2

1. Introduction

1.1 Document Purpose

This document specifies the software requirements of the Locate-A-Socket software. It does not provide details about how the requirements will be implemented – simply WHAT the system must do. It is intended for a broad audience – software architects, developers and testers as well as stakeholders who will be involved in the design and maintenance.

1.2 Product Scope

This product is Locate-A-Socket v1.0. Its purpose is to assist electric car drivers to find and use charging stations near them or along their route. It will allow drivers to use location-based services to navigate to sockets and provide a secure payment platform that enables them to complete charging transactions within the application. It will also make prices, charging speeds, availability and other useful information accessible to the driver to inform their decision to use the station.

The management of a grid of sockets is out of scope.

1.3 Document Overview

This document is structured as follows:

- Section 2: Product Overview. This section includes background and context that influences the requirements for Locate-A-Socket.
- Section 3: Specific Requirements. This section provides detail about the product including external interfaces, and functional and non-functional requirements.
- Section 4: Supporting Information. This section contains additional information, currently consisting only of references.

Note that there is currently no detail about Verification/Testing Requirements and no diagrams as these are out of scope for the initial draft required for assessment.

1.4 Definitions, Acronyms, and Abbreviations

Term	Definition
API	Application Programming Interface - A set of definitions and protocols for building and integrating application software
GDPR	General Data Protection Regulation - EU regulation on user privacy
OCPP	Open Charge Point Protocol – a standard to allow management software to interact with sockets
SLA	Service Level Agreement
Socket	A charging station for an electric vehicle
SRS	Software Requirements Specification - A document that describes the intended purpose, requirements, and nature of a software
UI	User Interface - The visual part of computer application through which a user interacts with a software

2. Product Overview

2.1 Product Perspective

The Locate-A-Socket service is a new product that fits a requirement within an existing, growing landscape. The application will be owned by Locate-A-Socket. It will need to interface externally with existing sockets provided by a range of companies, both to source information from them, and to unlock and interact with them. It will also interface with a payment gateway.

The service will be hosted on cloud-based infrastructure and users will interact with it via existing operating systems and browsers. It will need to be supported by both mobile and desktop: Windows, MacOS, Android and iOS.

2.2 Product Functions

The product will provide the following main functions:

- Account management
 - User registration and login
 - Account and personal detail management
 - Saved routes, sockets and payment methods
- Taking secure payments
- Map visual to find and navigate to sockets
- Reliable search function for finding sockets, including location-based
- Information about the sockets, including price, speed and availability
- Unlocking and accessing the sockets

2.3 User Characteristics

General Users: Drivers who are the main users of the application for finding and using sockets.

- Expertise: Low to high – a range of expertises, generally expected to be able to navigate a web application and payment systems
- Access level: To customer-facing functions only
- Frequency of use: Weekly, but high volume of users
- Accessibility needs: Ensure accessibility features are available

Admin: Professionals who manage the app, user accounts, fix bugs and ensure functionality.

- Expertise: High technical expertise
- Access level: All
- Frequency of use: Very regular, but low volume of users
- Accessibility needs: N/A

Socket Suppliers: Professionals who want to add, remove or update information about the sockets they manage.

- Expertise: Medium technical expertise, generally expected to be able to navigate a web application, but not necessarily interact programmatically with a backend.
- Access level: Customer-facing and supplier-facing functions
- Frequency of use: Low to medium regularity, low volume of users
- Accessibility needs: Ensure accessibility features are available

Support Staff: Professionals who provide support to drivers and socket suppliers, walking them through functionality, providing workarounds for bugs, or escalating issues.

- Expertise: High technical expertise
- Access level: All customer-facing functions, plus account management and backend information
- Frequency of use: Very regular, low volume of users
- Accessibility needs: Ensure accessibility features are available

2.4 Product Constraints

- Must operate in all major browsers – Chrome, Safari, Firefox, Edge, etc.
- Must integrate into the existing sockets
- Must support latest iOS and Android versions
- Must support payment gateway
- *Prefer* support for Android Auto and Carplay

2.5 Assumptions and Dependencies

Operational Assumptions

- Users have a reliable internet connection and use modern web-browsers that support location services
- Users will have basic technological literacy and familiarity with web-based applications.
- There will be organisational support for ongoing customer engagement staff

Technical Assumptions

- All upstream systems (e.g., databases, third-party APIs, payment gateways) will be available according to SLAs.
- The organisation will support the technical implementation, maintenance, and scaling of the application.

Environmental Assumptions

- The legal and societal pressures that are driving the change to electric vehicles will continue.

Dependencies

- Reliance on up-to-date information from socket suppliers, either programmatically, or via a management interface.
- Reliance on a secure payment platform
- Potentially dependent on the ongoing maintenance of the OCPP.

3. Specific Requirements

Integration with Payment Gateway (Interface and Functional Requirement) The application shall use a payment platform for authorisation and charging for socket-use. Acceptance criteria:

- As a user, I want to easily and reliably make payments online, so that I can interact with a socket entirely from within the app. – As a socket supplier, I want to receive payments, so that I can offer the customer the charging service.
- As a company owner/executive, I want to reliably receive payments, so that my company is profitable.
- As an IT support staff, I want integration with the payment platform to be seamless and backward compatible, so that the number of errors and missed payments is minimised.

Integration with Socket Information (Interface and Functional Requirement) The application shall ingest information about sockets that can be displayed to the user. This includes: name, id, branding, charging speed, price, location and number of available sockets at the location. Acceptance criteria:

- As a user, I want to know where a socket is and if it is available, so that I can decide whether to navigate to it.
- As a user, I want to know the price of a socket, so that I can compare it to others and decide which best fits my needs.
- As a user, I want to know the charging speed of a socket, so that I can compare it to others and decide which best fits my needs. – As a socket supplier, I want my branding on the sockets, so that I can build brand recognition and loyalty.
- As a customer support staff, I want this information available to a customer, so that the number of calls requesting information is reduced.

Map Support (Interface and Functional Requirement) The application shall integrate with a digital map supplier that can be used as the base for displaying sockets and navigating to them. Acceptance criteria:

- As a user, I want to visually locate and navigate to a socket, so that I can use it.
- As a developer, I want integration with the map software to be robust and backward compatible, so that I minimise code updates

User Support (Functional Requirement and potentially Interface) The application shall provide a chat, phone or some other communication functionality to provide a channel for user support. Acceptance criteria:

- As a user, I want to easily get support for an issue, so that I can solve my problem and return to using the service quickly.
- As a customer support staff, I want the customers to easily get in touch, so that I can best resolve their issues.

Socket Update Interface (Functional Requirement and potentially Interface) The application shall include an interface to allow updates to socket information. Acceptance criteria:

- As a socket supplier, I want to ensure that my socket information is up to date, so that I can provide the correct service to users.
- As a user, I want to have up-to-date information, so that I can make correctly informed decisions about which sockets to drive to.

Account Management (Functional Requirement and potentially Interface) The application shall include an account sign up and login system to manage the users personal payments, saved locations and history. This may include integration with third-party login services such as Google. Acceptance criteria:

- As a user, I want to easily make and log in to an account, so that I can keep my personal information saved for each transaction or interaction.
- As a user, I want an account, so that I can save my favourite charging locations.
- As a data analyst, I want users to have accounts, so that I can analyse their interactions and history, and link it to the correct user.

Security (Non-Functional Requirement) The application shall keep users' personal information, especially payment information, secure and private, as per relevant privacy frameworks, such as GDPR. Acceptance criteria:

- As a user, I want to be assured that my information is kept safe, so that my identity and capital is not threatened
- As a company executive, I want to ensure there are no security breaches, so that the viability and reputation of the company is not affected

Availability (Non-Functional Requirement) The application shall maintain daytime-uptime (5 AM - 12AM) of >99.5%. Acceptance criteria:

- As a user, I want the application to be available whenever I need to charge my vehicle, so that I am not left high and dry.
- As a socket supplier, I want the application to be highly available, so that my business is not affected, and I do not waste time trying to resolve customer issues.
- As an IT or customer support staff, I want the application to be highly available, so that my time is not wasted fixing issues and responding to customer complaints.

4. Supporting Information

Apps reviewed:

- Octopus Energy (2026) *Octopus Electroverse [mobile app]*, accessed 16 January 2026. <https://electroverse.com/>
- Zap-Map Ltd (n.d.) *Zap-Map EV charging app [mobile app]*, accessed 16 January 2026. <https://www.zapmap.com/>
- Recargo, Inc. (2026) *PlugShare [mobile app]*, accessed 16 January 2026. <https://www.plugshare.com/>

Template used:

- Montoya, J. (jam01) (2025) *SRS-Template: A markdown template for Software Requirements Specification based on IEEE 830 and ISO/IEC/IEEE 29148:2011 [GitHub repository]*., accessed 16 January 2026. <https://github.com/jam01/SRS-Template>.

Other References:

- Open Charge Alliance 2026, *Open charge point protocol*, accessed 17 January 2026, <https://openchargealliance.org/protocols/open-charge-point-protocol/>.