Requirements

The following document outlines the requirements for the Online Plant Store System (OPSS). To ensure that all corners of the requirements finding process were covered, the **FURPS+** model to assess functional and on-functional requirements was considered.

Functional

Requirement ID :	**FR001:** Select Item
Definition:	The system shall display detailed information when a user selects an item from the catalogue.
Specification:	- On click of a product in the BrowsePanel the system should: - Form a SELECT query on the Product table - Display a new ProductPanel with the following using JLabel's: 200x200 image of the product, product name, price and description

Requirement ID :	**FR002:** Cart Management
Definition:	Registered users may add products to their cart, from which they may alter the product quantity via a JSpinner
Specification:	- On click of the "Add to Cart" button the system must: - Validate the user is first logged in - Update the users cart object to display the product quantities and total price

Requirement ID :	**FR003:** Checkout Process
Definition:	Users complete the Order by initiating a checkout process, validated by inputting payment details
Specification:	- On click of the "Checkout" button the system must: - Build a form to input: Card Number, Card Holder, Address, CVV and Expiration Date (via JComboBox's) - On submit, the system will generate an INSERT query into the Orders table

Requirement ID :	**FR004:** Browse Catalogue
Definition:	The system shall display a populated catalogue of items with a scrollable UI
Specification:	- A JPanel displaying a series of product item containers which hold information about each product in the Product table. - Products are retrieved via a SELECT query in the Products table

Requirement ID :	**FR005:** Order History
Definition:	Users must be able to view past orders with the aim of cancelling orders should they wish
Specification:	- A JTable displaying a history of all orders made by the logged in user. - The table is populated via a SELECT query on the Customer table which INNER JOINS with the Orders table - When an order is selected, the user may cancel the order by clicking the "Cancel Order" button. - Onclick, a DELETE query in the Orders table is generated

Non-Functional

Requirement ID :	**NFR001:** Usability
Definition:	The system must be both learnable and accessible for new users
Specification:	- Learnability: Users must be able to comfortably adapt to the systems GUI, enabling them to purchase products quickly - Accessible: The system must be designed bearing in mind users who may have vision impairments, such as font sizes, colours etc. - Logging errors to a standard text file is a must, ensuring users can learn in more detail what errors may have occurred

Requirement ID :	**NFR002:** Reliability
Definition:	The system must reliably deal with invalid data input from the user
Specification:	- Data input must be handled appropriately according to what may constitute as “bad data” or malicious data. - Preventative measures against SQL Injection by using prepared statements

Requirement ID :	**NFR003:** Performance
Definition:	The system must respond quickly and appropriately to user input
Specification:	- Interaction between the system and database must be seamless, ensuring the customer is met with a responsive application - Any image scaling must be handled appropriately and with care, to ensure the performance drawback is not apparent to the user

Requirement ID :	**NFR004:** Supportability
Definition:	The system must be maintainable for future iterations and expansion
Specification:	- Code must be well documented and conform to standard Object-Oriented principles - System architecture must be well organised and make use of a MVC structure - Extensive version history must be available on a version control platform

Requirement ID :	**NFR005:** Security
Definition:	The system must be secure for the user to use
Specification:	- Any sensitive or precious data shall be handled with care - Passwords must be securely stored in the database by first hashing the input. A standard SHA-256 algorithm may be deployed.