Green Thumb

a web app for Lawn Service companies



Project Design Plan

Trimpin, Sattar, Woolman

CMSC495 Current Trends

University of Maryland, University College

April 14, 2019

Table of Contents

[1. Introduction 3](#_Toc6159106)

[1.1 Purpose 3](#_Toc6159107)

[1.2 Scope 3](#_Toc6159108)

[1.3 Definitions 3](#_Toc6159109)

[2. References 3](#_Toc6159110)

[3. Decomposition Description 4](#_Toc6159111)

[3.1 Module Decomposition 5](#_Toc6159112)

[3.1.1 User Interface 6](#_Toc6159113)

[3.1.2 Services 8](#_Toc6159114)

[3.2 Concurrent Process Decomposition 10](#_Toc6159115)

[3.3 Data Decomposition 11](#_Toc6159116)

[4. Dependency Description 13](#_Toc6159117)

[4.1 Inter-module Dependencies 13](#_Toc6159118)

[4.2 Inter-process Dependencies 15](#_Toc6159119)

[5. Interface Description 16](#_Toc6159120)

[5.1 User Interface Description 16](#_Toc6159121)

[5.2 Services and Product (Landscaping Supplies) site pages 18](#_Toc6159122)

[5.3 Cart site page 19](#_Toc6159123)

[6. Technology 21](#_Toc6159124)

[7. Development Schedule 22](#_Toc6159125)

# 1. Introduction

## 1.1 Purpose

This document will facilitate development of a basic E-commerce website with ability to do sale transactions and scheduling of orders to increase the efficiency of a lawn service business. The purpose of this document is to provide a set of specific requirements for the development of the web-based Green Thumb Lawn Service Software (GTLSS) suite while utilizing agile development practices.

## 1.2 Scope

The web-based GTLSS suite is an online portal facilitating Business-to-Consumer (B2C) transactions and scheduling for a lawn service to increase efficiency and revenue. Thus, the scope of the GTLSS project would be considered midsize due to fact the project will not provide functionality for allocating specific resources such as equipment or employees for specific tasks, accounting, optimizing routing systems for employees, securing and authorizing payment transactions. The GTLSS suite will provide businesses with a customer-facing web application that supports purchasing of lawn supplies/services, scheduling of services, as well as administration of the business-related database, which should simplify the workflow process.

## 1.3 Definitions

For the purpose of the Project Design Plan (PDP), the following definitions will be utilized.

1. Business Logic Processes – Jobs that run on the background of the user interface to carry out logic actions based on inputted values.
2. Services (SV) – An object that performs an action.
3. Domain Objects (DO) – An object, which holds information for specific function.
4. User Interface (UI) – The graphical element that lets a user interact with the system.
5. Concurrent Process – The act of storing, retrieving, deleting, and updating data in the DBMS
6. Entity – Is an active element that communicates with other entities.
7. Database Management System (DBMS) - Represents database system that allows entry, storage, and retrieval of business-related data.

# 2. References

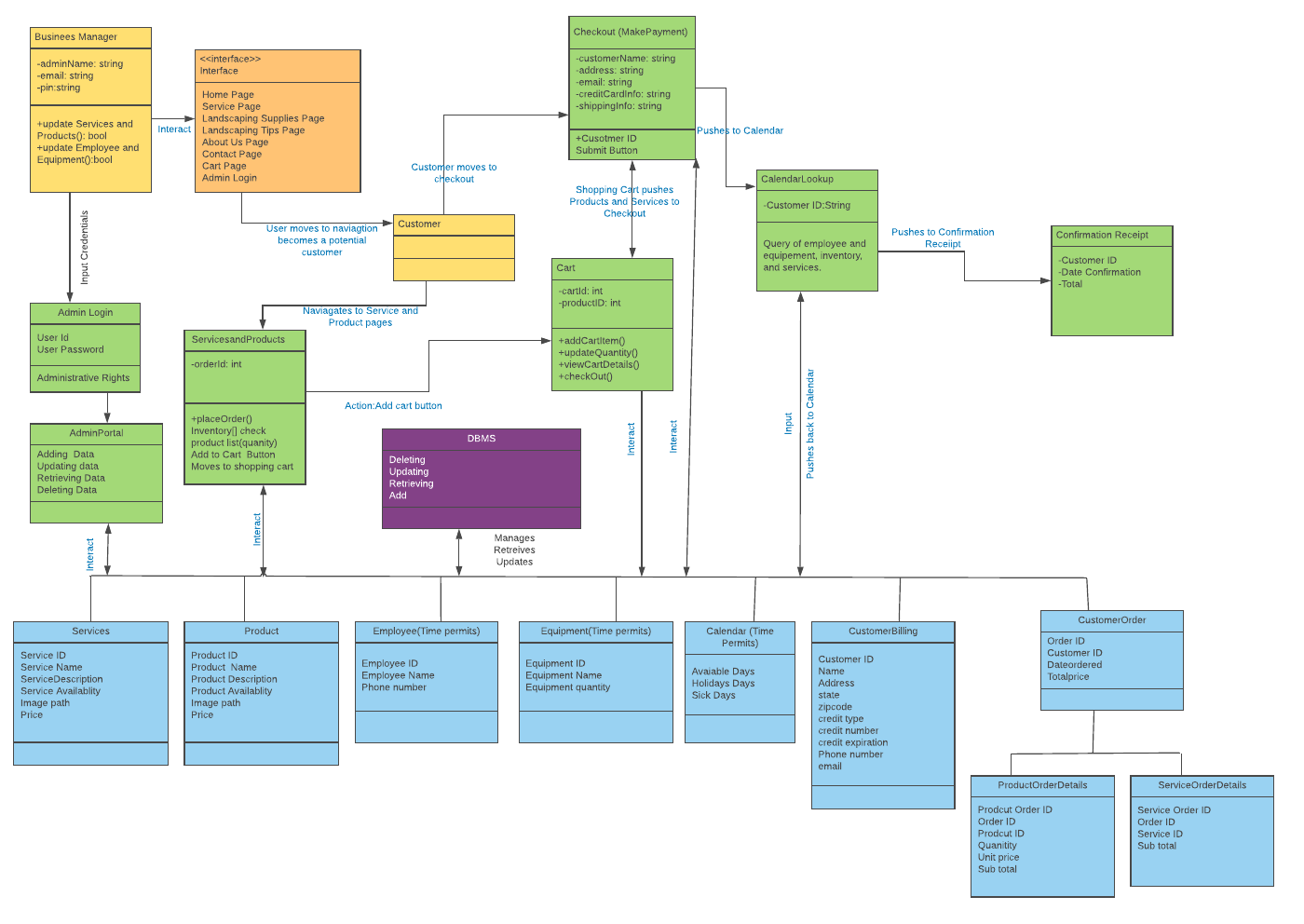
References for this document include the following:

1. IEEE Std. 1016-1998: IEEE Recommend Practice for Software Design Description
2. Project Plan
3. Test Plan

# 3. Decomposition Description

This section will cover the entities and their relationships, which are divided into modules, concurrent processes, and data. Figure 1 gives the entity relationship diagram (ERD) for the GTLSS package. The ERD shows the relationship between the domain objects (blue), the services (green), the user interface (orange), DBMS (purple), and the user (yellow). The ERD diagram shows the orange module as the User Interface allowing access to the customer and Admin (yellow) to initiate and push actions, which in turn utilize domain objects (blue) and queries towards the DBMS (purple) to manage business-process data, produce the scheduled date, and store/confirm the order.

**Figure 1:** **Entity Relationship Diagram (ERD)**

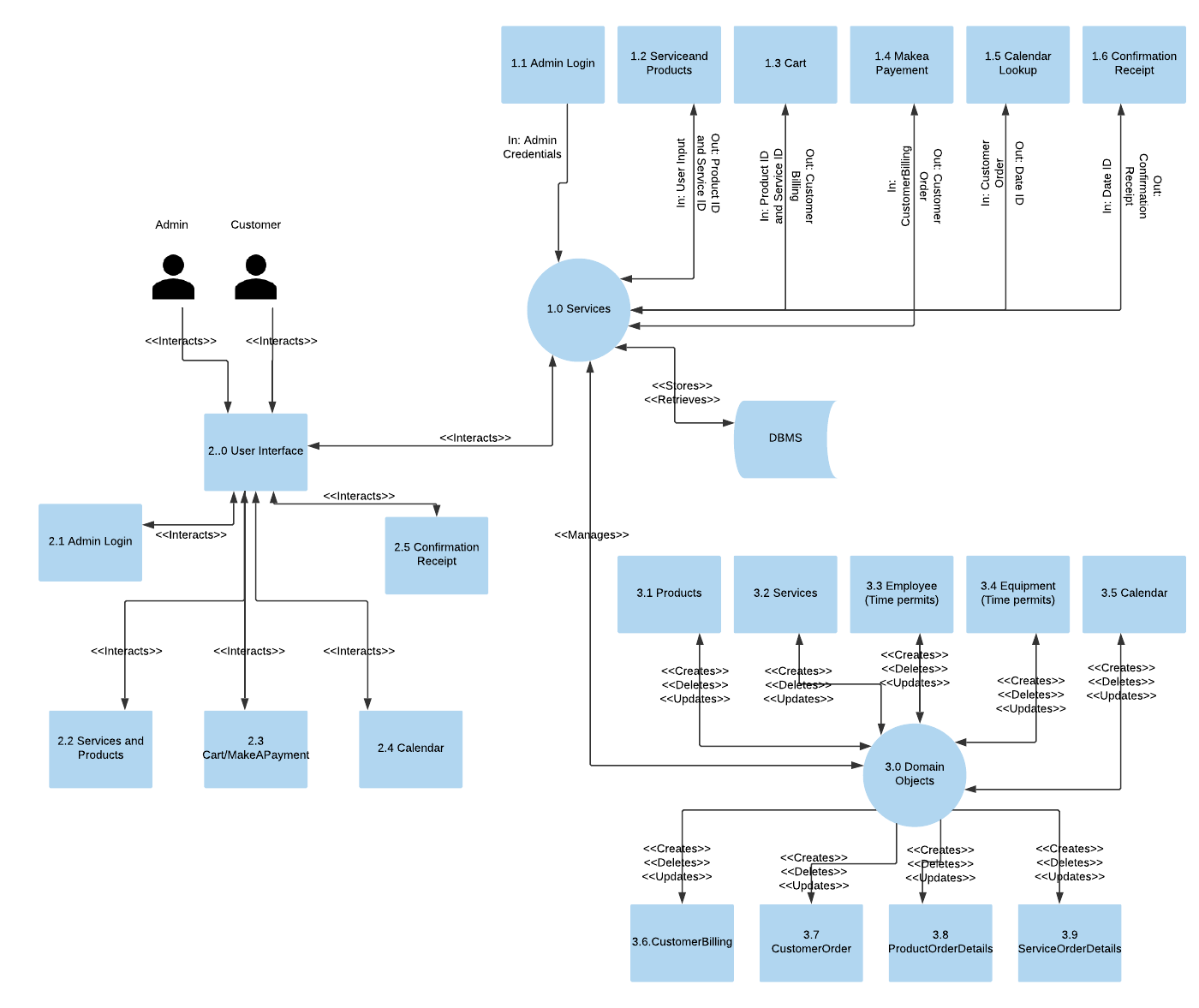


## 3.1 Module Decomposition

The design for the GTLSS suite is broken down into four different components:

1. User Interface
2. Services
3. Domain Object
4. Database

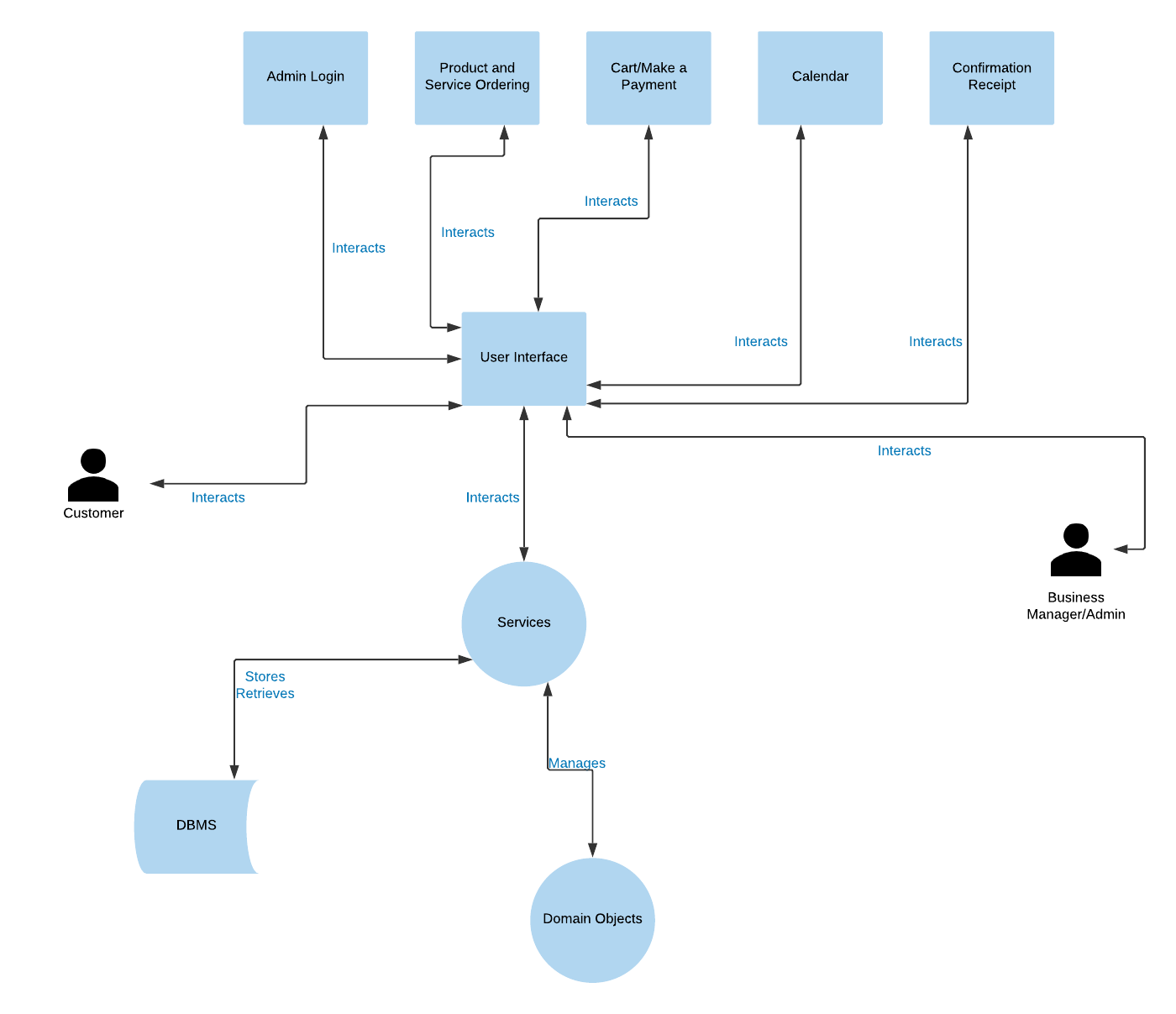
The architectural context diagram (see Figure 2) depicts the human-computer interface for the customer and admin for the GTLSS package through the user interface, and the interaction between the domain objects and the user interface via the service layer. A DBMS persistence is covered via each appropriate service. The diagram includes a DBMS reference to summarize persistence activities. The GTLSS's design uses functional business logic to abstract data in the service component reducing the complexity, allowing effective utilization of the data, and improving the overall business process.

**Figure 2. High Level Architectural Context Diagram** 

### 3.1.1 User Interface

The User Interface of the GTLSS package for users i.e. customers and business manager (Admin) is illustrated in Figure 3. Access to User Interface will require both the customer and business manager to utilize a desktop or laptop with internet connection capabilities.

**Figure 3. Architectural Context Diagram – User Interface**



**3.1.1.1 Admin Login**

The Admin Login site page will contain secure access to the Admin Portal by requesting a user name and password to update the DBMS containing the employee information, equipment information, calendar dates, customer information, and inventory. The mockup of the page will be provided in section 5.1.

**3.1.1.2** **Product and Services Order**

The Service and Landscaping Supplies site pages will contain product and service name, description, image path, price, and an add to cart button. The mockup of the pages will be provided in section 5.2.

**3.1.1.3** **Cart**

The Cart site page will contain order items i.e. services and products with price and total order price. The cart page will require input of customer and payment information. The mockup of the page will be provided in section 5.3.

**3.1.1.4 Calendar**

The calendar will appear on the cart site page for the customer to choose a date of the services. The mockup of the calendar will be provided in section 5.3.

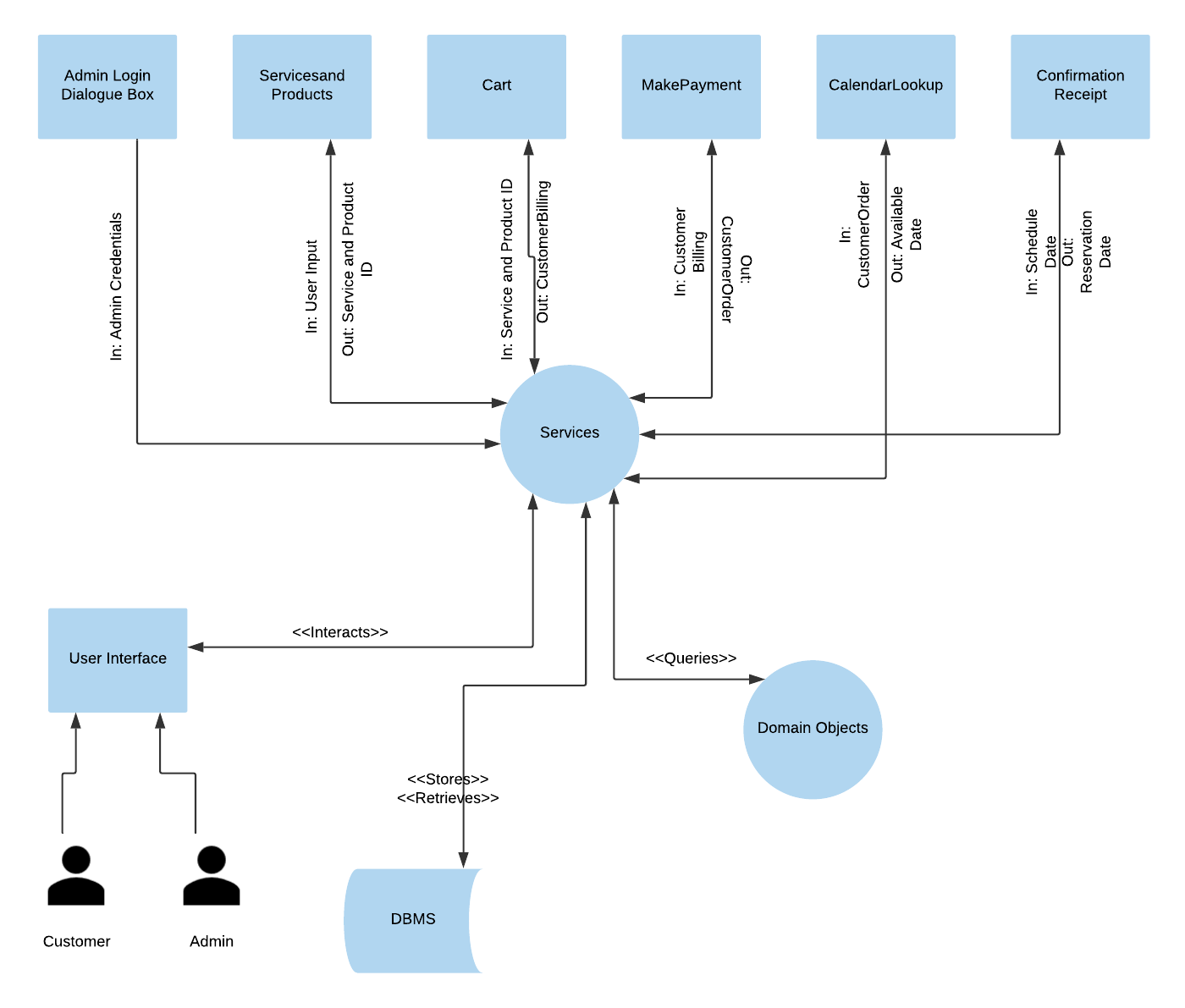
**3.1.1.5 Confirmation Receipt**

The confirmation receipt will appear on the cart site page as a pop up window with payment and schedule date confirmation. The mockup of the pop-up window will be provided in section 5.3.

### 3.1.2 Services

The service module will provide the business logic processes for the GLTSS such as inputs, outputs, necessary actions, and queries with the DBMS. The services for GLTSS are illustrated in Figure 4. The inputs by the user i.e. customer or Admin will induce an action for the GTLSS package to perform such as queries on the DBMS and output of pop up window for confirmation receipt. Examples of services would be functionality of add new product button, calculating of order total price, functionality of submit button, timely prompt and authorization of admin credentials, timely prompt for customer billing information (name, address, phone number, and email). The output of the actions by the customer will require services to perform action with DBMS in the form of queries to create inputs back to the customer for scheduling of date and confirmation receipt in the form of a pop-up window.

**Figure 4: Architectural Context Diagram – Services**



**3.1.2.1 AdminLogin Entity**

The service AdminLogin Entity assigns user permission to access the DBMS for updates on employees, equipment, inventory, and calendar.

**3.1.2.2 ServicesandProducts Entity**

The service ServicesandProducts Entity component handles adding items from the Service and LandScaping Supplies site page to the cart site page through the add to cart button and quering of the DBMS.

**3.1.2.3 Cart Entity**

The service Cart Entity handles the business logic for presenting all selected items for purchase, collecting customer information, and processing submit button functionality.

**3.1.2.4 MakePayment Entity**

The service Payment Entity handles the business logic confirming customer information is correct, issuing a customer ID, and calculating for the order total.

**3.1.2.5 CalendarLookup Entity**

The service CalendarLookup Entity handles the business logic to search for the available service date and collect the customer date selection for the customer ID.

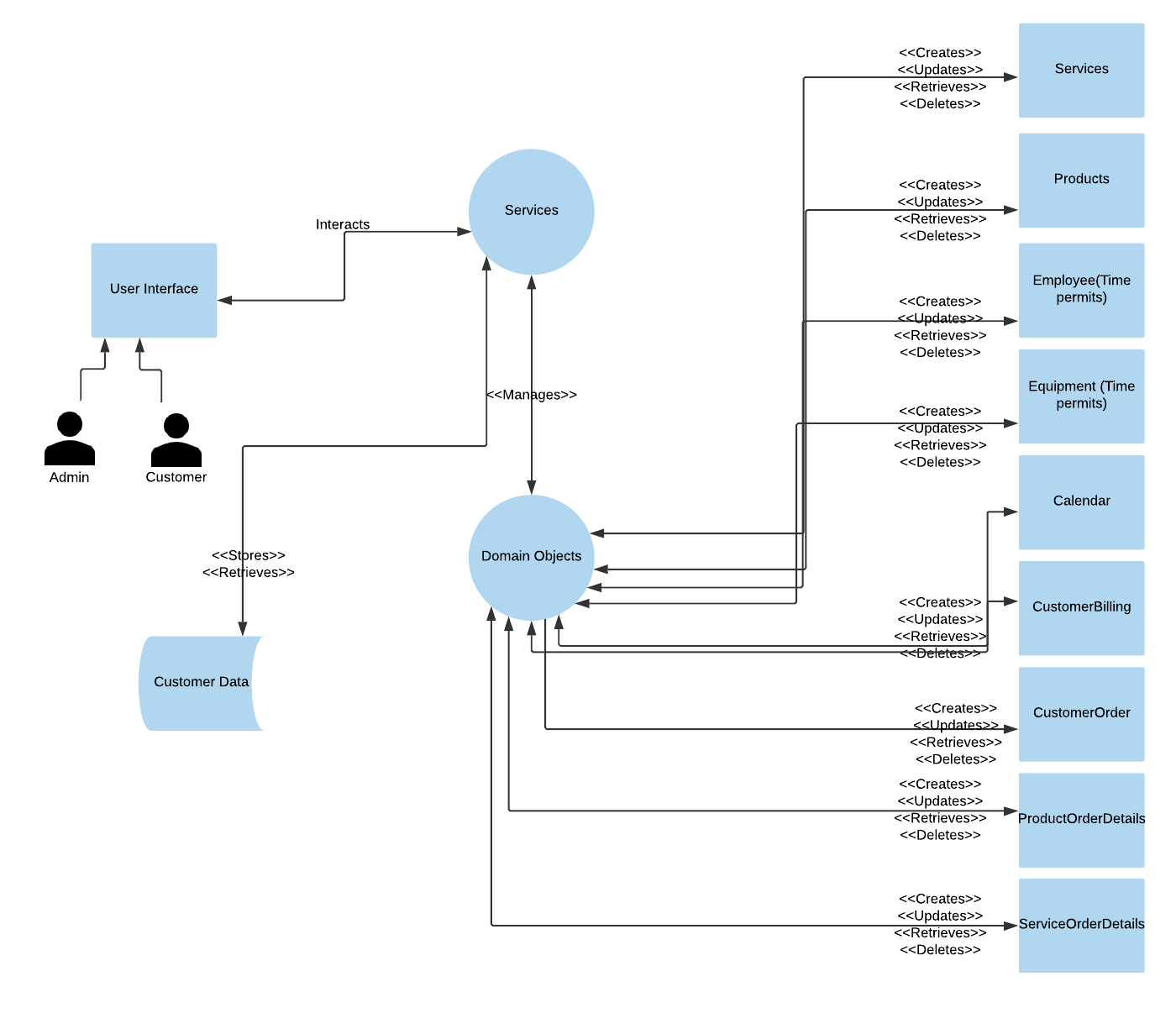
**3.1.2.6 ConfirmationReceipt Entity**

The ConfirmationReceipt Entity assigns the date to the customer ID and returns the confirmation of scheduled date and submitted order.

## 3.2 Concurrent Process Decomposition

The concurrent process decomposition is a module used to describe any service that acts independent of the user (see Figure 5). The process would create, manage, update, and retrieve data for any domain object within the GTLSS package.

**Figure 5: Architectural Context Diagram – Domain Objects**



**3.2.1 CalendarLookup Entity**

The service CalendarLookup Entity queries through all calendar objects searching for available dates to schedule date of service.

## 3.3 Data Decomposition

The data decomposition module describes the entities that hold data about particular function in the GTLSS package. For the purpose of this document, these entities are referred to as domain objects.

**3.3.1 Employee Entity**

The Employee Entity holds all the information related to the employees of the GTLSS business. The information in the domain object of employee includes metadata such as employee ID, first and last name, and phone number.

**3.3.2 Equipment Entity**

The Equipment Entity holds all the information related to the equipment of the GTLSS business. The information in the domain object of equipment includes metadata such as equipment ID, name, and description.

**3.3.3 Product Entity**

The Product Entity holds all the information related to the product inventory for the service ServiceandProduct Entity. The information in the domain object of product includes metadata such as product ID, image path, description, and price.

**3.3.4 Service Entity**

The Service Entity holds all the information related to the service inventory for the service ServiceandProduct Entity. The information in the domain object of service includes metadata such as service ID, image path, description, and price.

**3.3.5 Calendar Entity**

The Calendar Entity holds the information necessary for the service CalendarLookup Entity to produce a confirmation of scheduled date and order submission. The information in the domain object of calendar includes metadata of the available dates and holidays.

**3.3.6 Customer Billing Entity**

The Customer Billing Entity holds all the information related to the service Make Payment Entity, which ensures the addition of customer billing information to the DBMS. The information in the domain object of customer billing includes metadata such as customer name (first and last), address, phone number, email address, credit cart type, number, and expiration date.

**3.3.7 Customer Order Entity**

The Customer Order Entity holds general information related to the service ServicesandProducts, Cart, and Make Payment Entities, which ensures the addition of services and products and customer billing information to the DBMS. The information in the domain object of customer order includes metadata such as customer ID, order ID, date ordered, total price.

**3.3.8 Product Order Details Entity**

The Product Order Details Entity holds all detailed information related to the ServicesandProducts and Cart Entities, which ensures the addition of details for order of products to the DBMS. The information in the domain object of product order details includes metadata such as product order ID, order ID, product ID, quantity, unit price, subtotal.

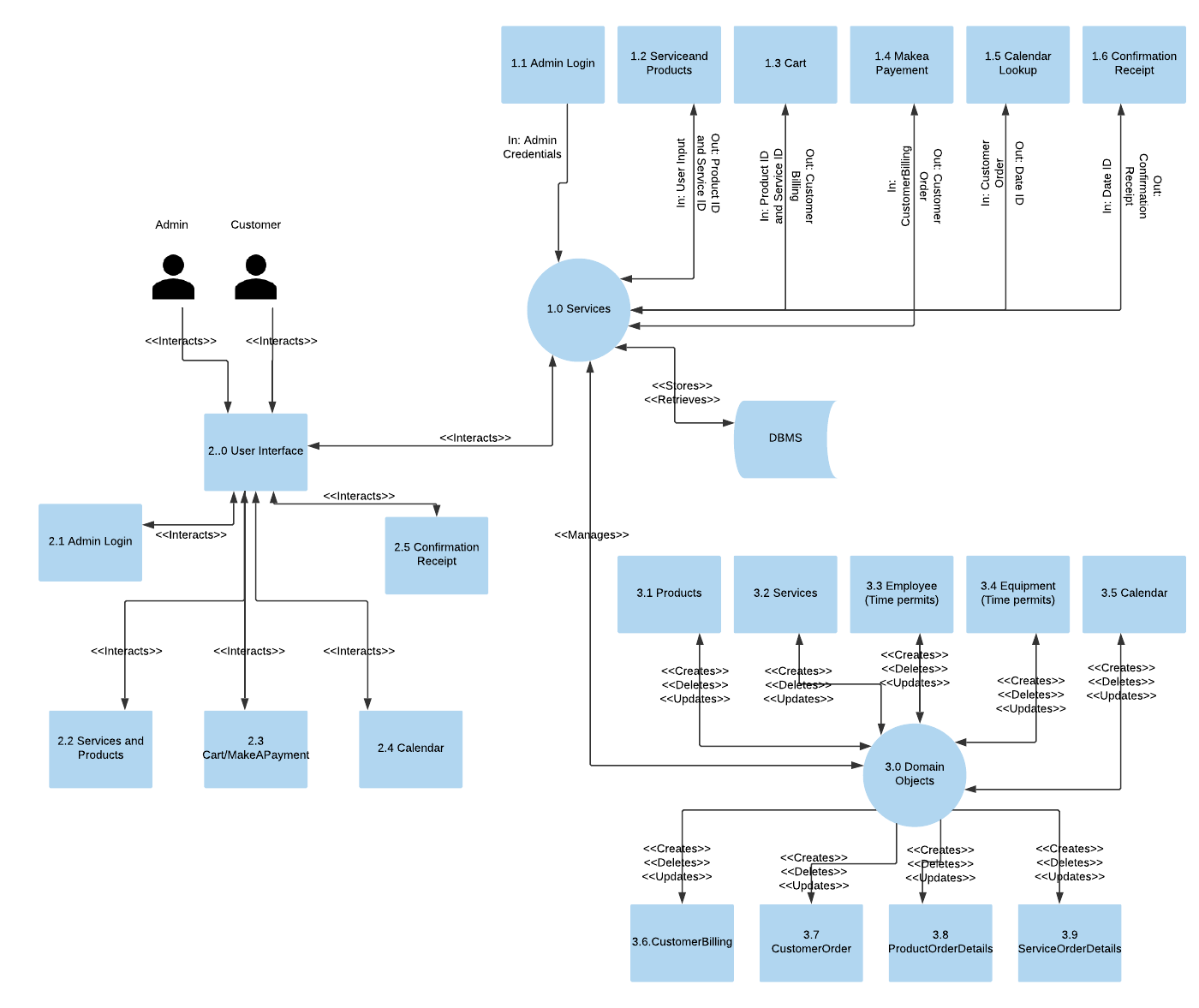
**3.3.9 Service Order Details Entity**

The Service Order Details Entity holds all detailed information related to the ServicesandProducts and Cart Entities, which ensures the addition of details for order of services to the DBMS. The information in the domain object of service order details includes metadata such as service order ID, order ID, service ID, subtotal.

# 4. Dependency Description

This section describes the dependency of service module, user interface, and the domain objects for GTLSS (see Figure 6). The service module adds, retrieves, updates, and deletes the domain object directly via the database module entries. The user interface module interacts with the service module to allow the customer and Admin to perform actions. The service module controls and manages the domain objects.

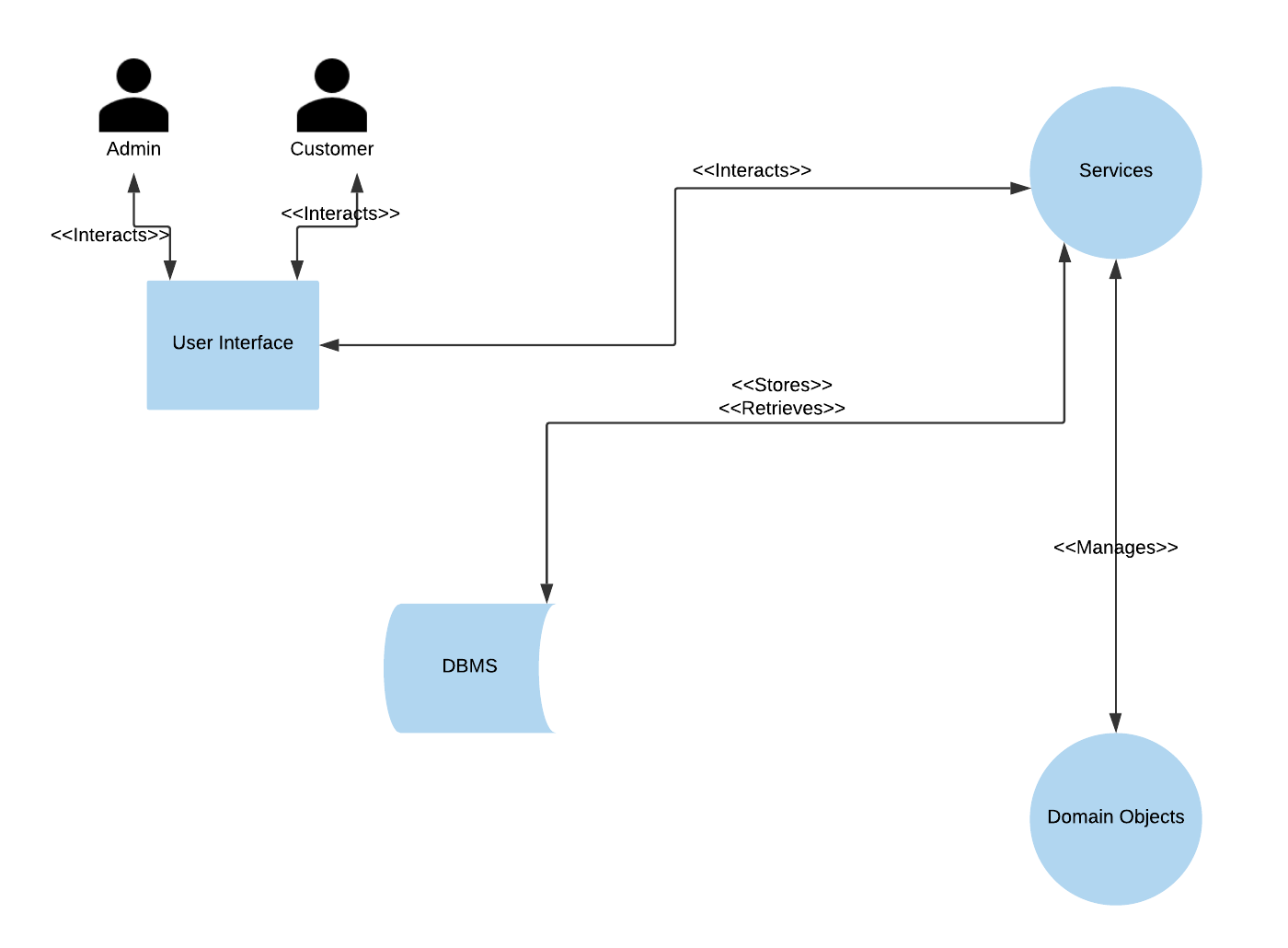
**Figure 6: Data Flow Diagram**



## 4.1 Inter-module Dependencies

For the GTLSS package, there are four core modules as demonstrated in Figure 7: user interface, services, domain objects, and database (BDMS). The services module is dependent on the user interface interaction with the customer and the Admin for the business. The database module will only be called by the services module to add, update, create, and delete data (since DBMS is not a core module, no description will be provided for it).

**Figure 7: Inter-module Dependency Diagram**



1. The User Interface module is tasked with displaying information for the GTLSS site, which results in a dependency with the Services module. The dependency relationship between the User Interface module and Services module allows the business logic from the Services module to be represented in the views directly at the User Interface. As a result, the customer or Admin has a dependency on the User Interface to present and interact with those views.

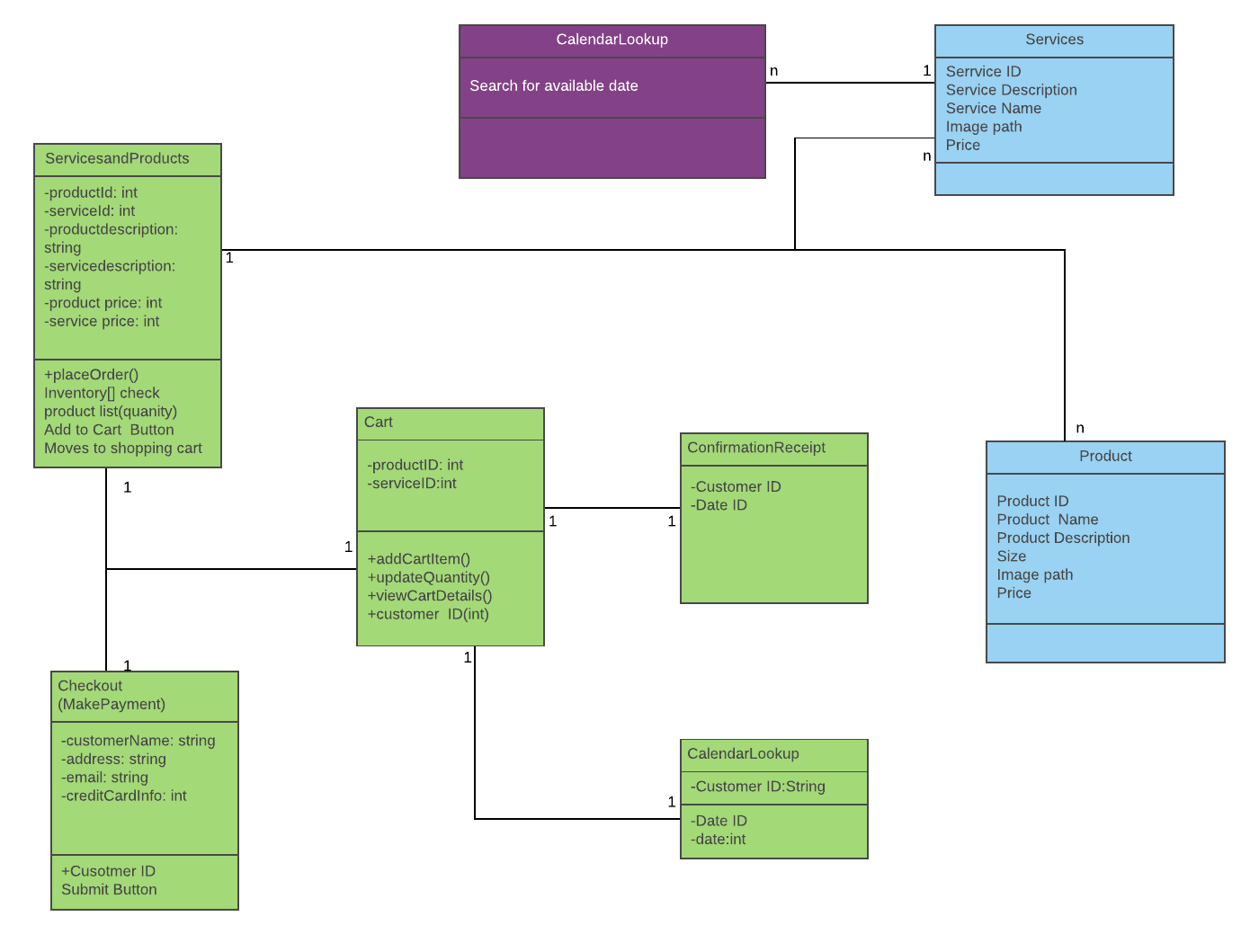
2. The Service module has dependencies on both the User Interface and domain object modules. The service module presents the views to the User Interface, whereas the domain object module adds, retrieves, updates, and deletes data within business-related database. As the domain objects hold business data, but business logic is in the service module, this dependency with domain object module is critical for states of action and pushing information within the system.

3. The Domain Object module has a dependency on the services module to process data with business logic.

## 4.2 Inter-process Dependencies

Figure 8 displays the five inter-process dependencies within the GTLSS package. The use of the domain objects within the inter-process dependency diagram describes the essential relationship between the Cart service and Service/Product domain objects and relationship with the CalendarLookup concurrent process.

**Figure 8: Inter-process Dependency Diagram**



1. Cart has a dependency on several other services to include:

1. ServicesandProducts – Defines the products and services available for the customer. Cart will contain the Product ID and Service ID of items ordered from the ServiceandProducts so that they can be confirmed by the customer.
2. Checkout (MakeaPayment) – This dependency is used to drive information for the customer input to issue a Customer ID. The relationship is defined as processing of the customer payment, which is linked to the ServicesandProducts.
3. CalendarLookup – Is linked to the Cart that pushes Customer ID so that the customer can view a calendar to schedule a date for service. The CalendarLookup drives to the Cart the date selected by the customer.
4. ConfirmationReceipt - Is linked to the Cart that pushes Customer ID and to the CalendarLookup. This dependency is used to drive confirmation information (i.e. confirmation of service date, total order price, and successful submission of the order) to the cart to be displayed to the customer.

2. The CalendarLookup process is loosely related to the CalendarLookup service. Since this process is defined as a concurrent service, it queries through the calendar searching for available dates to schedule date of service.

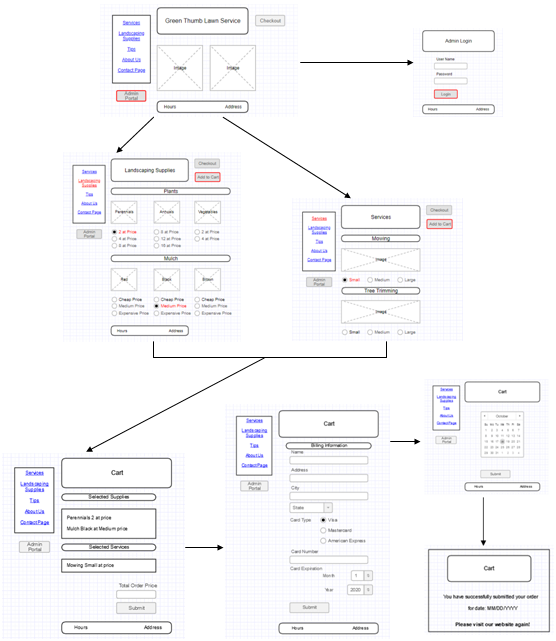
The data dependencies in GTLSS are linked through the process dependencies to services by abstracting the business logic away from the User Interface module and the Domain Object module.

# 5. Interface Description

## 5.1 User Interface Description

The User Interface allows customers and business manager/Admin to interact with GTLSS suite. Consequently, the admin can manage inventory, equipment, and employees, and customers can place/create orders. Figure 9 below provides the perspective of the human-computer interaction with the GTLSS package.

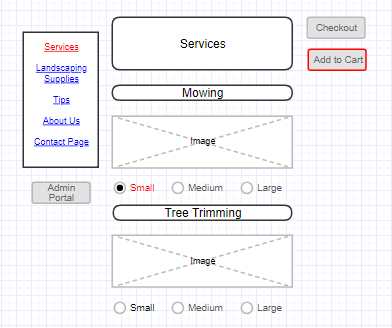
**Figure 9: User Interface workflow for GTLSS package**



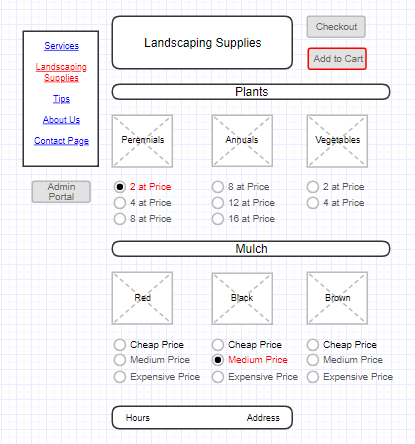
## 5.2 Services and Product (Landscaping Supplies) site pages

The Services, Product, and Cart site pages are the primary interface for the GTLSS package. The customer will order services and products from the business and move to the Cart site page. An Add to Cart button will be available to the user to added selected service/products to the Cart (see Figures 10 and 11).

**Figure 10: Service site page**



**Figure 10: Products (Landscaping supplies) site page**

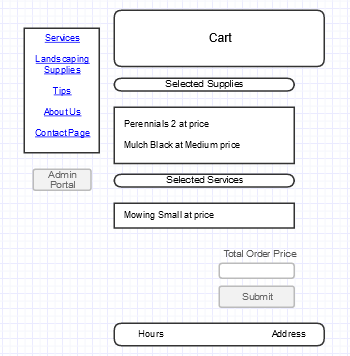


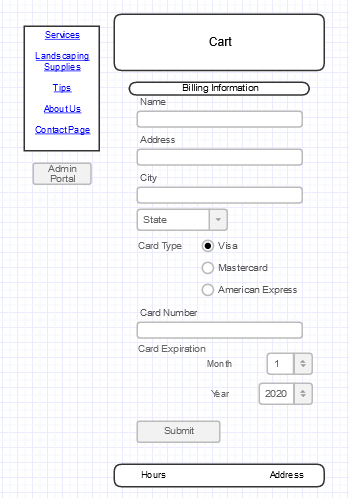
## 5.3 Cart site page

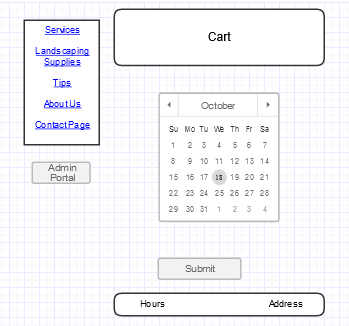
The Cart site page will support the primary user interactions after ordering. The Cart page will provide the following capabilities (see Figure 14):

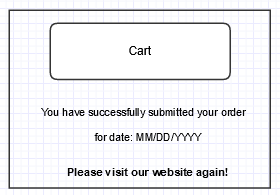
1. A view with the description of the order items, price, and a submit button.
2. As the submit button is pressed, the text boxes for customer billing information (i.e. name (first and last), address, phone #, email address, and credit card number) will appear along with a submit button.
3. After the customer input is entered and the submit button is pressed, a calendar will pop up (not a separate screen) to allow the customer to choose a date for service.
4. After the date is selected by the customer and the submit button is pressed, a notification will appear with either confirmation of scheduled date and placed order or request to select a different date.

**Figure 14: Cart site page**

****

****

****

****

# 6. Technology

* HTML/CSS, and JavaScript Technologies
* Servlet and Java Server Pages (JSP) Technologies
* Enterprise Java Beans (EJB) Technologies
* Java Persistence API (JPA)
* Java Server Pages Standard Library (JSTL)
* Java Database Connectivity (JDBC)
* MySQL Server for Database
* Apache Server

# 7. Development Schedule

The development schedule for the production of the web-based GTLSS suite is based on a collaborative work of all team members and utilization of agile development practices. The GTLSS developers will be using HTML, CSS, JavaScript, SQL, and other languages to complete the functionality. Figure 15 provides a detailed plan for development with team members assigned to different modules of the GTLSS package.

**Figure 15: Development Schedule for GTLSS**

