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](#_Toc7299151)

[1.1 Purpose 3](#_Toc7299152)

[1.2 Scope 3](#_Toc7299153)

[1.3 Definitions 3](#_Toc7299154)

[2. References 3](#_Toc7299155)

[3. Decomposition Description 4](#_Toc7299156)

[3.1 Module Decomposition 5](#_Toc7299157)

[3.1.1 User Interface 7](#_Toc7299158)

[3.1.2 Services 9](#_Toc7299159)

[3.2 Concurrent Process Decomposition 12](#_Toc7299160)

[3.3 Data Decomposition 13](#_Toc7299161)

[4. Dependency Description 15](#_Toc7299162)

[4.1 Inter-module Dependencies 16](#_Toc7299163)

[4.2 Inter-process Dependencies 18](#_Toc7299164)

[5. Interface Description 19](#_Toc7299165)

[5.1 User Interface Description 19](#_Toc7299166)

[5.2 Product site pages 21](#_Toc7299167)

[5.3 Cart site page 22](#_Toc7299168)

[6. Technology 24](#_Toc7299169)

[7. Development Schedule 25](#_Toc7299170)

[8. Project Change Log 27](#_Toc7299171)

# 

# 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, 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

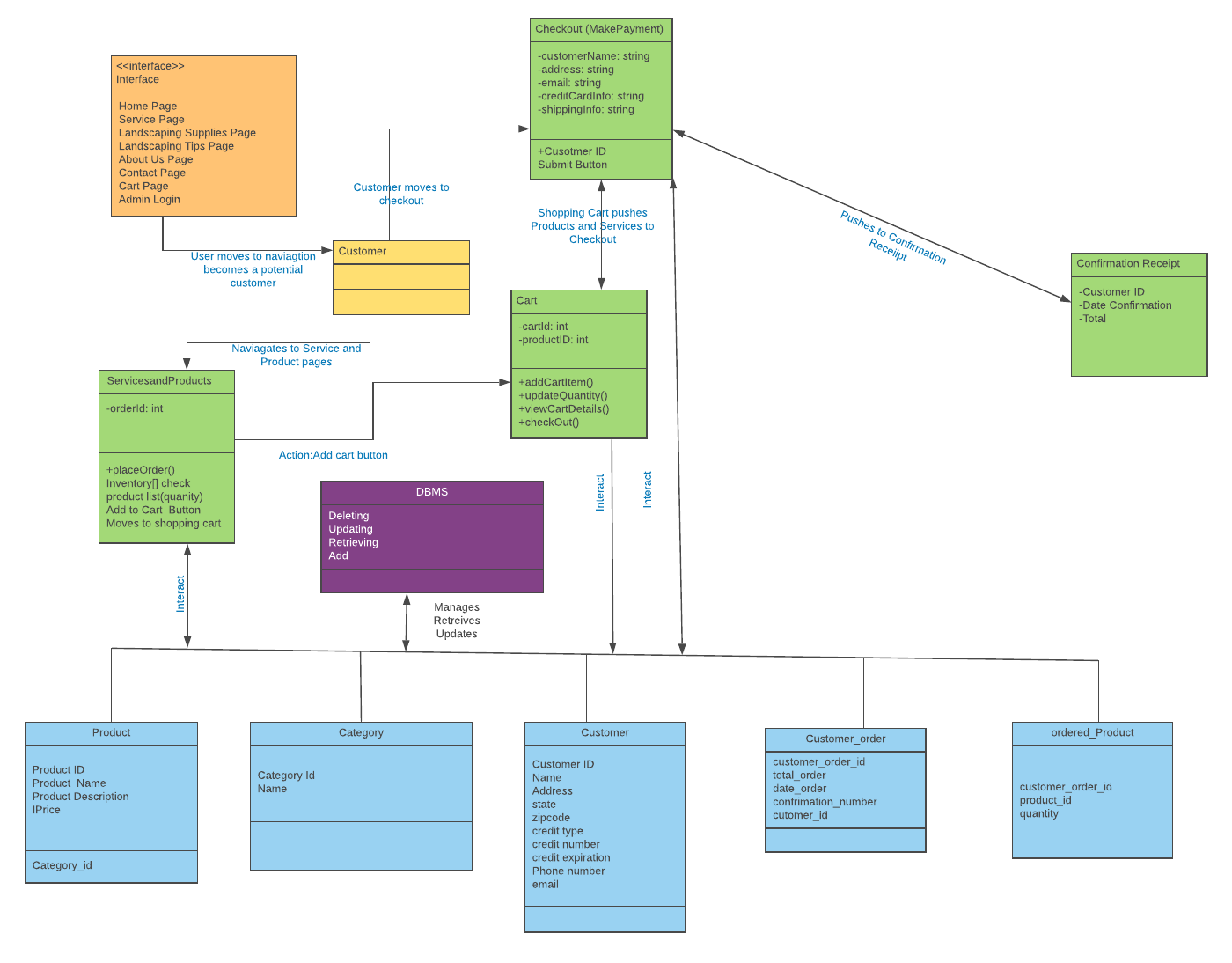
References for this document include the following:

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, and store/confirm the order.

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



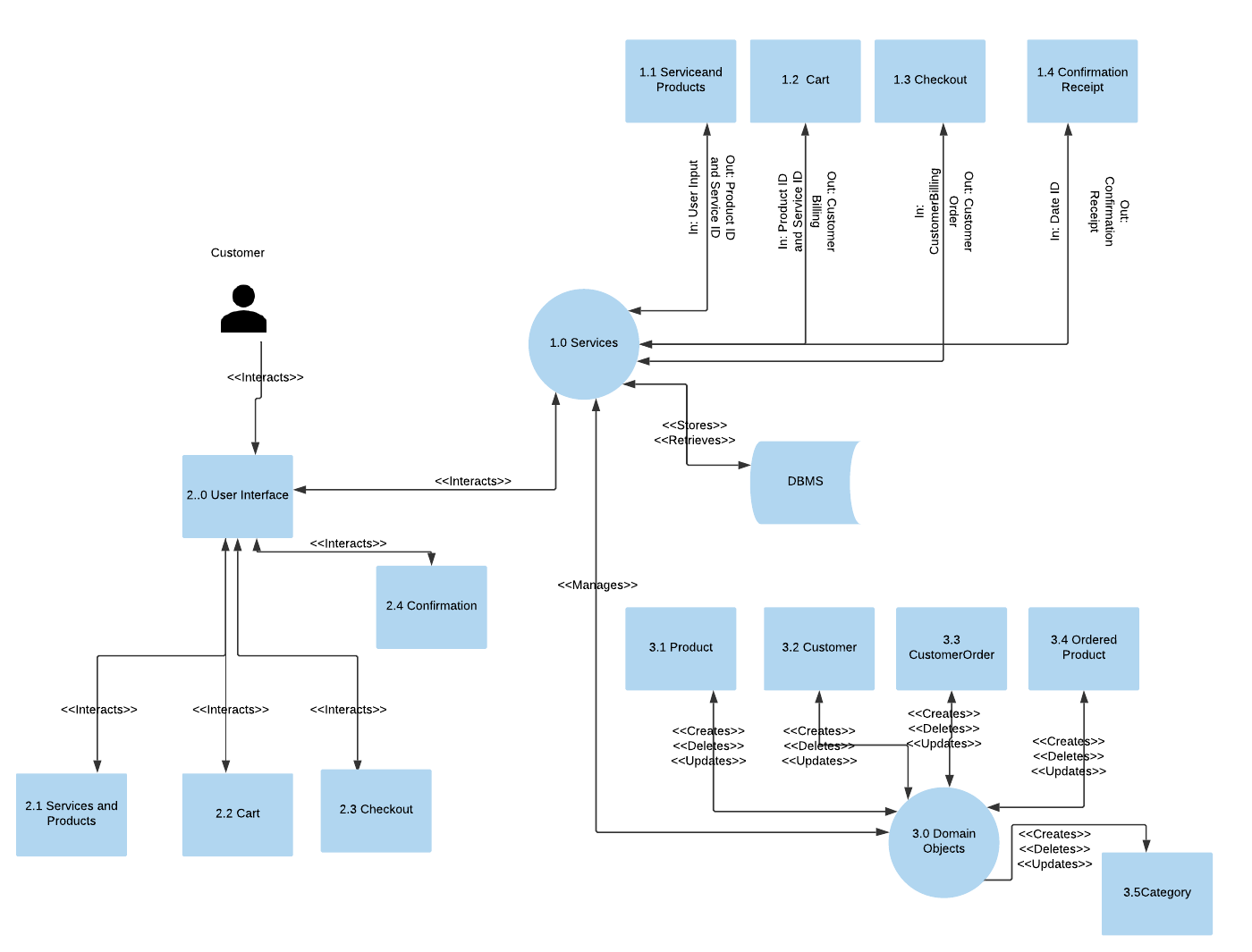
## 3.1 Module Decomposition

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.

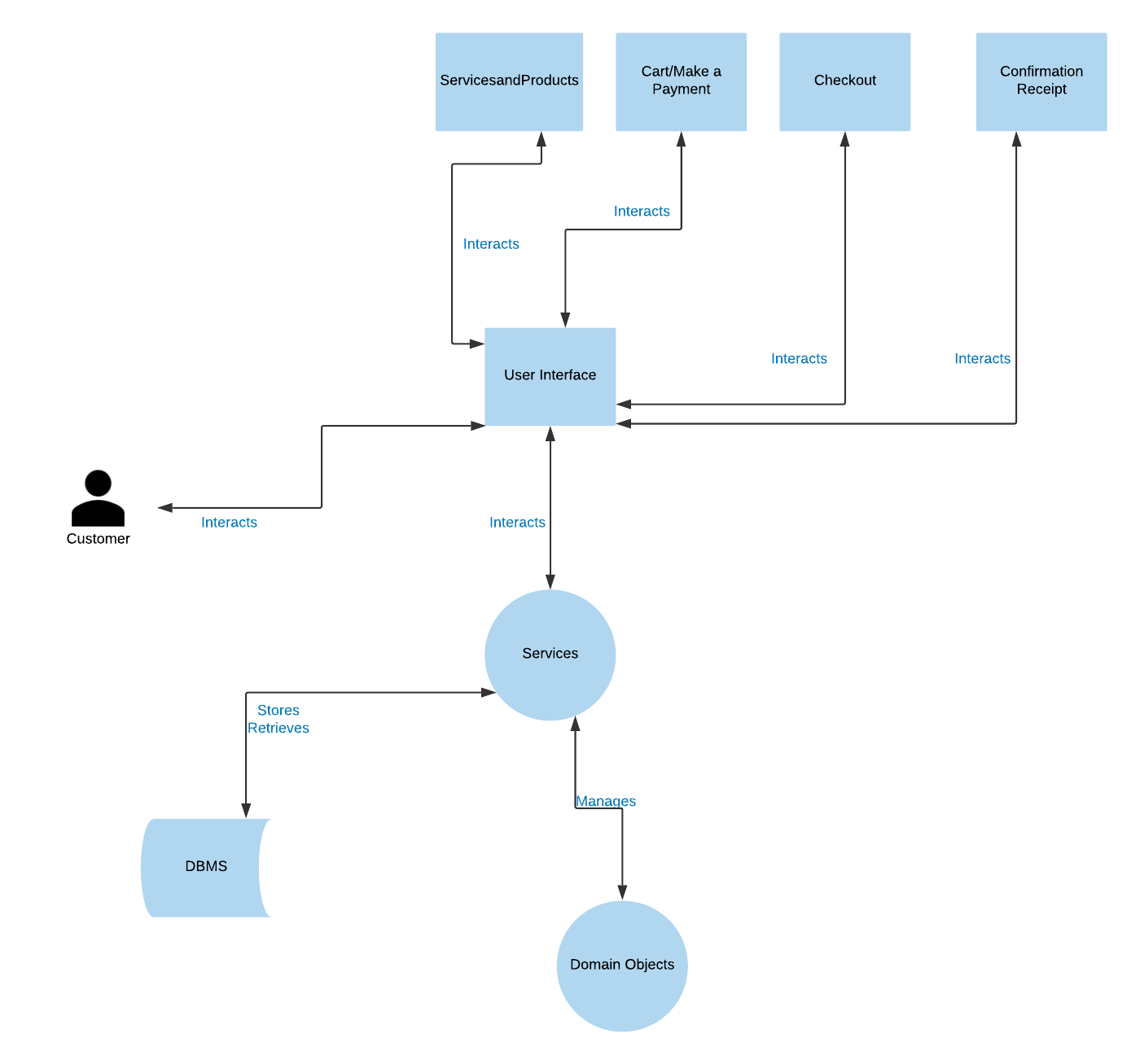
**Figure 2. High Level Architectural Context Diagram**



### 3.1.1 User Interface

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

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



**3.1.1.1** **Product and Services Order**

The Product and Service site page will contain product and service name, description, 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.

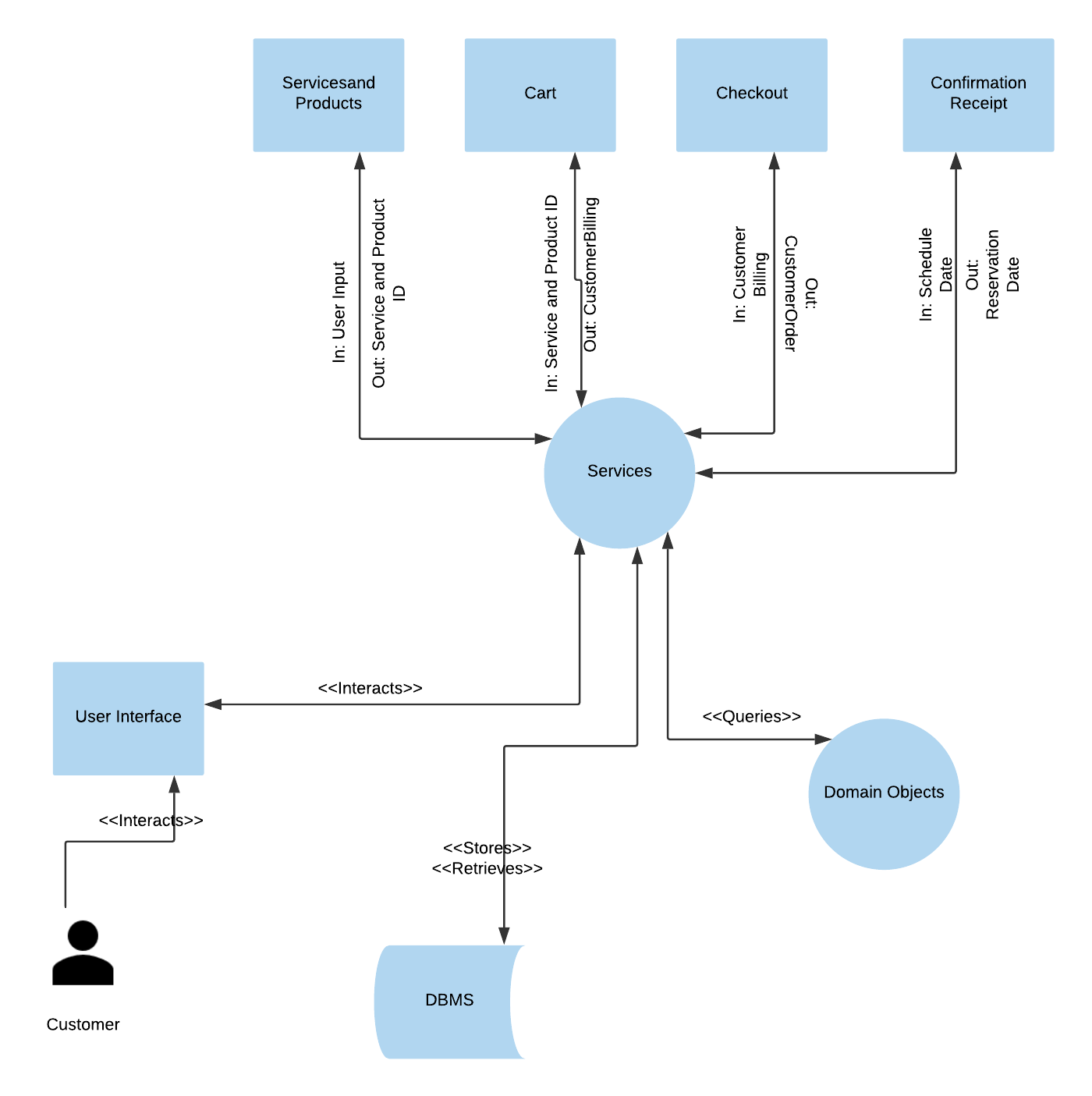
**3.1.1.5 Confirmation Receipt**

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 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 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 confirmation receipt in the form of a pop-up window.

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



**3.1.2.2 Services and Products Ordering Entity**

The service Services and Products Ordering Entity component handles adding items from the Products and Services 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.

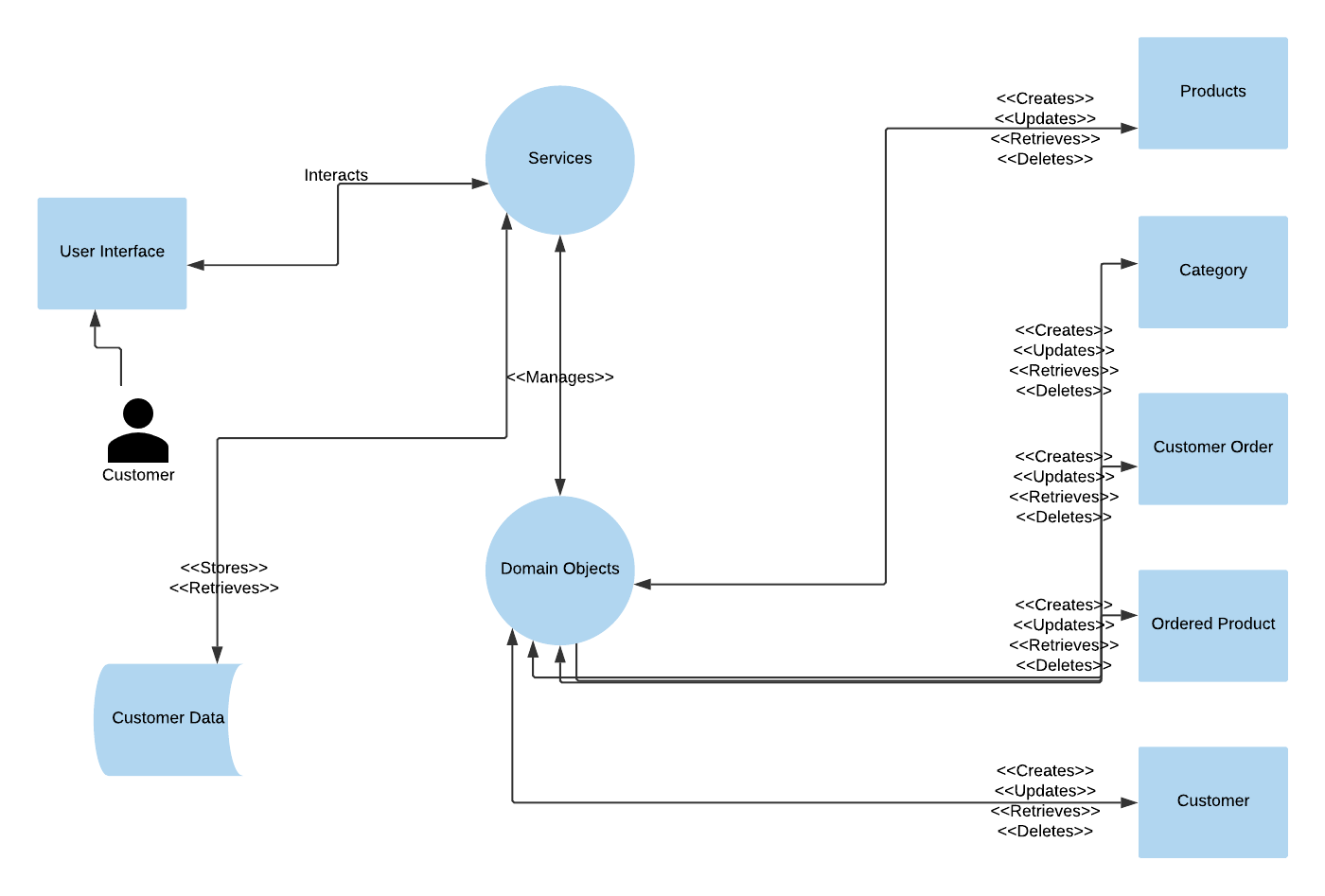
**3.1.2.6 ConfirmationReceipt Entity**

The ConfirmationReceipt Entity 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, update, retrieve, and delete data for any domain object within the GTLSS package.

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



## 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 Category Entity**

The Category Entity holds the information related to the category of products for sale by the GTLSS business. The information in the domain object of category includes metadata such as category ID, category name.

**3.3.3 Product Entity**

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

**3.3.6 Customer Entity**

The Customer 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 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 Services and Products Ordering, 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, confirmation number.

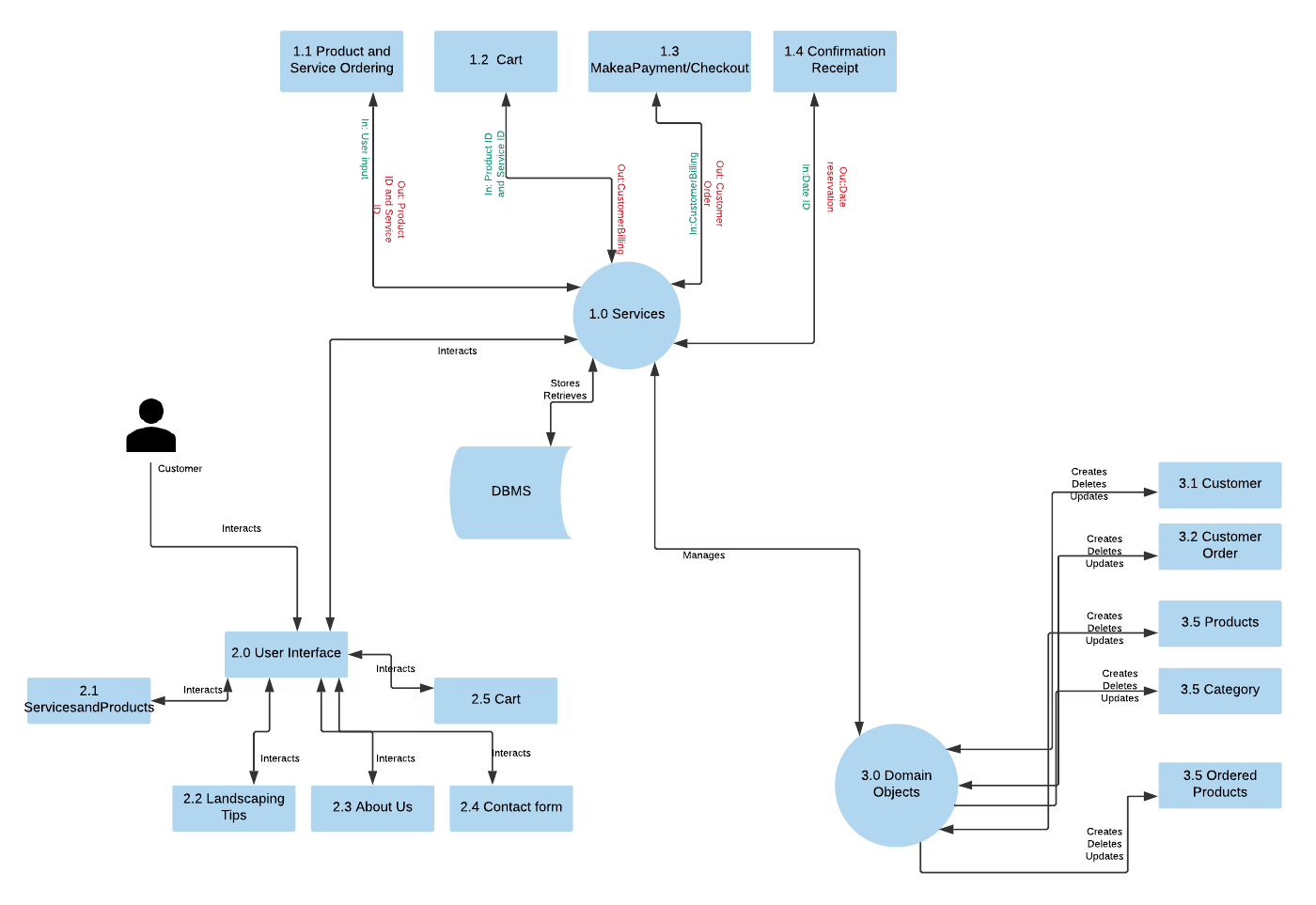
**3.3.8 Ordered Product Entity**

The Ordered Product Entity holds all detailed information related to the Services and Products Ordering and Cart Entities, which ensures the addition of details for order of products to the DBMS. The information in the domain object of ordered product includes metadata such as customer order ID, product ID, quantity

# 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 to perform actions. The service module controls and manages the domain objects.

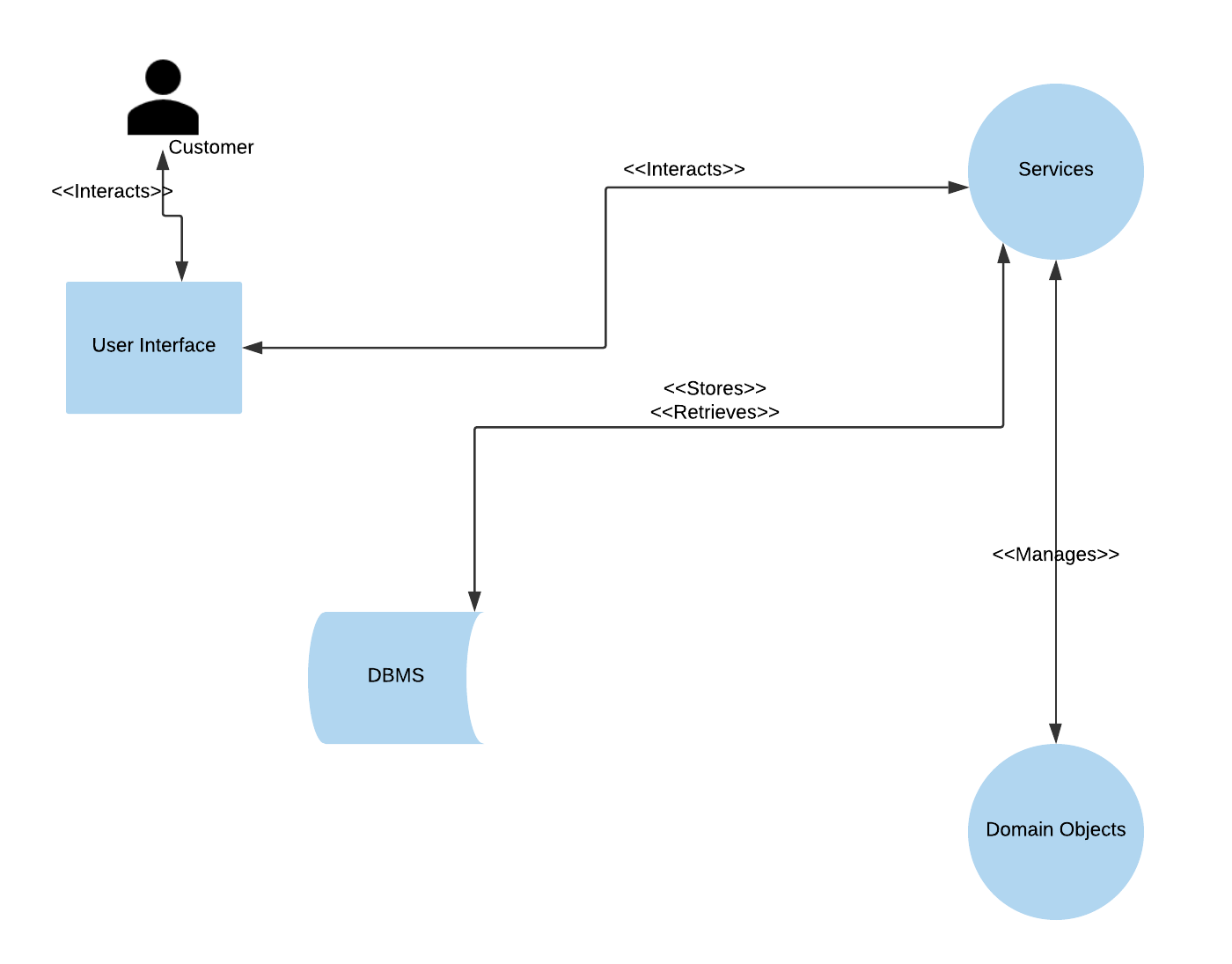
**Figure 6: Data Flow Diagram**



## 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 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 has a dependency on the User Interface to present and interact with those views.

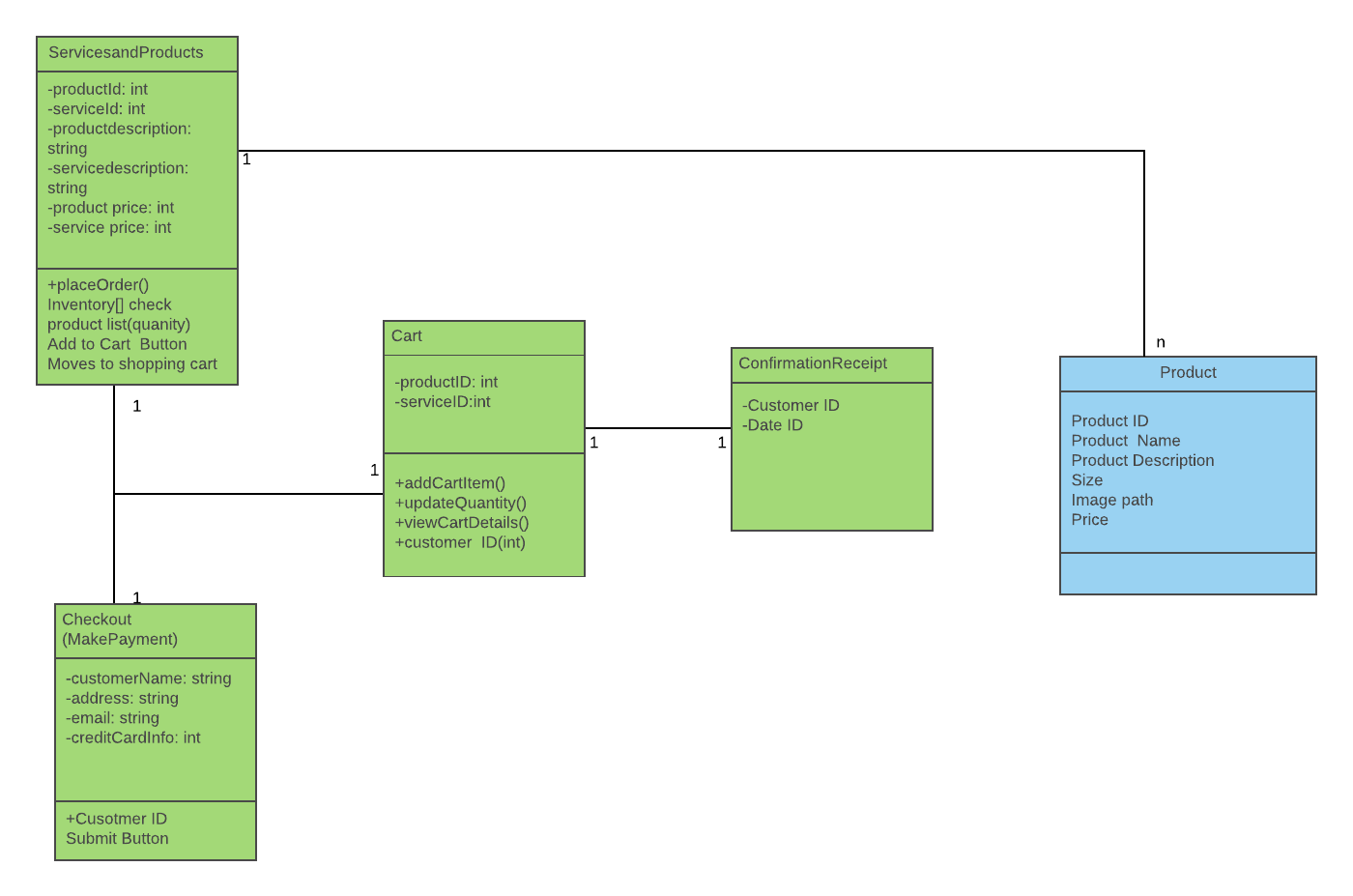
2. The Service module has dependencies on both the User Interface and domain object modules. The service module presents the views to the User Interface, whereas the domain object module adds, retrieves, updates, and deletes data within business-related database. As the domain objects hold business data, 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 Product domain object.

**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 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. ConfirmationReceipt - Is linked to the Cart that pushes Customer ID. 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.

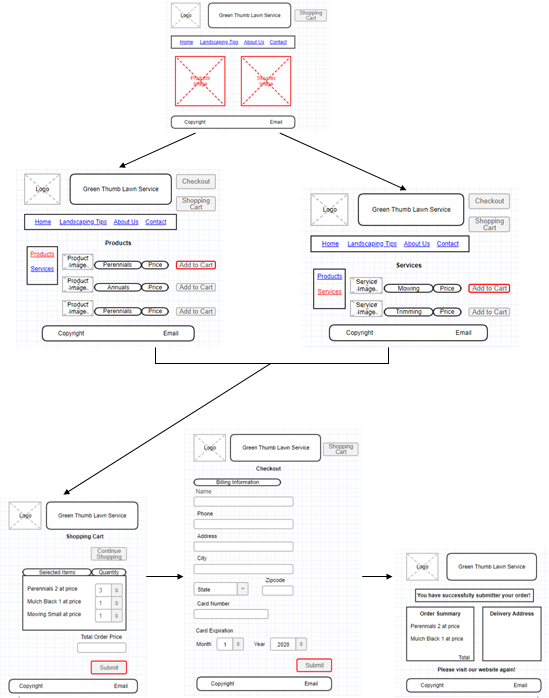
The data dependencies in GTLSS are linked through the process dependencies to services by abstracting the business logic away from the User Interface module and the Domain Object module.

# 5. Interface Description

## 5.1 User Interface Description

The User Interface allows customers to interact with GTLSS suite. Consequently, customers can place/create orders. Figure 9 below provides the perspective of the human-computer interaction with the GTLSS package.

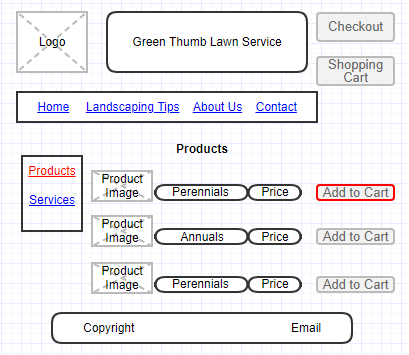
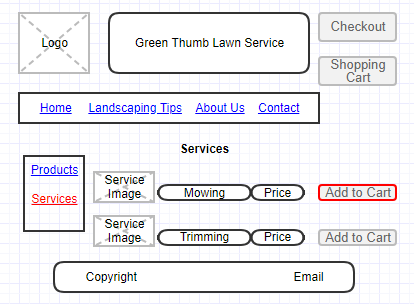
**Figure 9: User Interface workflow for GTLSS package**

****

## 5.2 Product and Services site page

The Product (services/products) 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 Figure 10).

**Figure 10: Products and Services site page**

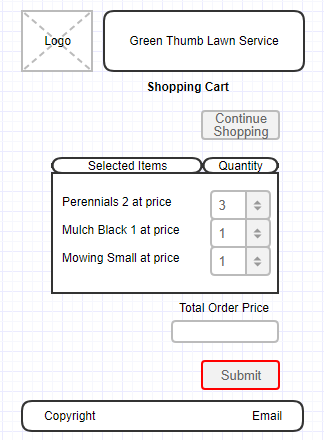
 

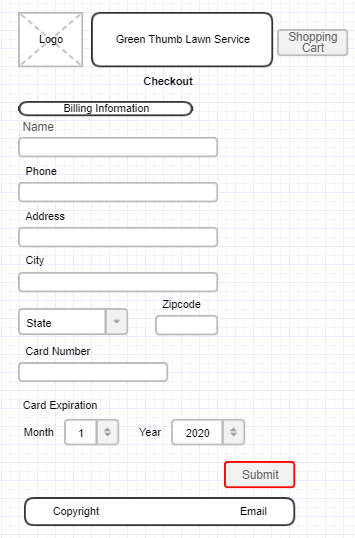
## 5.3 Cart, Checkout, and Confirmation site page

The Cart, Checkout, and Confirmation site pages will support the primary user interactions after ordering. These pages will provide the following capabilities (see Figure 11):

1. The Cart page will allow to review the ordered items, price, and will have option to change quantity of items and a submit button.
2. As the submit button is pressed, the Checkout page will appear providing the text boxes for customer billing information (i.e. name (first and last), address, phone #, email address, and credit card number) along with a submit button.
3. After the customer input is entered and the submit button is pressed, the Confirmation page will appear with confirmation of placed order.

**Figure 11: Cart, Checkout, and Confirmation site page****s**

****

****

****

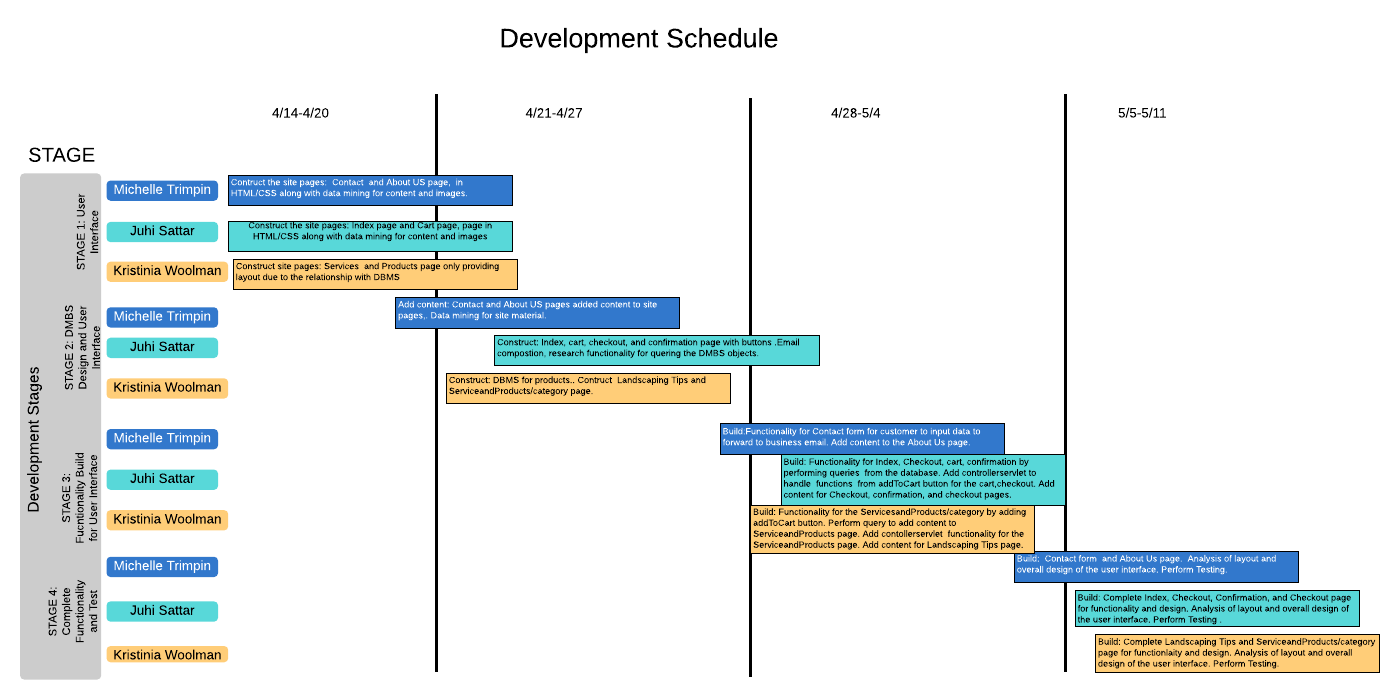
# 6. Technology

* 1. Integrated Development Environment - NetBeans IDE 8.2 All bundle
* 2. Language – HTML/CSS/JavaScript/SQL/Java
* 3. JDK 8.0
* 4. Servlet and Java Server Pages (JSP) Technologies
* 5. Enterprise JavaBeans (EJB) Technology
* 6. Java Persistence API (JPA)
* 7. Java Server Pages Standard library (JTSL)
* 8. Java Database connectivity (JDBC)
* 9. Internet connection (LAN, Wi-Fi).
* 10. Third-party web browser (Firefox, Chrome, Edge)
* 11. Server (GlassFish Server)
* 12. SQL server for database (MySQL)

# 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**



# 8. Project Change Log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Project Change Log for GTLSS | | | | |
| ID | Change Description | Document Changed | Date Revision | Comments |
| 001-D | Change of DB tables | Project Design Plan | 04/27/2019 | Changed Product Order Details to Ordered\_Product and removed Service Order Details. Consolidated Customer Billing table with Customer table. Consolidated Services table with Product table. Added Category table. Removed employees and equipment tables. |
| 002-D | Removal of Calendar Scheduling feature | Project Design Plan | 04/27/2019 | Removed any reference to Calendar Scheduling feature. |
| 003-D | Removal of the Admin Login and Admin Portal pages | Project Design Plan | 04/27/2019 | Removed any reference to Admin Login and Admin portal. |
| 004-D | Change of diagrams | Project Design Plan | 04/27/2019 | Changed the following diagrams:   * ERD, * High Level ACD, * ACD-User Interface, * ACD-Services, * ACD-Domain Objects, * Data Flow Diagram, * Inter-module Dependency Diagram, * Inter-process Dependency Diagram, * User Interface workflow and pages mockups.   Changes included:   * Removed calendar, calendar lookup, admin login, all domain objects. * Added product, category, order\_product, customer\_order, and customer domain objects. |
| 005-D | Addition of additional tools and methods to 6. Techniques on page 24 | Project Design Plan | 04/15/2019 | Added more tools and methods:   * Integrated Development Environment - NetBeans IDE 8.2 All bundle * Language – HTML/CSS/JavaScript/SQL/Java * Platform – JDK 8.0 or newer * Internet connection (LAN, Wi-Fi). * Third-party web browser (Firefox, Chrome, Edge) |
| 006-D | Update of the Development Schedule | Project Design Plan | 05/02/2019 | Updated of the Development Schedule |