Software Design Document

Supply Chain Management in Logistics Management

LinkLogistix

Version 1.0

Printed: November 7th, 2018

LinkLogistix

Team:

Nistha Singh, Aditi Goyal, Advay Agrawal

Supervisor: Prof. Anand M S

PES University

RR Campus

Revisions

Version	Primary Author	Description of Version	Date Completed
1.0	Aditi Goyal, Nistha Singh, Advay Agrawal	Initial Release	10/27/23

Table of Contents

1. Introduction

1.1 Purpose

The Software Design Document describes the architecture and system design for LinkLogistix, aroad trip planning website. LinkLogistix is designed to help travelers plan and oversee their trip.This document is intended for Project Managers, Software Engineers, and anyone else who will be involved in the implementation of the system.

1.2 Scope

This document describes the implementation details of the LinkLogistix (LLX). LLX will consist of six major components: Supplier Registration and Management, Transportation, Warehousing and storage, Inventory Management, Order fulfillment, Demand Forecasting, and Reverse Logistics. Each of the components will be explained in details in this Software Design Document.

1.3 Definitions, Acronyms and Abbreviations

Acronym	Meaning
SCM	Supply Chain Management
SDD	Software Design Document
OS	Operating System
API	Application Programming Interface
LLX	LinkLogistix

1.4 References

-Google Maps JavaScript API

-Bootstrap React

-Anastasov, Nick. “Making Your First Web App with React.” *Tutorialzine*, 22 Apr. 2015, tutorialzine.com/2015/04/first-webapp-react.

-Mead, Andrew. “The Complete React Web Developer Course (with Redux).” *Udemy*, May 2018, www.udemy.com/react-2nd-edition/.

-Njeri, Rachael. “React Apps with the Google Maps API and Google-Maps-React.” *Scotch*, Oct. 2018, scotch.io/tutorials/react-apps-with-the-google-maps-api-and-google-maps-react.

-Ackah, Beverly, Hirji, Shaila, Wirtz Frederick “*Software Requirement Specification”* Oct.2018, https://docs.google.com/document/d/1aLGlXoLIEamAil1PFfOe-jgGSpmQ2hZ8G58vHGNznX4/edit?usp=sharing

2. System Overview

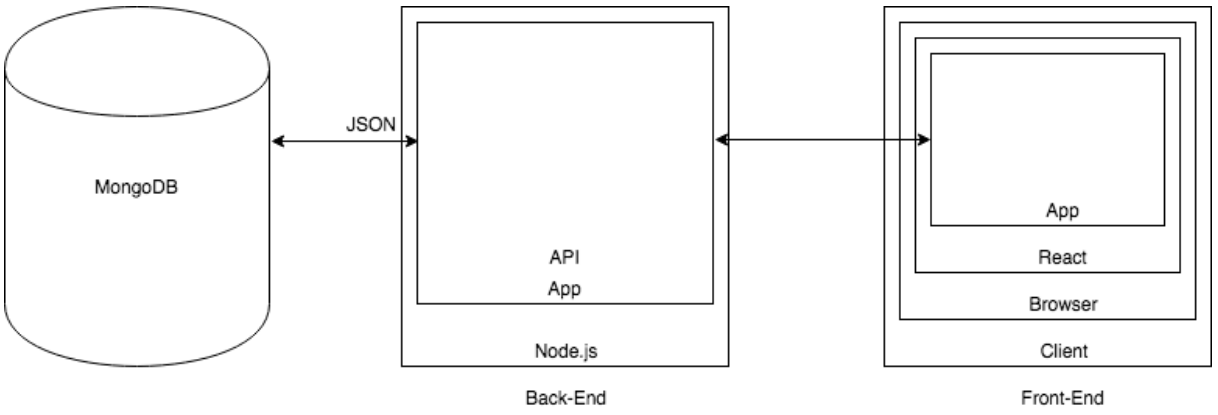


FIGURE 1

Figure 1 above represents the architectural structure we have chosen for the development of the LLX. The backend will be incharge of communicating and pulling information from all the APIs the LLX system is dependent on. We decided to go with NodeJS as our framework as it provides fast, efficient and tight coupling between the client and the server amongst other additional features. NodeJS is also highly scalable which will allow the system to grow further and accommodate a wider range of customers.

The LLX system will also be storing information like Supplier names and prices, shortest routes, warehouse locations, inventory and stock, etc. Storing these details will enable our system to be more efficient in the longrun as we can visit the database for same route trips and just make real time updates on the time duration of the trip. Saving route details will also allow the system to suggest trips to users based of other users with similar preferences. Storing user information will enable the system to maintain userprofile and keep records of their trips.

Additionally, by following this architecture and structure, we can further extend,if we wish, the RTA system into being a mobile application since the backend will be all set.

3. System Components

3.1 Decomposition Description

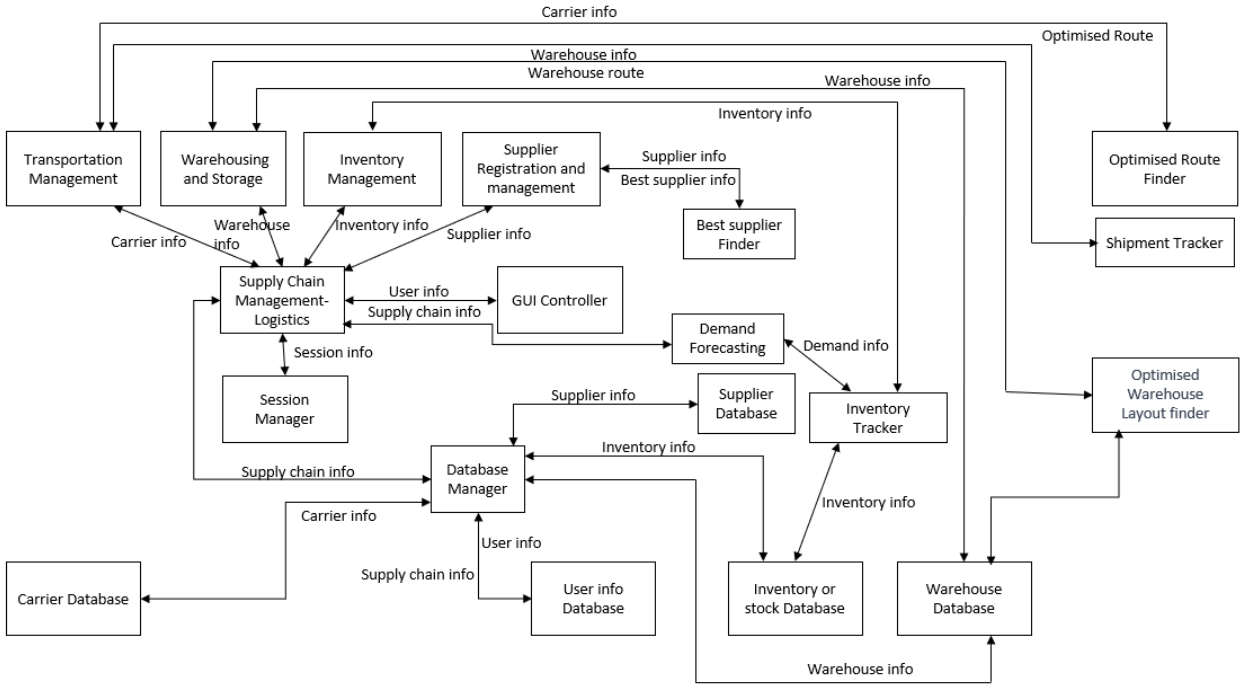


FIGURE 2

Figure 2 above shows a top down description of how the web application of LLX is expected to work and howcomponents will interact with one another. The Supply Chain Management- Logistics module is the main component. The Transportation Management Module is used for shipment route optimization where the most cost effective carriers (transportation providers) and routes are selected using carrier info from the Carrier Database using the Optimized route finder. Inventory Management Module dynamically monitors and updates stock levels based on the results of demand forecasting and present stock levels using data from the Inventory Database. It also notifies when stock levels fall below a certain threshold. Demand forecasting uses algorithms, recent trends and historical data to predict future demand. Warehousing and storage module manages storage facilities, layouts, and inventory retrieval processes. Its main job is Warehouse Layout Optimization where it determines the best configuration for efficient storage and retrieval, for which it uses the Optimized warehouse layout finder, which uses data from the Warehouse database. The Supplier Registration and Management module designed to centralize and streamline the process of enrolling new suppliers, as well as to oversee the ongoing relationship, performance metrics, and interactions with existing suppliers. It finds the best supplier (using Best Supplier Finder), who is most cost effective while having best quality of product. The session manager is for logging in, logging out, and authenticating users. The Supply Chain Management- Logistics module can get and update user profiles that are saved with the database manager. It can also access the data saved in all databases. Finally, the GUI controller, shows how the user interacts with the LLX website.

3.2 Dependency Description

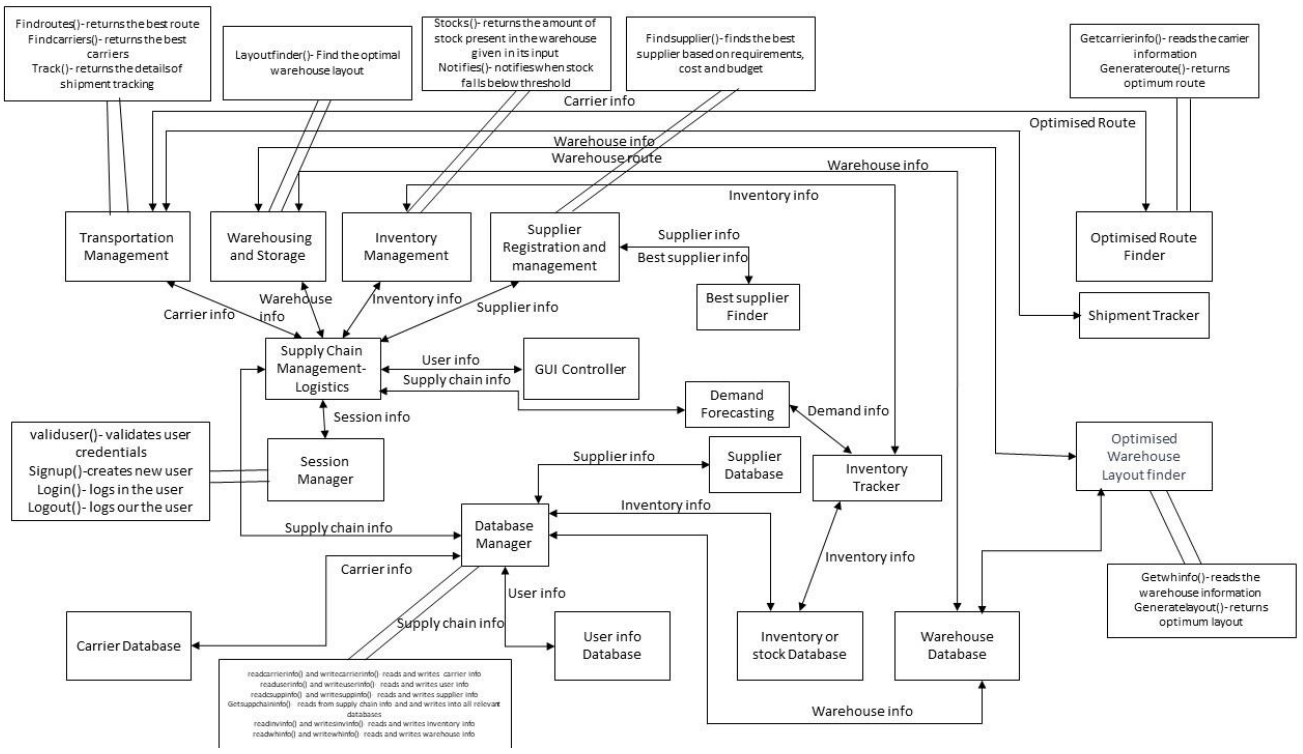


FIGURE 3

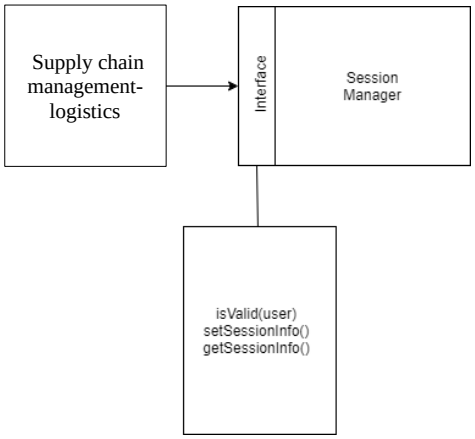
The Supply Chain Management Supplier Relationship Dependency involves the reliance on specific suppliers for the timely delivery of raw materials or products. The quality, lead times, and reliability of suppliers can significantly impact the efficiency of the entire supply chain.

Transportation Mode Dependency involves the choice of transportation modes (e.g., trucking, shipping, air freight) can affect transit times, costs, and the overall distribution network. Dependency on a particular mode can have implications for service levels and cost optimization.

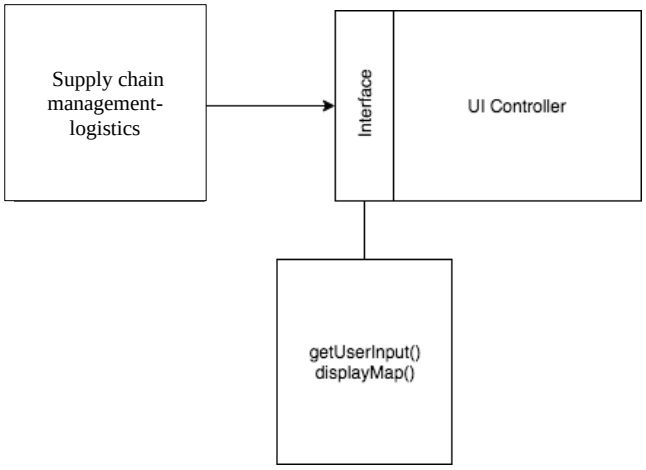
Warehouse and Distribution Center Dependency involves Managing the flow of goods through warehouses and distribution centers is critical. Dependency on these facilities' locations and capacities affects inventory management and order fulfillment.

3.3 Interface Description

3.3.1Trip Planner to Session Manager Interface



3.3.2 Trip Planner to UI Controller Interface



3.4 Module Interfaces

the diagram we briefly touched on in Software engineering.

3.5 User Interfaces (GUI)

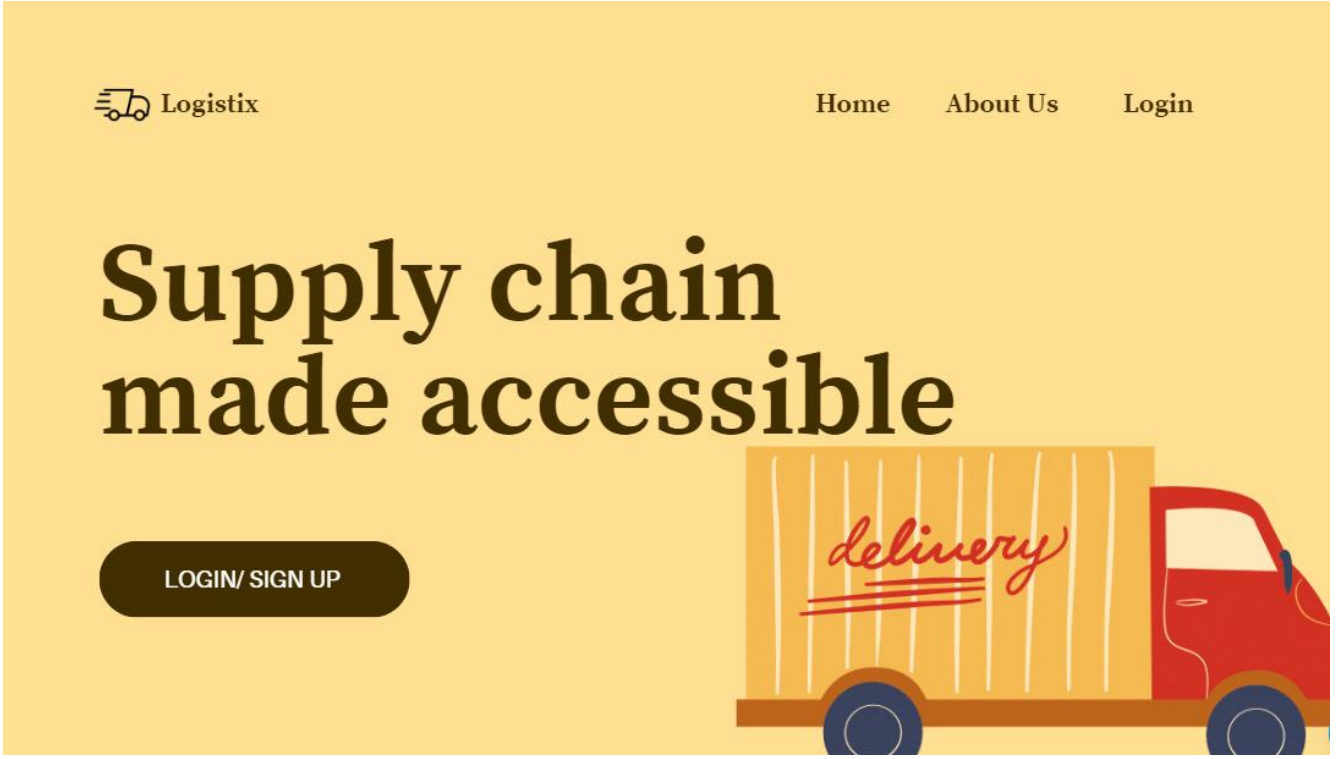


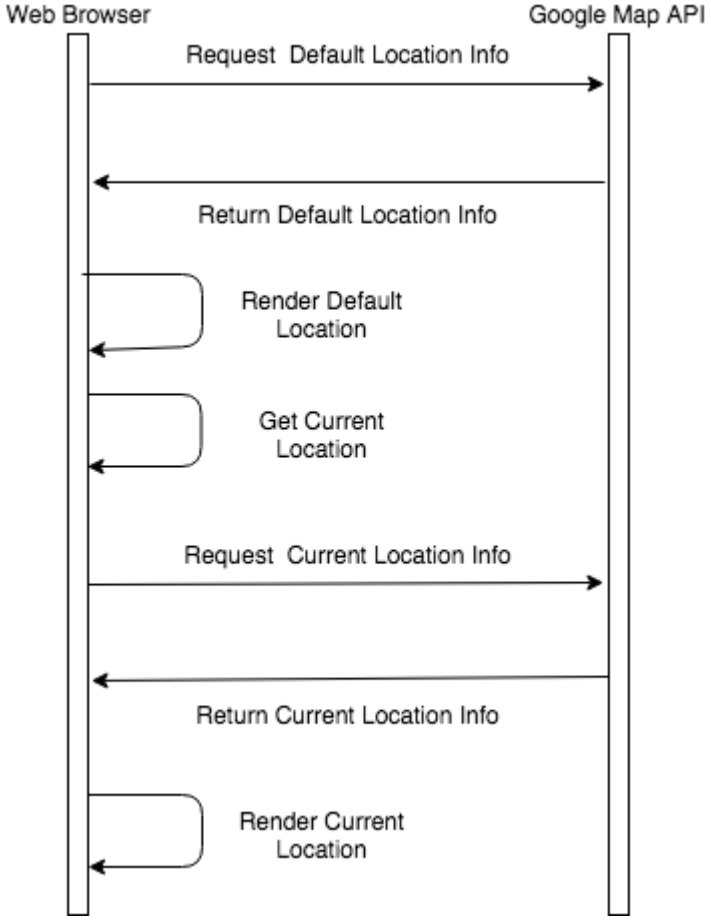
Figure 1 - Landing Page

1. Detailed Design

1.1 Module Detailed Design

1.1.1 Mark Current Location

Sequence Diagrams



Pseudocode

Browser shows default location on google maps

    Request default location information with google's api

    Google returns default location

Uses browser to get current location

Request current location information from google api

Google's API returns current locations info

Browsers renders current location

1.2 Data Detailed Design

1.3 RTM

Requirement-ID	Requirement Description	Design Component	Test-Case #
3.2.1.1	User access to website	Session Manager	3.1
3.2.1.2	Register a profile	User Profile and Database Manager	3.16
3.2.1.3	Find a route for transportation	Optimized Path Finder	3.7 3.8 3.9 3.10 3.11 3.12 3.13
3.2.1.4	Login to profile (edit, plan, view saved trips)	Session Manager	3.14 3.15 3.16 3.18 3.19 3.20
3.2.2.1	Edit/adjust details for supply chain.	Database Manager	3.22