Lab 1 – Load.In Product Description

Gregory Kukanich

Old Dominion University

CS411

Janet Brunelle

03/28/2021

Version 3

Table of Contents

List of Figures

List of Tables

1 Introduction

Do it yourself (DIY) movers often lack the knowledge to handle the logistics of their move. Professional movers ease the physical burden of moving and handle logistics for the customer, but their costs are often very expensive (Wood, 2020). DIY movers must estimate costs, keep an inventory of the items being moved, pack all their items, and load the moving truck efficiently and safely, the last of which is an extremely important part of the moving process. Weight distribution plays an important role in truck and item safety when transporting one’s belongings - an improperly loaded moving truck could result in damaged items or a motor vehicle accident.

Market research has shown that nearly 70% of movers choose to do it themselves (Wood, 2020). This is primarily a result of the high costs that professional movers charge for their services. For example, a four-bedroom household conducting an in-state move would have to pay nearly $3000, and if the move is out of state that cost nearly triples (Meyers, 2018). These prices are not affordable for most Americans, who therefore choose to move themselves. Further market analysis shows that there is a shift in the market towards software-based solutions for helping with the moving process. While there is currently some software available to help a person move, it is basic and helps with issues such as keeping an inventory for the move. No current software will help the mover to the degree that Load.In does.

Load.In uses computer vision to create 3D models of all the customers’ items and creates an artificial intelligence (AI) generated loading and unloading plan of how to properly load and unload the moving truck. Load.In also handles all the logistics planning for a DIY move. This aids the user with things they may not have the knowledge or experience to handle. DIY movers with the help of Load.In pay the normal costs of a DIY move, but now have the knowledge of a professional mover.

2 Product Description

Load.In is a software-based solution that has an Android application that assists DIY movers by providing 3D modeled loading instructions for the truck and logistical support for the move. Users take photos of the packed boxes and any furniture they are moving. Load.In uses these pictures to create 3D models of all the items in the moving truck to give the user a full visualized plan of how to load the truck properly.

The Load Plan that is generated guides the user on how to load their moving truck while accounting for important aspects such as weight distribution, fragility, size, and importance of the item. Users can search the Load Plan to find items should they need to know where it is after they have finished loading the moving truck. Load.In provides tips and tricks for properly packing different materials and items, in the application to assist the mover. Load.In’s time and cost estimates are important features that help the user understand how much time and money they can expect to spend on their DIY move. The goal of Load.In is to deliver a product that allows users to have the expert knowledge and experience of a professional mover without having to pay the high costs that they charge.

2.1 Key Product Features and Capabilities

The Load Plan is the main feature of Load.In. After the user takes photos of items they are moving or enters the dimensions of the items, a 3D model of each item is generated with accurate dimensions. The 3D models are used in combination with a 3D model of the rental truck to generate the Load Plan for this move. The Load Plan features step by step instructions to ensure the user understands how and where to load every item on the moving truck. This is important as this is one of the advantages that professional movers have: they possess the knowledge and experience to load a truck properly. With Load.In every user can load the truck like a professional.

The Move Inventory is a list of all items that a user is moving. Items are added to the move inventory when the user takes a photo of the item or enters the dimensions of the item manually. Users can add the contents of boxes so they can keep track of important items or documents that are already packed during the move process.

Load.In provides users with time and cost estimates based on the number of items they have and the size of the rental truck they need. Once Load.In has all the 3D models generated for the Load Plan, the user can then choose which truck they would like to rent. For this, the application provides the costs, the number of trips, and the total time required to complete their move based on the different truck options available. This gives the user the ability to customize their move, within reason, to what is best for their needs and provides all the logistical information that they need for their move.

Load.In features an Expert Tips section that provides helpful information from expert packers and movers. Should a user need help on how to pack something fragile or needs information about different aspects of moving, they can search through the Expert Tips section for useful and relevant information pertaining to their item. While adding items to their move inventory users will also see a small notification appear that offers packing tips specific to the item being added.

The chatbot that Load.In features is available to help users easily find solutions from the expert tips section or if there is no answer to their question the chatbot redirects the user to a live expert to assist the user with their inquiry.

Load.In offers users the ability to see real-time rental truck prices and availability. Load.In pulls this data from third-party rental companies. These rental companies have access to the rental management dashboard that Load.In features where they can see analytics regarding their truck rentals facilitated through the application.

Load.In has data analytics to look for ways to improve the product. Such as location data, costs, and inventory data, which is all analyzed and used to improve the application. Through the analysis of move distances, costs of fuel, and rental prices Load.In can improve its time and costs estimates for users. Load.In also features a post move feedback system. The feedback from users after they finish their move is crucial for improving the application and improving the AI used for generating the Load Plans.

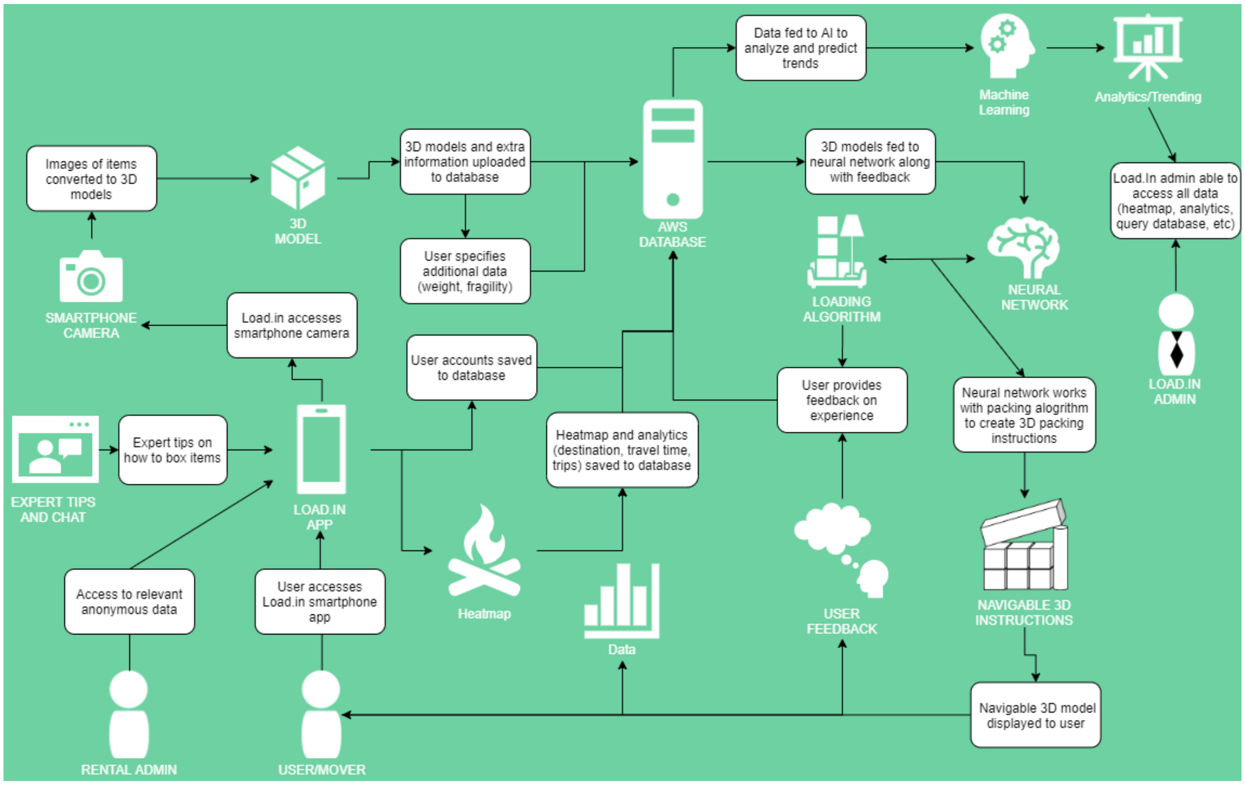
The heat map feature will show where on the application users visit the most or where users seem to struggle and close the application. This information can help developers identify what parts of the application are working well and problem areas of the application where users face issues or are driven away by the user interface.

2.2 Major Components (Hardware/Software)

Load.In consists of a smartphone application, database, website client, and a web API. Figure 1 shows the Major Functional Component Diagram for Load.In. This diagram shows how Load.In will function and how the different components interact. Movers use the Android application to input their items in the Move Inventory, this is stored in the database and is processed through the loading algorithm to generate their Load Plan.

Figure 1

Load.In Major Functional Component Diagram


Data fed to AI to analyze and predict trends
Machine Learning
Analytics/Trending
Images of items converted to 3D models
3D MODEL
3D models and extra information uploaded to database
AWS DATABASE
3D models fed to neural network along with feedback
Load.In admin able to access all data (heatmap, analytics, query database, etc)
SMARTPHONE CAMERA
User specifies additional data (weight, fragility)
LOADING ALGORITHM
NEURAL NETWORK
Load.in accesses smartphone camera
User accounts saved to database
LOAD.IN ADMIN
User provides feedback on experience
Neural network works with packing alogrithm to create 3D packing instructions
EXPERT TIPS AND CHAT
Expert tips on how to box items
LOAD.IN APP
Heatmap and analytics (destination, travel time, trips) saved to database
Heatmap
Data
USER FEEDBACK
NAVIGABLE 3D INSTRUCTIONS
Access to relevant anonymous data
User accesses Load.in smartphone app
Navigable 3D model displayed to user
RENTAL ADMIN
USER/MOVER

The most important component of Load.In is the database. For Load.In a relational database using Amazon Relational Database Service (RDS) was created. MySQL and Amazon Lambda handle the database and connections to the back end. The database is important for data storage as the algorithms that are needed for Load.In to generate a Load Plan require access to the users Move Inventory which is stored in the database.

Load.In is an Android application that requires access to the user’s smartphone camera. The user’s smartphone handles part of the workload for the system to minimize the amount of data being transmitted and to limit stress on the Load.In server. Photos taken for the move stay locally on the user’s phone. This requires the user to have some available storage on their smartphone to use Load.In. All other components require a stable internet connection from the smartphone to be functional.

The website client for Load.In serves multiple purposes but is not used for the moving process itself. The first purpose of the website client is to serve as an information tool about Load.In. Secondly it serves as an analytics tool for administrators. The client displays all the current analytical data for the administrators to look at and use. The website has minimal hardware requirements and works on any updated browser.

The web API for Load.In is created using Java and uses Amazon Web Services Elastic Beanstalk (AWS Elastic Beanstalk) and Apache CXF with Tomcat. The web API allows for the smartphone application and the website client to retrieve, add, or alter information that is stored in the database. The web application is created using Java and is using AWS Elastic Beanstalk and Spring Model-View-Controller (Spring MVC) with Tomcat. The web application provides information from the web API to the website client.

Vendor synchronization for Load.In is created using Java and uses AWS Lambda. The vendor synchronization operates on a timed schedule. It brings in data from third-party vendors such as rental truck companies to provide rental truck information and availability details to the user.

3 Identification of Case Study

Load.In is being developed for people moving anywhere in the country that want to conduct a DIY move. For the case study, the focus will be on a family of three moving across their current city of residence. The average house size in the United States is 2,200 sq ft so this will be used for the case study (US Census Bureau 2019). The family will use a rental truck to move, as their already owned vehicles are not large enough to move the family’s possessions. For the case study, larger items such as furniture will be included to see how well the 3D model generation and Load Plan works. Load.In is expected in the case study to estimate the costs for the move, estimate the time to complete the move, generate 3D models for all items, and generate a 3D modeled Load Plan for loading the rental truck. The goal is to ensure that Load.In can complete a normal cross-city move and display the ability to be scaled up in the future for a bigger sized move or a move over longer distances.

4 Product Prototype Description

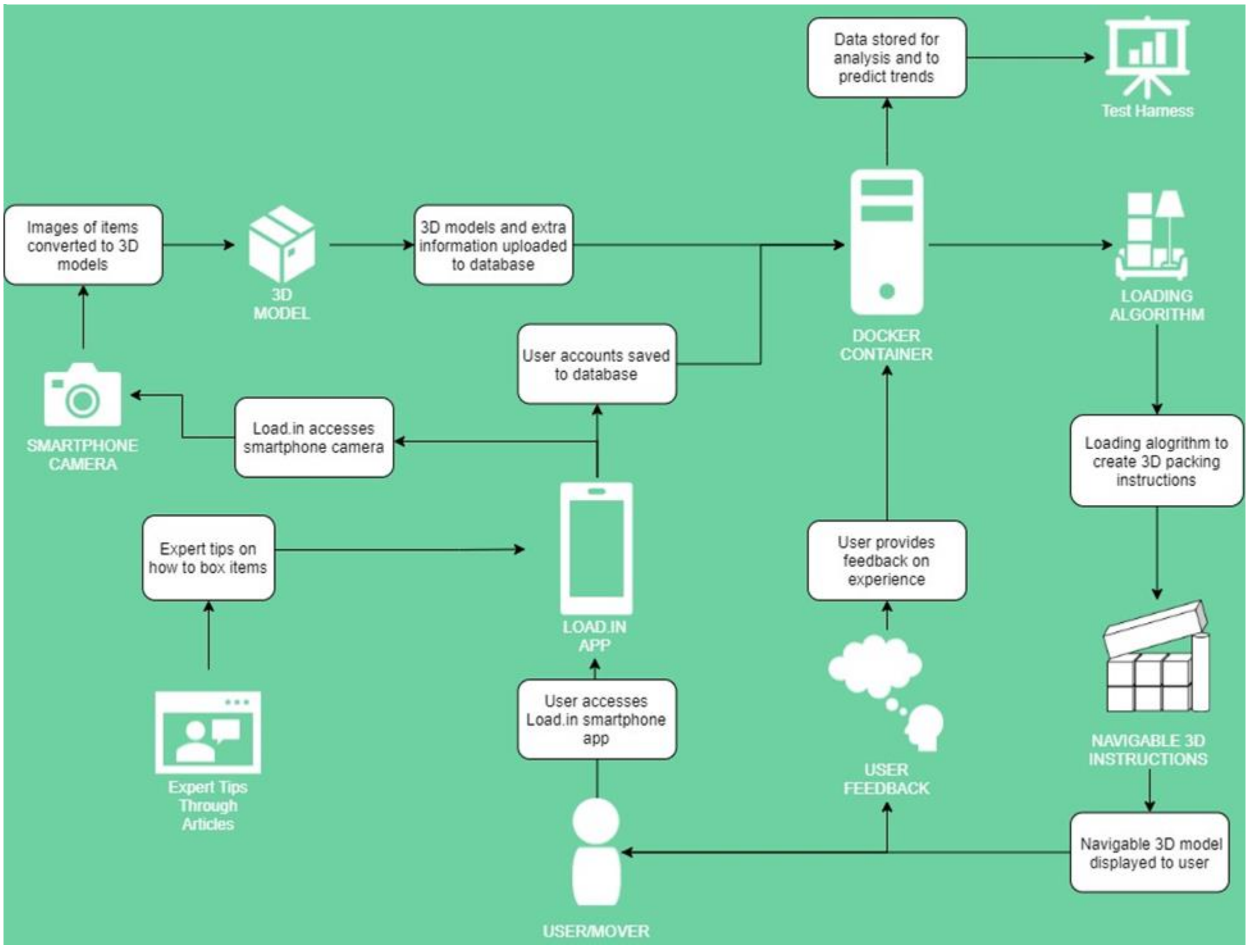
The Load.In prototype will be created and tested to be a proof of concept of the key features and ideas behind the application. While the application has reduced functionality from what was initially envisioned, it will allow for further development to be done on the key features during the prototype process.

4.1 Prototype Architecture

The prototype for Load.In will be hosted on a CS department virtual machine operating using the Ubuntu Linux operating system. The prototype uses Docker containers running Apache Tomcat, Apache CXF, and MySQL. Apache Tomcat and CXF are for the web server and web API. The MySQL container is for the database to store user data and move data. Using the web API, the smartphone application can connect to the MySQL database. The prototype also features a test harness that has an interface for testing and demonstration of the prototype. The Major Functional Component Diagram for the prototype can be seen in Figure 2.

Figure 2

Prototype Major Functional Components Diagram


Data stored for analysis and to predict trends
Test Harness
Images of items converted to 3D models
3D MODEL
3D models and extra information uploaded to database
DOCKER CONTAINER
LOADING ALGORITHM
User accounts saved to database
SMARTPHONE CAMERA
Load.in accesses smartphone camera
Loading alogrithm to create 3D packing instructions
Expert tips on how to box items
LOAD.IN APP
User provides feedback on experience
NAVIGABLE 3D INSTRUCTIONS
Expert Tips Through Articles
User accesses Load.in smartphone app
USER FEEDBACK
Navigable 3D model displayed to user
USER/MOVER

The Load.In prototype as seen in Figure 2, has limited capabilities but is being developed to prove the key features of Load.In such as the Move Inventory, 3D model generation, and the Load Plan function as intended. The prototype will be able to generate a Load Plan that is able to be used to demonstrate the impact Load.In will having in assisting a DIY mover.

4.2 Prototype Features and Capabilities

The Load.In prototype is being built to demonstrate a proof of concept of the important core features of the application. The features of Load.In are listed in Figure 3 along with how they will be implemented in the prototype.

Table 1

Prototype Features Table

Feature	**Description**	**Implementation**
	UI	
Login Page	Create a landing page for the login screen.	Full Functionality
Guess User Interface	Ability for a new user to view the application without being logged in.	Full Functionality
Admin User Interface	Ability to have super user privileges when interaction with the application.	Full Functionality
DIY User Interface	Create a user role for the end user with normal privileges.	Full Functionality
	Authentication	
User Registration	Create an account inside of the load.in application.	Full Functionality
User Login	Authenticate with the server that an account exists.	Full Functionality
Reset Login	End user to be able to reset his/her own password inside of load.in application.	Partial:
	Move Inventory	
Furniture/Item Measurement	Ability to measure items using the phone camera.	Partial: Ability to manually type in dimension of boxes.
3D Model Generation	Generate a 3D Model based on item measurements	Partial
Box Locator Search Feature	Ability for the user to find the location of a box via search	Full Functionality

	Move Plan	
Load Plan	Generate a plan that will show users where and how to load boxes	Partial: Preloaded data inside of the database
	Logistics Planning	
Estimated Number of trips	Calculate the number of trips one will take based on the truck size.	Full Functionality
	Expert Help	
Packing Tips and suggestions	A list of tips will be suggested to the user on how to load the truck properly.	Full Functionality
Tips Search	Ability to search items an end user needs additional help loading.	Full Functionality
Move expert articles	Expert level tips written by professionals.	Full Functionality
	Analytics	
Feedback data	Ability for the end user to provide feedback for a move.	Partial: User can only give a thumbs up or down.

The first features that will help in the move process are the packing tips and expert articles. Prototype users will have the ability to search for help on how to pack items around their house. The next important feature in the prototype is the measuring of the boxes. Prototype users can use their smartphone camera and take a picture of the boxes they have. The application will use the images to generate measurements and a 3D model of the box. Should they have any issues with this, the user is able to input the measurements manually. This demonstrates the ability to generate 3D models of boxes even if there are issues with the phone’s camera measurements.

The Load Plan is another feature that will be developed in the prototype. Once all the 3D models of the boxes have been generated, the application will create a Load Plan showing how to properly pack the boxes on the moving truck. If the user makes any mistakes during the loading process, they can go back a step of the Load Plan, and if the user does not like the Load Plan created, they have the option to generate a new one.

The prototype will also feature trip estimation to allow the user to see the estimated time for their move based on the number of items and distance to travel. This is important to allow the user to see an estimate of how much time their move will take. At the end of the move, Load.In has a feedback page to allow the user to provide feedback on their move experience. This feedback allows for improvement of the application by learning where the application may need small changes to improve the process.

The goal of the Load.In prototype is to show that the key ideas and features behind Load.In are feasible on a smaller scale. By doing this Load.In can demonstrate to prototype users the potential usefulness of the application in a real-world situation.

4.3 Prototype Development Challenges

While the prototype for Load.In has been scaled back to allow for the demonstration of the more key concepts and features, there are still challenges during the development process. Moving involves many variables and it requires much more time developing Load.In to build an application that addresses and handles all these different variables. To handle a lot of this, there is a test harness. This will allow for the developers to set and change different values in the database of items such as boxes, truck sizes, and more when testing the different features of

Load.In. Testing is an important part of the development process for Load.In as there are so many aspects of the move. If one thing goes wrong the whole move process can be affected. By having the test harness and repeatedly testing the application, the developers can address as many of the issues as possible.

5 Glossary

3D – A three-dimensional form or appearance.

Administrator – Someone who will access elevated features of the Load.In system in order to maintain and detect issues.

Amazon Lambda – A serverless compute service that allows code to be run without the need for provisioning or managing servers.

Amazon RDS – A distributed relational database service provided by Amazon Web Services.

AWS – Amazon Web Services: a cloud platform on which Load.In’s databases are hosted.

Android – A mobile operating system based on a modified version of the Linux kernel and other open-source software.

Android Client App – The client-side application for Load.In which runs on the Android platform.

API – Application programming interface: an interface for programs to share information and functionality with one another through a series of calls or connections.

AWS Elastic Beanstalk – An orchestration service offered by Amazon Web Services for deploying applications which orchestrates various AWS services including EC2, S3, Simple Notification Service, CloudWatch, autoscaling, and Elastic Load Balancers.

AWS Elastic File Storage – An AWS service that provides file storage with the ability to auto scale up with increased demand.

Apache CFX – A popular library for hosting web APIs.

Apache Tomcat – An open-source implementation of the Java Servlet, JavaServer Pages, Java Expression Language, and WebSocket technologies. Tomcat provides a "pure Java" HTTP web server environment in which Java code can run.

Chatbot – A feature within Load.In that provides information to users and guides them towards helpful articles and other resources interactively.

Cloud – A term used to describe several computing models such that a company or individual can purchase resources for hosting a variety of things in a centralized location accessible from anywhere in the world.

Computer Vision – a subclassification of artificial intelligence that involves computing information about the world from various sensory data such as images. Techniques of this classification are used throughout Load.In to observe real world objects.

Container Loading Algorithm – A type of algorithm that attempts to optimally fill a three-dimensional space with physical objects. Load.In uses this kind of algorithm to generate Load Plans.

CPU – Central processing unit: the primary component of a computer that processes instructions.

CSS – Cascading style sheet: a style sheet language that is used for formatting the layout of Web pages.

Do-It-Yourself (DIY) Mover – Non-professional movers who rent a truck for their move, and handle all packing, unpacking, and manual labor themselves. This is the primary end user of Load.In.

Expert Tips – Feature of Load.In that allows for a mover to search for helpful articles pertaining to a variety of useful information on how to accomplish various tasks during a move.

GHZ – Gigahertz: a commonly used unit when measuring computer processing speeds.

Guest – Someone who is accessing the Load.In system anonymously and has not registered for an account or someone who has registered but has not authenticated to the system at the time of access.

GUI – Graphical user interface: the aspect of a software program that the end user interacts with.

HTML5 – Hyper Text Markup Language version 5: a markup language used for structuring and presenting content on the Web.

Heatmap – A data visualization technique that shows magnitude of a phenomenon as color in two dimensions.

Java – A set of computer software and specifications that provides a system for developing application software and deploying it in a cross-platform computing environment.

JavaScript – A scripting language that runs in the browser and performs one or more functions to animate an otherwise static HTML document.

Linux – An open-source and community-developed operating system for personal computers and workstations.

Load Plan – A set of instructions on how to optimally load a container – generated automatically by Load.In from the boxes and furniture input into the system by the user.

Logistics Planning – A feature of Load.In that assists the mover with determining what rental trucks cost, how many trips the truck might need to take and whether the truck is available to rent based off proximity to the mover.

Mbps – Mega-bits per second: a unit of measurement for network speeds.

Megapixel – One million pixels: typically used to measure the size and quality of images.

Move Analytics – A feature of Load.In in which information gathered from previous moves are used to determine estimations for future moves as well as predict market trends for Rental Companies.

Move Inventory – A feature of Load.In that catalogs all boxes and items the mover intends to move.

MySQL – An open-source relational database management system.

MacOS – An operating system used on Apple's MacIntosh line of personal computers and workstations.

OS – Operation system: a collection of programs designed to provide a platform on a device to run other applications and typically provides a layer of abstraction from the hardware it interacts with.

Pixel – A small square of color that is part of a larger display screen or image.

Photogrammetry – A computational method of deriving three-dimensional information from images. This method is used in Load.In to construct 3D models of boxes, furniture, and other items from pictures taken from the end user's cell phone camera.

PNG – Portable Network Graphics: a common image file format that Load.In uses.

Professional Mover – Professionals who handle the physical labor of loading and unloading a moving truck as well as driving the truck to the destination.

Rental Administrator – A representative of a rental company who will access the Load.In system on behalf of the rental company.

Rental Company – Any company that rents moving vehicles for a Do-It-Yourself Mover to assist them with their move.

Rental Estimate – A feature provided by Load.In that pulls data from the internet to determine the cost of renting a moving truck.

Smartphone – A device, typically handheld, that can act as both a cellular phone and a computer by running one or more applications typically through a touch screen interface.

SPRING MVS – An application framework and inversion of control container for the Java platform.

Test Harness – A set of special features used during the development of Load.In to enable testing and demonstration of the application.

Vendor Synchronization – A feature of Load.in that brings in truck sizes and availability of rental information from third party moving company websites.

Windows – An operating system developed by Microsoft for use on personal computers and workstations.

6 References

Andrew, P. (2020, January 26). Is your house the “typical American home”? *Hsh.* https://www.hsh.com/homeowner/average-american-home.html

CADCode Systems. (n.d.). *Optimizing & machining.* CADCode. Retrieved September 20, 2020, from https://www.cadcode.com/category/categories/optimizing-machining

Collins, T. (2018, April 20). A look into photogrammetry and video games. *Medium*. https://medium.com/@homicidalnacho/a-look-into-photogrammetry-and-video-games-71d602f51c31

Dube, E. (2006). *Optimizing three-dimensional bin packing through simulation.* Semantics Scholar. https://www.semanticscholar.org/paper/OPTIMIZING-THREE-DIMENSIONAL-BIN-PACKING-THROUGH-Dube/bb9986af2f26f7726fcef1bc684eac8239c9b853#references

Economy Moving & Storage, LLC. (2015, January 4). *How to properly pack and load a moving truck- movers Cincinnati* [Video]. YouTube. https://www.youtube.com/watch?v=rjmofUZOdwo&feature=youtu.be

Knoblauch, M. (2019, May 8). One in ten Americans would prefer a week in jail over moving. *New York Post*. https://nypost.com/2019/05/08/one-in-ten-americans-would-prefer-a-week-in-jail-over-moving/

Manwaring, K. (2020, November 24). *The average cost of moving truck rentals.* Move.org. https://www.move.org/average-cost-truck-rental/

Meyers, S. (2018, October 10). *How much does it cost to move a four (4) bedroom house?* Movers.com. https://www.movers.com/moving-guides/moving-four-bedroom-house-costs.html

Nat and Friends. (2017, April 18). *Google Earth's incredible 3D imagery, explained* [Video]. YouTube. https://www.youtube.com/watch?v=suo_aUTUpps&feature=youtu.be

The American Institute of Stress. (n.d.). *The Holmes-Rahe Stress Inventory* PDF. Retrieved September 20, 2020, from https://www.stress.org/wp-content/uploads/2019/04/stress-inventory-1.pdf

US Census Bureau. (2019, October 10). *Historical households tables*. The United States Census Bureau. Retrieved September 2020, from https://www.census.gov/data/tables/time-series/demo/families/households.html

White, M. (2018, May 10). *How to pack & load a moving truck*. Moving.com. https://www.moving.com/tips/loading-truck-rental/

Wood, P. (2020, November 4). 29+ Moving industry statistics. *MoveBuddha*. Retrieved November 20, 2020, from https://www.movebuddha.com/blog/moving-industry-statistics/

Yale, A. J. (2019, March 28). *Report: Where Americans are moving - and how far they're going*. The Mortgage Reports. https://themortgagereports.com/49116/report-where-americans-are-moving-and-how-far-theyre-going