Lab 1 – Load.In Product Description

Byron Aquilino

Old Dominion University

CS411

Janet Brunelle

20 March 2021

Version 3

Table of Contents

List of Figures

List of Tables

1 Introduction

When it comes to relocation, not every American can afford to pay for professionals (Wood, 2020). Every move incurs expenses. These expenses include moving truck costs, packing material, gas, and insurance. In addition to these common expenses, professional movers incur labor costs as well (Wood, 2020). Out of the nearly 31 million moves that occur every year, a staggering 78.3% of movers move themselves (Wood, 2020). Compared to the cost of a "do it yourself" (DIY) relocation over a short distance, a professional relocation over the same distance is on average approximately seven times more expensive because of the additional labor expense (Wood, 2020). DIY movers have a strong incentive to want to move themselves.

However, the lack of professional management is not without its downside. Professional movers bring additional benefits to the mover such as expertly packing items safely and securely, loading a truck safely, and loading a truck efficiently. DIY movers do not know how to pack and load as well as a professional mover can. This lack of expertise can cause issues to a DIY move ranging from frustration and anxiety to damaged property and time wasted due to mistakes (Knoblauch, 2019). Additionally, the DIY mover also faces other difficult logistical considerations such as knowing how big of a truck to rent, how to organize his boxes, and keeping track of all his possessions.

Load.In is a software-based solution that helps the DIY mover acquire expert knowledge. Load.In has features based on Computer Vision, Artificial Intelligence, and 3D model generation to help the DIY mover plan, prepare, and execute his move successfully. This provides an approach that features the best aspects of a professional move with the cost effectiveness of a traditional DIY move.

2 Product Description

Load.In has key features such as Load Plan generation, Move Inventory, Expert Tips, and Logistics Planning. These key features address one or more aspects of the DIY moving problem. In addition to providing benefits to the DIY mover, Load.In introduces several key benefits to the rental company industry as well by improving advertisement of available products, reaching new potential customers, and providing insights into moving trends. The Vendor Synchronization feature scrapes data from rental companies into Load.In so that DIY movers can see what rental trucks are available within proximity of the move and help guide movers to the right choice of rental vehicle based on customized search parameters. Move Analytics, another critical feature of Load.In, provides valuable analytics for consumption by the rental companies so that they can better forecast moving demands and trends. The Feedback feature gets feedback from the customers on truck rental quality as well as how well Load.In performed.

2.1 Key Product Features and Capabilities

The first important feature is the Move Inventory. This feature consists of a user interface where the mover can enter information regarding what is being relocated. This feature keeps an accounting of all boxes, the size and number of each box, what is in each box, and the un-boxed items of the DIY mover. By cataloging everything and uniquely identifying each box, Load.In can accurately account for all inventory items. Once the DIY mover establishes the Move Inventory, the mover can search for an item at any time, know exactly what box that item is in, and the location of the box.

After the mover has cataloged his items through the Move Inventory, he is ready for the Load Plan feature. This feature takes the items from the Move Inventory and optimally suggests placement of each item in a selected truck of a given size. The Load Plan accomplishes several

goals during this process. The first goal it accomplishes is loading everything safely. This means that when Load.In plans the load, it distributes the weight in a manner that is consistent with proper loading practices thereby avoiding unsafe conditions. This feature benefits the customer who may be unaware that distribution of weight can be a safety consideration. The second goal of the Load Plan is to accomplish maximum utilization of space. This benefits the DIY mover by reducing wasted space in the truck so that a mover can avoid making unnecessary round trips. The third goal is to reduce the difficulty of a move by providing detailed instructions on where everything should go on the truck and how to load everything in the correct order. This benefits the DIY mover by reducing the time it takes to solve the problem of how to load the truck themselves.

Because Load.In has both the Move Inventory and the Load Plan, the Load.In solution can accurately establish financial estimations for moving. Each estimate provided is based off a particular truck size. This process of establishing estimations is at the core of what the Logistics Planning feature does. Each estimate includes the number of projected round trips and the estimated costs of the rental. Load.In estimates the cost using an Artificial Intelligence base algorithm using historical costs from other DIY moves. When presented with several different rental options, the DIY mover can determine whether the price estimates and the number of trips provided addresses his needs. For example, a mover who wanted to compare a box truck to a cargo van might want to see whether the cargo van might require more trips than the box truck for their move. When presented with the information, the DIY mover might decide to either optimize his time or his cost savings depending on what is the priority. This is essentially a what-if analysis tool that allows the user to make a knowledge-based decision using the estimates that Load.In presents.

Before a mover establishes the Move Inventory, a mover must pack items into boxes with confidence. The Expert Tips feature of Load.In solves this problem in two ways. When a mover is attempting to pack an item that he wants more information about, he consults the Expert Tips feature for information. The mover can search for tips written by moving experts on how to pack certain items. These tips also contain videos as well as images and text demonstrating exactly what to do and how to do it. The second way that the Expert Tips help is through a friendly push of tips as the mover packs and catalogs their inventory. Load.In uses key words from the description of the box contents to search for tips and display a helpful suggestion as the mover inventories all their boxes. The mover can choose to view the suggestion or ignore it.

If there are no articles to find on a particular subject or if the mover is unable to find the tips, a chat-bot feature is also available that allows for the mover to post a question to the chat-bot. The bot automatically searches for a tip. If the mover is unable to still resolve their issue, Load.In connects them to a live move expert who can help them with their packing needs. This brings the expert knowledge of packing within reach of the DIY mover.

The Vendor Synchronization feature keeps Load.In up to date with the latest information about rental vehicle options and rental inventory. From this information Load.In knows what truck sizes are available, their exact dimensions, where the trucks are available to rent, and from whom the mover can rent them from. The synchronization service connects either with a rental company's APIs or a rental company's website and periodically scrapes and pulls in information into Load.In. This automation eliminates the need for Rental Admins of the Load.In system to enter the data into the system manually though they can if they still choose to.

Logistics Planning requires data which comes from the Move Analytics feature of Load.In. This is because when producing estimates on move costs, Load.In needs historical data. Load.In

keeps track of data in an anonymous fashion including the locations of where people moved, what routes they took and the number of trips they made, the total distance traveled, the gas costs, the rental costs, and the supply costs of the move. Other data that Load.In captures is the aggregate weight of the items moved, the percentages of items that were fragile, and the average dimensions of the non-boxed items. Load.In also tracks standard box sizes and the quantities of boxes per move.

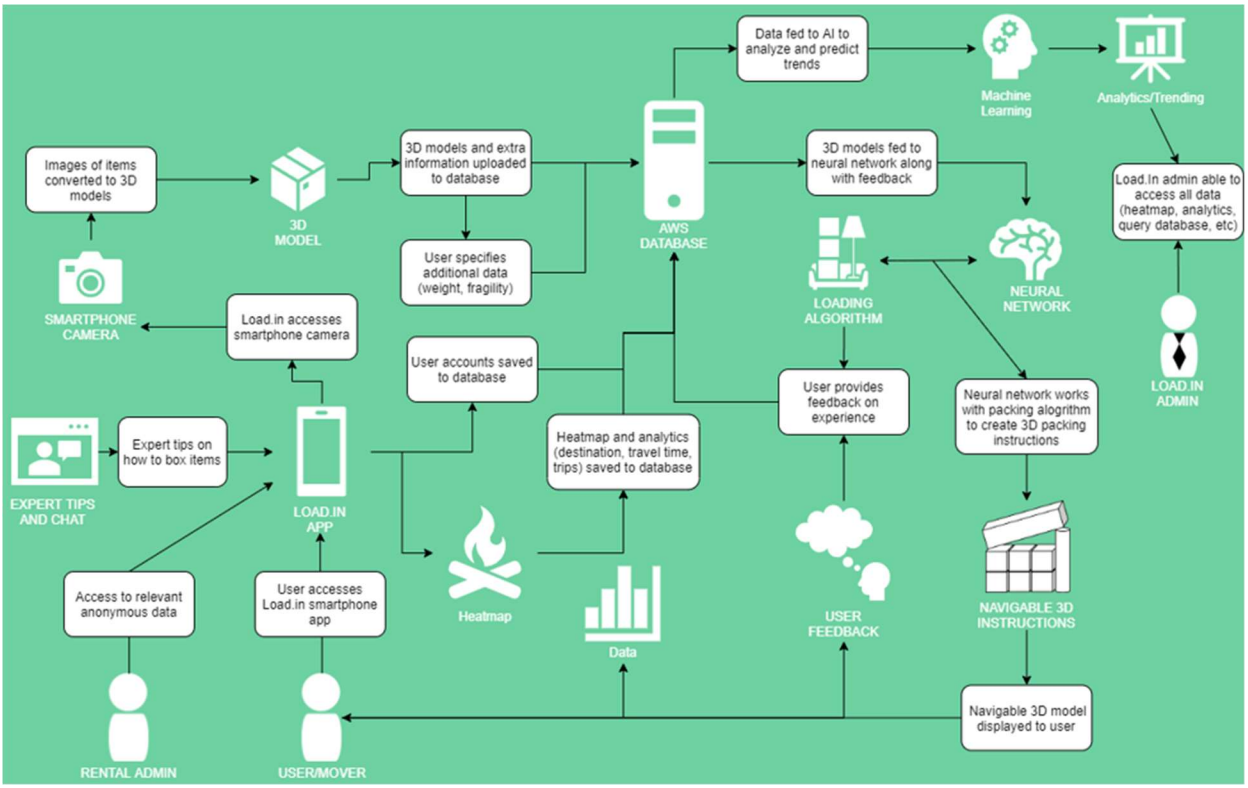
In addition to raw numbers, Load.In collects more qualitative information in the form of mover feedback and ratings. This allows not only Load.In to improve but the partner rental companies as well. Rental companies can search and view mover feedback of their rentals so that they can better serve the needs of future customers.

2.2 Major Components (Hardware/Software)

Load.In's main solution consists of a centralized database, a smartphone client, a website client, a central web API, and a website. Figure 1 shows the major architectural structure of the solution. In the diagram of Figure 1, movers interact with the smartphone client which provides them with the user interfaces for important features such as the Load Plan, Move Inventory, and Logistics Planning. The Load.In smartphone client interacts with hardware in the smartphone to allow the mover to take photos of the items to generate the 3D models.

Figure 1

Major Functional Component Diagram



Since the smartphone client and website must access and share the same data, all data for Load.In resides in a central database. This is also the case when it comes to more than one mover collaborating on a move together. Because each mover must be able to modify the same move inventory, the information must reside in a centralized location. Amazon's RDS, or Relational Database System, provides the hosting platform on which the database software executes. The database software that the client application and website interact with is MySQL.

Because mobility is important and smartphones are highly mobile, the use of a smartphone client is the ideal choice to host the important features regarding a move. Another important consideration when it comes to the smartphone is the available hardware. The Load.In

application needs access to a camera for the Move Inventory and when loading a truck it needs a screen capable of displaying the instructions.

For the smartphone client to run reasonably well and create a great user experience for the mover, there are some minimum requirements that the mover's smartphone must meet. Ideally, for best user experience, the processor on the device should contain at least eight cores with each core measuring around 1.8 GHZ clock speed. This results in enough processing power that the rendering of the 3D models could take place locally without transmission to the cloud. In local rendering mode, the device also needs to support additional storage to facilitate Load.In storing the pictures temporarily on the device while the rendering takes place. Considering a buffer of 100 pictures on the device of around 12 megapixels in quality, the device might need around one to two gigabytes of storage for temporary content. If storage is unavailable or the processor is not fast enough, then Load.In falls back to using cloud storage and cloud processing resources to produce the 3D models required for the Load Plan feature to work. In cloud rendering mode, internet connectivity must be at least 15 Mbps or at least 4G for a cellular network connection.

Load.In's smartphone client runs on the Android operating system with a minimum version of 4.4 so as to support older devices. Load.In utilizes photo compression so that it can reduce transmission times to the web API and the storage requirements of the photos. The PNG file format provides adequate compression while supporting an open standard.

In addition to the DIY movers, Rental Administrators, Administrators, and Guests use the Load.In system. The website client provides a way for these other users to interact with the Load.In system. Their interaction requirements will vary, but the core emphasis of the website client is to display data and analytics. The website client hardware does not need to be

particularly powerful. In fact, a typical thin client with an operating system that supports one of the major browsers capable of rendering a website from HTML5, JavaScript, and CSS will work. For the sake of keeping performance at a recommended level, the website client's hardware needs to have a CPU with at least four cores in it and have at least two gigabytes free of memory when loading the website.

Load.In's web API provides data access and manipulation for the website client and the smartphone client. Java is the programming language for the web API. The web API is built on the Apache CFX web API framework and runs on Tomcat, which runs in an instance of AWS Elastic Beanstalk. It interacts with the database, hosted in MySQL, and utilizes Amazon's Elastic File Storage option for storage of photographs associated with the move inventory.

Load.In's web application serves information to the website clients. Java makes up the primary programming language of the solution although other languages comprise the solution as well. These other languages and frameworks are Spring MVC, HTML5, CSS, and JavaScript. The web application communicates primarily with the web API. AWS, using Elastic Beanstalk and Apache Tomcat, serves as the platform for the web application.

Load.In's Vendor Synchronization functionality resides on AWS using AWS Lambda, which allows for execution of Java code to run on triggers and can scale up into multiple distinct instances. The synchronization process executes on a regular schedule and communicates with the MySQL database to bring in vendor related information into the Load.In system. One such process runs for each vendor that Load.In interacts with.

3 Identification of Case Study

For the purposes of the case study, Load.In will focus its main attention on a typical or otherwise average moving family. According to the Census Bureau, the average family size in

the United States as of 2019 is 2.52 (2020). For the purposes of the case study, the family has three members and one dog. The average house size in the United States is approximately 2,200 sq ft (Andrew, 2020). Therefore, with this house size in consideration and the consideration that this family will have established furniture and other household furnishings, a move via a family vehicle such as a pickup truck would be too much effort for this family in terms of trips and thus requires the rental of a moving truck. This average family is also concerned with budget and would want an accurate estimation of the cost involved. They would want to reduce costs wherever possible. A typical distance of a move for a family would be within 20 miles of their home, so for this case study, they will be moving across town.

For the composition of the family, there is a mother, a father, and a child. The mother and the father would be the DIY movers and they would like to collaborate on the move by sharing a move plan. For the purposes of this exercise, they will need to be able to create a move plan, catalog their inventory, generate a Load Plan, and be able to get rental estimates.

4 Load.In Product Prototype Description

The Load.In prototype attempts to take the most important features from the real-world product and demonstrate them in a way that still proves the functionality and innovation of the system while reducing the features' scope and complexity. The Load.In prototype includes features such as the Load Plan, the Move Inventory, the Logistics Planning, and the Expert Tips.

4.1 Prototype Architecture (Hardware/Software)

Load.In's prototype architecture is distinct from the real-world product's architecture due to the nature of the reduced feature set. Instead of using AWS, the Load.In prototype will be hosted by a series of containers that will be deployed to a single virtual machine running Ubuntu Server 16.04 and Docker, which operates as the main platform for the containers. The

prototype's main components will consist of the Android client application, the web API for brokering the communication to the database, the database hosted on MySQL, and a test harness running as a standard Java application. The web API container will host an instance of Apache CXF which runs on Tomcat and a Linux kernel. The MySQL container will host an instance of MySQL and will ultimately host the data that Load.In will utilize from the web API. Both the Android client and the test harness will interface with the web API in order to exchange data and be able to operate. Figure 2 shows the interactions of these components with one another.

Figure 2

Load.In - Prototype Major Functional Component Diagram

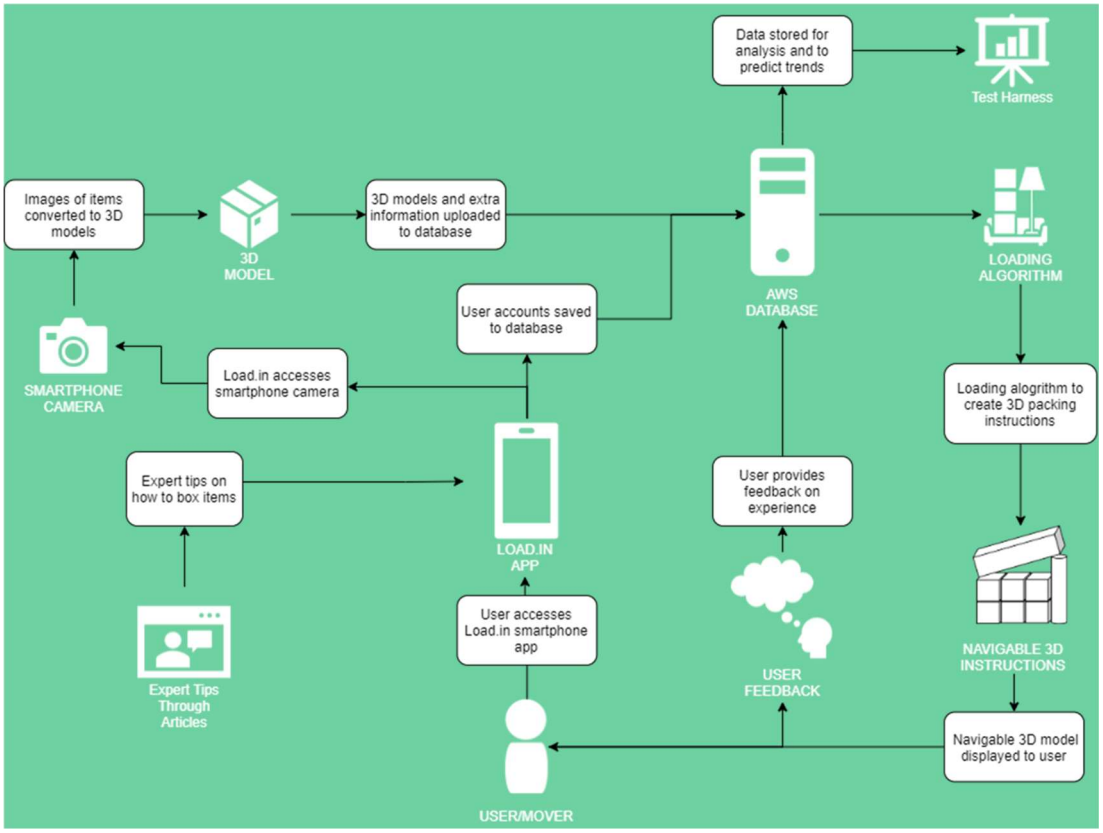


Figure 2 shows the reduced capabilities of the prototype as well. The smartphone client provides the interface for the mover for the interaction with the Expert Tips feature, the 3D model generation, Load Plan display and Move Inventory features.

4.2 Prototype Features and Capabilities

In order to demonstrate the functionality of the prototype, the Android client will serve as the main application and will host the critical features such as the Load Plan, Move Inventory, and the Logistics Planning. The test harness GUI will assist the Android application by allowing a test user to be able to generate data pertaining to a move, setup user accounts, load Expert Tips, and automatically load truck sizes or change truck sizes. This will simulate the other features that are not present in the prototype application such as the Vendor Synchronization. From the Android application the following actions will be demonstrated: measuring a non-box item using photogrammetry, entering in a box dimensions and content description, 3D model generation, locating a box from the inventory, generating a load plan, estimating a move's number of trips, getting packing tips, finding expert articles, and providing feedback for a move experience. These actions demonstrate the core features of Load.In. Table 1 shows a comparison of features from the RWP, also known as real-world-product, versus the prototype.

Table 1

Real World Product versus Prototype

Feature	Real World Product	Prototype
Move Inventory		
Furniture/Item measurement	Fully Functional	Partial
3D model generation	Fully Functional	Partial

Item weight	Fully Functional	Eliminated
Item fragility	Fully Functional	Eliminated
Box locator search feature	Fully Functional	Fully Functional
Move Plan		
Load Plan	Fully Functional	Partial
Truck unloading instructions	Fully Functional	Eliminated
Logistics Planning		
Estimated number of trips	Fully Functional	Fully Functional
Estimated time to move	Fully Functional	Eliminated
Estimated rental truck costs	Fully Functional	Eliminated
Expert Help		
Packing Tips and suggestions	Fully Functional	Fully Functional
Tips search	Fully Functional	Fully Functional
Move experts' articles	Fully Functional	Partial
Chatbot	Fully Functional	Eliminated
Live expert	Fully Functional	Eliminated
Vendor Integration/Data Import		
3rd party vendor web scraper	Fully Functional	Eliminated
3rd party vendor web API reader	Fully Functional	Eliminated
Box dimensions	Fully Functional	Eliminated
Truck sizes	Fully Functional	Eliminated
Truck availability	Fully Functional	Eliminated

Analytics		
Location data	Fully Functional	Eliminated
Move data	Fully Functional	Eliminated
Feedback data	Fully Functional	Partial
Heatmap	Fully Functional	Eliminated
Rental interest statistics	Fully Functional	Eliminated

There are risks that the prototype will have which it must address. There are two main risks from a customer perspective that the Load.In prototype must address. The first risk that the prototype confronts is the mover may not like aspects of his move. The implementation of the feedback feature mitigates this risk, and the feature will be present in the prototype the same it is in the real-world-product. Second, if the end user misses a step in the load instructions generated by the Load Plan feature, the user needs to be able to repeat one or more steps so that this issue does not impact the quality of the move. To solve this, Load.In's prototype will include a feature that allows the mover to back up or rewind any number of steps so that the display can replay steps in the correct order and the mover can make sure they have followed the instructions appropriately.

4.3 Prototype Development Challenges

There will be challenges related to the development of Load.In's prototype. One of the biggest challenges is the development of the test harness and the implementation of functions or routines that will allow for the change of values from the prototype's database such that the client Android application reflects the changes. One of these scenarios that needs to be demonstrated is what happens when the available trucks change on a mover after a mover has

selected his truck for a move. The prototype needs to be able to demonstrate to an audience the impact of such an event on the Load Plan and ultimately the number of trips for the mover. Another challenge is changing a Move Inventory to show the impacts to a Load Plan. For example, a mover might decide to add in other boxes to the Move Inventory, or the mover may have a wide variety of odd-shaped boxes.

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