**Lab 1 – Load.In Product Description**

Byron Aquilino

Old Dominion University

CS411

Janet Brunelle

8 February 2021

Version 1

**Table of Contents**

[1 Introduction 3](#_Toc63665074)

[2 Product Description 4](#_Toc63665075)

[2.1 Key Product Features and Capabilities 4](#_Toc63665076)

[2.2 Major Components (Hardware/Software) 7](#_Toc63665077)

[3 Identification of Case Study 11](#_Toc63665078)

[4 Load.In Product Prototype Description 12](#_Toc63665079)

[4.1 Prototype Architecture (Hardware/Software) 14](#_Toc63665080)

[4.2 Prototype Features and Capabilities 15](#_Toc63665081)

[4.3 Prototype Development Challenges 16](#_Toc63665082)

[5 Glossary 17](#_Toc63665083)

[6 References 21](#_Toc63665084)

**List of Figures**

[Figure 1 *Major Functional Component Diagram* 8](#_Toc63665085)

[Figure 2 *Load.In - Prototype Major Functional Component Diagram* 15](#_Toc63665086)

**List of Tables**

[Table 1 *Real World Product versus Prototype* 12](#_Toc63665087)

# Introduction

When it comes to moving, 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. When it comes to professional moves, professional moves incur labor costs in addition to the regular expenses (Wood, 2020). Out of the nearly 31 million moves that occur every year, a staggering 78.3% of all moves are done by individuals moving themselves (Wood, 2020). Compared to the cost of DIY moving, otherwise known as “do it yourself” moving, professional moving overall is exponentially more expensive (Wood, 2020).

It is no wonder why individuals, who want to save some money, might elect to move themselves. However, the avoidance of incurring these costs is not without its downside. Professional movers bring additional benefits to the mover such knowing how to pack items safely and securely, load a truck correctly, and load a truck efficiently. This expertise is something a DIY mover lacks when compared with a professional because this expertise comes from experience. This lack of expertise and experience can cause issues to a DIY move ranging from frustration and anxiety to damage to property, potential traffic accidents, time wasted due to mistakes, or even overturned trucks (Knoblauch, 2019). Other logistical considerations cause the DIY mover frustration such as not knowing the location of important items or where certain boxes are, how big of a truck to rent, how to pack certain items so they do not get damaged, and what impact to the number of trips does choosing the wrong truck size make.

Since DIY moving is such an attractive alternative to spending big money on professional movers, it would be great if some technology-based solution could solve some of the problems associated with moving to make life easier on the DIY mover. That is where Load.In comes in. Load.In solves the issues associated with a typical DIY move by introducing powerful features based on computer vision and 3d model generation to help combat costs, reduce stress, and bring some of that expert knowledge to the table so that a DIY mover regardless of their experience with moving has the tools on hand to make their move successful and smooth.

# Product Description

Among the features of Load.In are several 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 problems. In addition to providing benefits to the DIY mover, Load.In introduces several key benefits to the rental company industry as well to make it easier for movers to get rentals. The Vendor Synchronization feature brings data from various 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. 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 how rentals went as well as how well Load.In worked for their move.

## Key Product Features and Capabilities

The first important feature to introduce is the Move Inventory feature of Load.In. This feature consists of an interface by which the mover can enter information regarding what the mover is intending to move. This feature keeps an accounting of all boxes, what is in each box, and the furniture of the DIY mover. By cataloging everything and uniquely identifying each box, the DIY mover can keep an accurate accounting of all their possessions. The DIY mover also knows where everything is through this feature. This can be especially helpful if the DIY mover needs access to certain items that may be important. 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 where that box is.

Once the mover has cataloged his/her inventory through the Move Inventory, he/she is ready for the Load Plan feature. This feature will take the inventory from the Move Inventory and figure out a way to load everything into a selected truck with 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 will distribute 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 efficiency 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 make life easier on the DIY mover 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 figure out how to load the truck themselves.

Because Load.In will have both the Move Inventory and the Load Plan, the Load.In solution accurately establishes several estimations for moving. Each estimate provided is based off a particular truck size. This process of establishing several estimations is at the core of what the Logistics Planning feature does. Each estimate includes the number of projected round trips and the estimated time it will take to move everything. Load.In derives the estimate of time based off how long other moves from other DIY movers have taken in the past. When presented with several different rental options, the DIY mover can determine whether the price estimates and the number of trips provided fits his/her 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/her time or his/her cost savings depending on what is the priority.

Before a mover establishes the Move Inventory, a mover must pack items into boxes with confidence. This is where the Expert Tips feature of Load.In comes in. When a mover is attempting to pack an item that they are unsure of how to proceed, he/she would consult the Expert Tips feature for information. With this feature, the mover can search for articles written by moving experts on how to pack certain items. These articles may also contain videos as well as images and text demonstrating exactly what to do and how to do it.

If there are no articles to find on a particular subject or if the mover is unable to find the article, 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 the best article covering the subject. 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.

In order for the Logistics Planning feature to work appropriately, it must rely on accurate and up to date information. 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 web sites and periodically pull in information into Load.In. This automation eliminates the need for some user of the Load.In system to enter the data into the system manually.

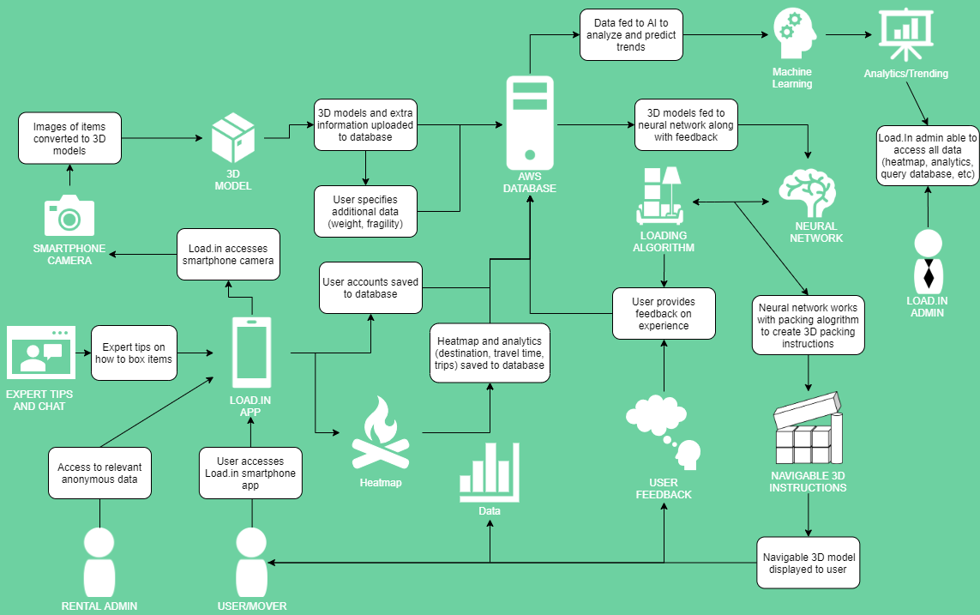
The power behind the move estimates comes from the analytics feature of Load.In. This is because when producing estimates on move times, Load.In needs lots of data. Load.In keeps track of certain data in an anonymous fashion including the locations of where people moved, what routes they took and the 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 items. Load.In also tracks standard box sizes and the quantities of boxes per move. Load.In keeps track of the length in time of the entire move process such that it can determine how long each mover spent on the road, and how long it took for each move to load and unload the truck per item. These analytics help drive better move estimations as well as improve rental companies’ insights into new potential solutions and services to better equip the DIY mover to do what they do even easier.

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 the customers.

## 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 below shows the major architectural structure of the solution. In the diagram of Figure 1, movers interact with the smartphone client which provides them with an interface to access important features such as the Load Plan, Move Inventory, and Logistics Planning. The Load.In smartphone client interacts with hardware in the smartphone to be able to take the necessary photos to generate the 3d models.

Figure   
*Major Functional Component Diagram*



Since the smartphone client and website must contain the same data, all data for Load.In resides in a central database. Amazon’s RDS, or Relational Database System, provides the hosting platform that the database software will run on. The target database software that the client application and website interact with is MySQL.

When one considers a typical dwelling, one’s dwelling involves typically more than one room. When packing for a move, it would not make sense then for a mover to take his/her belongings into another room of the house where the one computer is sitting at a desk to be able to catalog his/her moving inventory. Therefore, portability of the solution is paramount to the success of the operation because the mover should be able to catalog his/her inventory wherever he/she is currently at that time. This is what makes the use of a smartphone client the ideal choice when considering the platform to host the primary functions of a move and will be the logical choice for the DIY mover. Another important consideration when it comes to the smartphone is the available hardware. The Load.In application by necessity, needs access to a camera for the move inventory and when loading a truck, 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 would also need 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 cut down on transmission and the storage requirements of the photos. The PNG file format provides a fair degree of 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. The Load.In software development team wrote the web API in the Java programming language and utilized the Apache CFX web API framework. It 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 Bean Stalk 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.

# 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 square feet (Andrew P., 2020). Therefore, with this house size in consideration and the consideration that this family will have established furniture and other household furnishings over several years to accommodate three family members, 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 require 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 move for a family would be for a family to move within 20 miles of their home, so for this case study, they too will be moving across town only.

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.

# 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 cutting down on the effort involved to create it. The Load.In prototype includes some key features such as the Load Plan, the Move Inventory, the Logistics Planning, and the Expert Tips. Table 1 shows a comparison of features from the RWP, also known as real-world-product, versus the prototype.

Table   
Real World Product versus Prototype

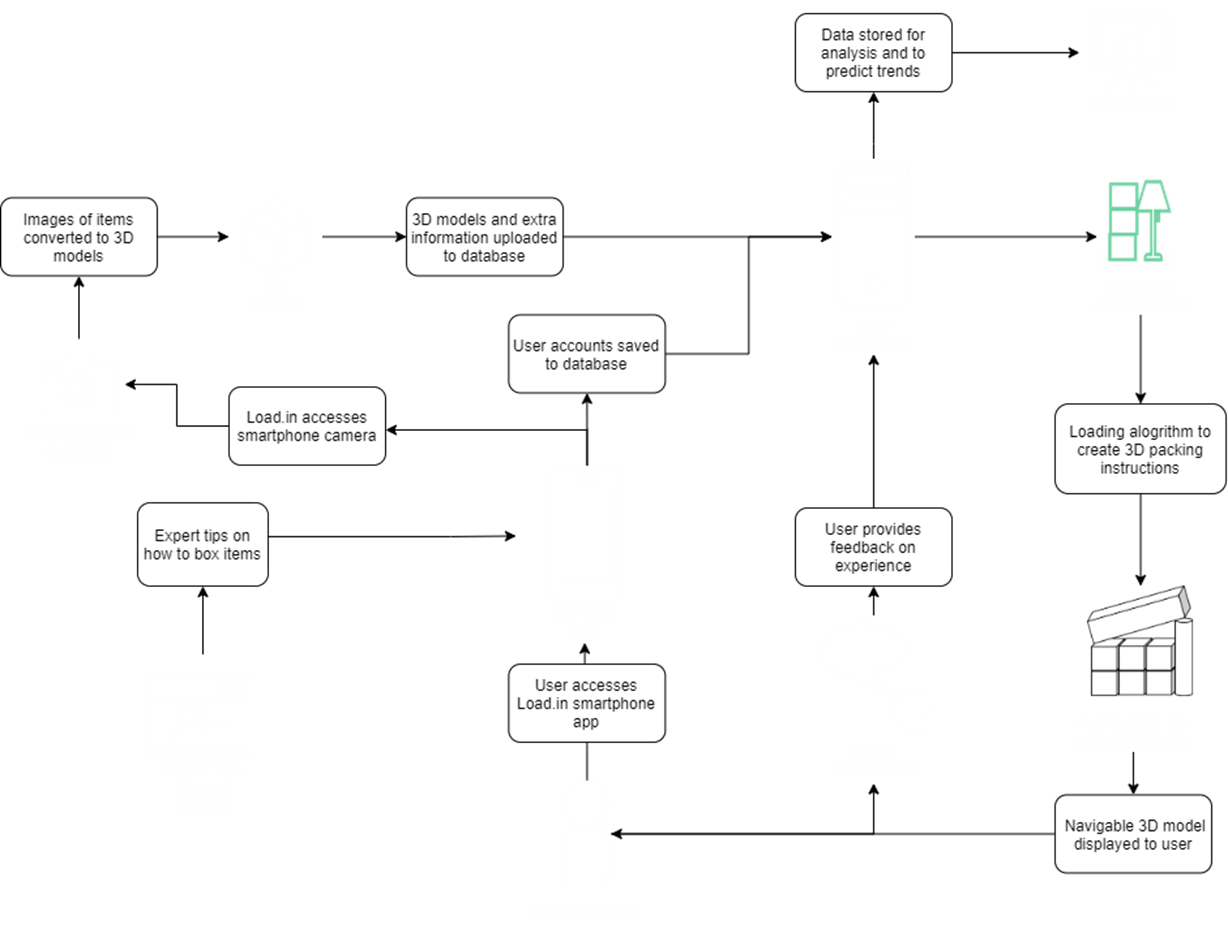
|  |  |  |
| --- | --- | --- |
| Feature | Real World Product | Prototype |
| Move Inventory | | |
| Furniture/Item measurement | Fully Functional | Partial |
| 3D model generation | Fully Functional | Partial |
| Item weight | Fully Functional | Eliminated |
| Item fragility | Fully Functional | Eliminated |
| 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 |

Because there is no actual move for the purposes of the demonstration of the prototype, certain aspects of the prototype must be simulated or modeled to show some base cases which may be applicable to an actual move. For example, the entering of one item which is non-standard using computer vision might simulated the concept of entering more than one item into the solution for a real move, and the entering of one box contents into a move inventory might serve as a simulation as to entering multiple boxes into an inventory. The truck sizes will have to be simulated to represent real trucks and the test harness will have to control the truck sizes, box sizes, and move inventory to some extent to make the simulation valid for simulating a real move while at the same time reducing the amount of time it takes to demonstrate the key features of the application.

## 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 run out of 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 the containers run on. The prototype’s main components will consist of the Android client app, the web API for brokering the communication to the database, the database hosted on mySQL, and a test harness running out of 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   
Load.In - Prototype Major Functional Component Diagram



## 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 on the fly. 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 an item, entering in a box, 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, which are what makes Load.In innovative because there are no other solutions that can do what Load.In can.

There are some risks that the prototype will have in and of itself which it must address for a successful launch. There are two main risks from a customer perspective that the Load.In prototype must address. First risk that the prototype faces is the risk that the mover may not like the recommendations of the application. Some examples of how this might manifest: the mover may not like how the Load Plan recommends the mover should load the truck or the mover dislikes the Expert Help which recommends the wrong way to pack an item. The implementation of the feedback feature mitigates this risk, which 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 any number of steps so that the display replays steps can in the correct order and the mover can make sure they have followed the instructions appropriately.

## Prototype Development Challenges

As with any prototype, there will be challenges related to the development of Load.In’s prototype. One of the biggest challenges which awaits the team ahead is the development of the test harness and the implementation of some functions or routines that will allow the team to change certain values from the prototype’s database such that the client Android app reflects the changes. One of these scenarios that need to be demonstrated is the scenario of what happens when the available trucks change on a mover after a mover has selected his/her 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 that is ahead is the challenge of 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. Other challenges ahead will be keeping track of where boxes currently are at any given time, for move inventory searches to reflect that, dealing with multiple trips, prioritizing certain expert tips over others in response to a search for tips, and dealing with complicated edge cases, like a truck that is too small or a move that has many fragile items.

# Glossary

**3D** – Three Dimensional

**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 lets you run code without provisioning or managing servers.

**Amazon RDS** - Amazon relational database service

**Amazon Web Services (AWS) –** 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.

**Application Programming Interface (API)** – An interface for programs to share information and functionality with one another through a series of call 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.

**CPU** – Central processing unit.

**CSS** – Cascading style sheet.

**Do-It-Yourself(DIY)Mover** – Non-professional movers that rent a truck for their move, but and handle all packing, unpacking, 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

**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

**Java** - a set of computer software and specifications developed by James Gosling at Sun Microsystems, which was later acquired by the Oracle Corporation, 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 function 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 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.

**Operating System (OS)** – 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.

**Portable Network Graphics (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.

**Real World Product (RWP)** – The actual Load.In solution as it was intended.

**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 which rents moving vehicles for a Do-It-Yourself Mover to assist them with their move.

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

# References

29+ Moving Industry Statistics [Updated 2020]. (2020, November 04). <https://www.movebuddha.com/blog/moving-industry-statistics/>

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. <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? <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. <https://www.moving.com/tips/loading-truck-rental/>

Wood, T. (2020, January 6). Moving industry statistics. *MoveBuddha*. <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. <https://themortgagereports.com/49116/report-where-americans-are-moving-and-how-far-theyre-going>