**Load.In - Product Description**

Jason Moran

Old Dominion University

CS410

Janet Brunelle

02/06/2021

Version 2

**Table of Contents**

[1 Product Description 5](#_Toc63526568)

[1.1 Key Product Features and Capabilities 5](#_Toc63526569)

[1.2 Major Components (Hardware/Software) 8](#_Toc63526570)

[2 Identification of Case Study 11](#_Toc63526571)

[3 Load.In Prototype Description 12](#_Toc63526572)

[4 Glossary 15](#_Toc63526573)

[5 References 19](#_Toc63526574)

**List of Figures**

[*Figure 1*. Comparison of Cost of Do-It-Yourself Moving and Hiring Professional Movers. 3](file:///C:\Users\Jason\Desktop\Misc\JMORA_Lab1v2.docx#_Toc63526582)

[*Figure 2*. The Major Functional Components of Load.In. 8](#_Toc63526583)

[*Figure 3*. The Minimum System Requirements for the Load.In Smartphone Client 9](#_Toc63526584)

[*Figure 4*. The Minimum System Requirements for the Load.In Website Client 10](#_Toc63526585)

[*Figure 5*. A Comparison of Features Between the Real World Product and Prototype 12](#_Toc63526586)

[*Figure 6*. The Major Functional Components of the Load.In Prototype. 13](#_Toc63526587)

Introduction

Moving is a stressful life event that presents Do-It-Yourself movers with a variety of challenges. These challenges include logistics planning, effective space usage in their moving truck, packing, and managing moving costs: all activities that moving families are unlikely to undertake frequently. Making effective use of a moving truck is difficult in its own right: there are many weight distribution considerations one must make on top of trying to maximize space usage – not to mention keeping track of the location of every box (White, 2020). All the while, Do-It-Yourself Movers must navigate the business model of the companies they rent trucks from and often encounter hidden fees that unexpectedly increase their cost (Manwaring, 2020).

Cost is a major factor that prevents families from hiring professional movers. On average, hiring professional movers for a short distance move is approximately seven times more expensive than only renting a moving truck (Manwaring, 2020; Meyers 2018).

*Figure 1*. Comparison of Cost of Do-It-Yourself Moving and Hiring Professional Movers.

Truck Rental Cost data from Manwaring (2020). Professional Mover Cost data from Meyers (2018).

As a result of this huge price disparity, only 21% of moves utilize the help of professional movers – leaving a large majority of movers to fend for themselves when it comes to managing the logistics of their move. In 2019, 31 million people, nearly 10% of the population, moved residences in the United States of America (“*29+ Moving Industry Statistics,”* 2020). This fact, when taken into consideration with the fact that most movers do it themselves, reveals that millions of Americans go without the expertise and skills of professional movers every year.

Load.In is an Android app that utilizes Artificial Intelligence and Computer Vision to simulate the expert knowledge of professional movers at a price attainable to Do-It-Yourself movers. Load.In is an Android application that uses Computer Vision and Artificial Intelligence to enable Do-It-Yourself movers to perform their move efficiently and effectively. Load.In serves the Do-It-Yourself moving market, a majority of the moving industry, by simulating the expert knowledge of Professional Movers for at an accessible cost.

# Product Description

Load.In has multiple features that assist the Do-It-Yourself mover throughout the move process. Most notably, Load.In generates customized plans for truck loading using a computer vision algorithm and 3D models created from images taken by a smartphone camera through a process called photogrammetry. This Load Plan balances efficient space usage, weight distribution, protection of fragile items, and unloading priority with minimal user input. Users have the ability to view the load plan throughout their move – giving them insight as to which boxes are already unloaded, on the truck, or have yet to be loaded. As a side effect of the load plan generation algorithm, users are given unique insights in to how many trips a move will take with a specific truck. These insights provide Do-It-Yourself movers the unique ability to minimize either their cost or the time spent in the moving process. Without Load.In, only rough estimates are used to determine truck size and hidden fees can drastically change the final cost of a move, making time or cost optimizations difficult. Finally, Load.In provides expert tips and tricks in the packing process – serving as an information hub and resource to Do-It-Yourself movers throughout every step of the moving process.

This section will provide an overview of the features and capabilities as well as the major hardware and software components of Load.In.

## Key Product Features and Capabilities

The most notable feature of Load.In is its Load Plan generation. The Load Plan is made possible through photogrammetry, a technique used to take three dimensional measurements from two dimensional images. Through this technique, accurate 3D models of boxes, furniture, and other household items can be obtained with just a smartphone camera. These 3D models are used to simulate the truck loading process through a modified Container Loading Algorithm. This algorithm calculates the most effective space usage, giving each user a completely customized set of automatically generated instructions to most effectively utilize their moving truck. In this process, approximate weight, fragility, and unloading priority are also considered.

This load plan is not useful only for helping Do-It-Yourself movers load their truck, but also in giving them logistical insights throughout the move. By keeping a digital record of which boxes are loaded on to the truck, Load.In can keep track of which boxes are at the destination, at the starting point, or are currently loaded on the truck. If a move takes multiple trips, Load.In generates a load plan each time the truck is re-loaded. This enables Load.In to present the user with a timeline showing which boxes were loaded in which trip. This feature gives Load.In users the ability to approximately locate a box through the app, giving them peace of mind with minimal effort.

Another useful aspect of the Load Plan is that it enables accurate rental costs estimation. Truck rental prices are based on time and mileage. This means that the total cost of a move is unclear until after a move has taken place, which can leave families renting trucks with a bill higher than they expected. With Load.In’s move plans, an accurate estimate of how many trips a move will take can be obtained. This information, combined with the travel distance between the origin and destination and truck rental fees, can be used to accurately predict truck rental costs before the move, allowing users to minimize cost, move time, or if they are lucky, both. The Rental Estimate feature is supported by a web scraper that collects data from rental truck company websites in order to provide up to date information for the cost estimate. Other information, such as local gas costs, moving supply costs, and average loading and unloading times are be obtained from various sources and used in the estimation.

Separate from Load.In’s Load Plan generation and its related features, Load.In also offers a variety of tips and tricks for the packing process. These tips and tricks are specifically written for Load.In users by expert packers and can be accessed easily through a search feature inside the Load.In app. This feature works synergistically with the Computer Vision driven Load Plan, both features work together to impart expert knowledge upon Load.In users.

As a companion to the Tips and Tricks feature, Load.In also features a chat option where users can directly ask questions. These questions are first be received by an AI chat-bot. However, a human operator is also be available if the AI is unable to help the user. The goals of this feature are to guide the user to the tips most relevant to them and to serve as an information collection point in order to improve the app over time.

The AI Chatbot is not the only feedback mechanism within Load.In. Load.In features usage analytics in order to understand how the app is being used in practice and, most importantly, when users abandon the app for unsuccessful moves. In addition, the app features feedback screens where users can specifically call out problematic aspects of the app’s performance. Combined, this feedback is incorporated into an iterative development process to improve Load.In over time.

## Major Components (Hardware/Software)

The major technical components of Load.In are its backend database, smartphone client, website client, Web API, Web Application, and Vendor Synchronization system. This section outlines the technical details of each. Figure 2 provides a visual overview of the working components and their interactions.

Diagram

Description automatically generated

*Figure 2*. The Major Functional Components of Load.In.

Load.In’s database back-end is maintained in the Amazon Web Services cloud service as a MySQL powered Relational Data Service (RDS). This database system is used to store a variety of data; from user account data to usage data to 3D model data used in Load Plan generation.

The end-user of Load.In primarily interacts with the Smartphone Client, which serves as the primary point of data input. In order for Load.In to function properly, there are certain requirements to ensure that the system can function properly. These requirements can be viewed in Figure 3.

|  |  |
| --- | --- |
| Specification | Estimated Targeted Value |
| Operating System | Android 4.4 (KitKat) |
| RAM | **4GB RAM** |
| CPU | **8 Core @ 1.8Ghz per core** |
| Storage | At least 2 GB internal storage for cache for images and rendered model data  1GB reserved cache for 3D Models  Target is 100 cached photos  100 \* 55.84 Mb per photo = 5,584 Mb ~ 698 MB |
| Cellular Connectivity | **4G Cellular up to 15 Mbps** |
| Wireless | 2.4 GHZ @ Wireless N with min 150 Mbps |
| Internet Connectivity | 30 Mbps |
| Camera | Single camera lens @ 12 Megapixel |
| Photo Size | 4290x2800 (~12MP)  PNG Compression Approx. 6.98 MB per photo or  55.84 Mb per photo |
| Photo Transmission Time | **< 3 Second @ 30 Mbps Internet Speed**  **< 6 Seconds @ 15 Mbps Cellular** |

*Figure 3*. The Minimum System Requirements for the Load.In Smartphone Client

The Load.In ecosystem also features an internet dashboard that collates usage data into a dashboard of moving trends and other insights into the moving industry. Viewing this dashboard requires a computer with an internet connection, which imposes its own set of hardware requirements. These requirements can be viewed in the table below.

|  |  |
| --- | --- |
| Specification | Estimated Targeted Value |
| Operating System | Linux, Windows, Mac OS    **Any operating system that supports current browsers** |
| Browser | Edge, Chrome, Firefox, Safari  Browsers must support ES6 or above and HTML 5    **Firefox 78 and Above**  **Chrome 84 and Above**  **Edge 83 and Above**  **Safari 13.4 and Above** |
| RAM | 4GB RAM |
| CPU | 2 cores @ 2GHZ |
| Storage | 1 GB Cache for images and website content |
| Wireless | 2.4 GHZ @ Wireless N with min 150 Mbps |
| Internet Connectivity | **30 Mbps** |

*Figure 4*. The Minimum System Requirements for the Load.In Website Client

The Mobile and Website Clients are both served by a Web API which is programmed in Java and served on Amazon Elastic Beanstalk using the Apache CFX framework. These technologies used in conjunction allow for a maintainable and scalable services infrastructure.

Finally, Load.In contains a Vendor Synchronization system which is used to obtain data regarding truck rental prices, fees, and other information. This system is written in Java and uses Amazon Web Services Lambda to scrape data on a predetermined schedule.

# Identification of Case Study

Load.In is tailored for the average Do-It-Yourself move. 53% of moves occur over a distance less than 50 miles (Yale, 2019), so this case study looks at 50 mile moves where travel time does not prohibit the possibility of making multiple trips throughout the move. The average house size in the United States is 2,200 square feet (Andrew P., 2020), and the average family size is approximately 3 people (US Census Bureau, 2019). It is assumed that a move of this size requires the usage of a moving truck, as most personal vehicles do not have the capacity to efficiently move a house of this size. Most moves occur without the assistance of professional movers, so it is assumed that this family wants to minimize cost. Friends or other family members may be called upon to help in the moving process. It is assumed that the family does not have any exceptionally un-ordinary items that need to be packed or loaded.

In order to gauge the effectiveness of Load.In, moves using Load.In and not using Load.In will be observed for this case study. Truck loading time, unloading time, and number of trips will be measured in order to quantify the time savings that Load.In provides. In addition, qualitative data regarding the moving families’ experience moving with and without Load.In will be collected.

In the future, Load.In may be adapted for extremely small moves, such as loading the back of a pickup truck or even extremely large moves – assisting the professional movers Load.In seeks to emulate today. As time goes on, Load.In will need to be adapted to support packing and loading of unusual items as the need arises. Overall, Load.In’s technology has potential for productization wherever space optimization is necessary.

# Load.In Prototype Description

A prototype version of Load.In with limited capabilities will be developed in order to prove the viability of the product. This prototype version will feature limited photogrammetry and load plan generation capabilities due to the complexity of those features and will not support numerous aspects such as analytics, truck rental company integrations, and the live chatbot. Figure 5 compares the features of the Real World Product and Prototype.

|  |  |  |
| --- | --- | --- |
| Feature | Real World Product | Prototype |
| Move Inventory: | | | | |
| **Furniture/Item measurement** | Fully Functional | Partial |
| **Photogrammetry** | Fully Functional | Partial |
| **Item Weight** | Fully Functional | Eliminated |
| **Item Fragility** | Fully Functional | Eliminated |
| **Box Locator** | 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 Move Time** | Fully Functional | Eliminated |
| **Estimated Truck Costs** | Fully Functional | Eliminated |
| Expert Help: | | | |
| **Packing Tips and Tricks** | Fully Functional | Fully Functional |
| **Search Feature** | Fully Functional | Fully Functional |
| **Expert Articles** | Fully Functional | Fully Functional |
| **Chatbot** | Fully Functional | Eliminated |
| **Live Expert** | Fully Functional | Eliminated |
| Vendor Integration: | | | |
| **3rd Party Vendor Web Scraper** | Fully Functional | Eliminated |
| **3rd Party Vendor Web API Reader** | Fully Functional | Eliminated |
| **Box Dimension Finder** | Fully Functional | Eliminated |
| **Truck Size Finder** | Fully Functional | Eliminated |
| **Truck Availability Finder** | Fully Functional | Eliminated |
| Analytics: | | | |
| **Location Data** | Fully Functional | Eliminated |
| **Move Data** | Fully Functional | Eliminated |
| **Feedback Data** | Fully Functional | Partial |
| **Usage Heatmap** | Fully Functional | Eliminated |
| **Rental Statistics** | Fully Functional | Eliminated |

*Figure 5*. A Comparison of Features Between the Real World Product and Prototype

The Load.In prototype will be produced using an Ubuntu 16.04 Virtual Machine development environment. At a surface level, the user will interact with an Android App GUI. This GUI will also contain test harness elements which are used for testing and demonstration purposes. The Load.In database will be contained in a docker container and accessed using MySQL. A second docker container will contain the Load.In Web API which will utilize Apache Tomcat as a platform. Figure 6 shows the major functional components of the Load.In prototype.

A picture containing meter, parking, display, phone

Description automatically generated

*Figure 6*. The Major Functional Components of the Load.In Prototype.

The Load.In prototype will demonstrate a minimalistic version of the Real-World Product workflow. While the Real-World Product will create and utilize 3D representations of any object, the Prototype will only support measuring boxes. The Prototype will have the ability to view a Move Inventory, which will provide information about each box entered into the app, including which trip the box was moved on. The Prototype will also feature a rudimentary version of the Load Plan, which will provide step by step instructions for truck loading. Because of the information generated by the Load Plan and Move Inventory, the number of trips required to finish a move will be shown in the Prototype as well. In addition, packing tips and expert articles will be available in the Prototype version. Finally, a basic feedback feature will be included in the Prototype.

Together, these features represent the core workflow of Load.In. While the Prototype version will not be as fully featured or sophisticated as the Real World Product, it will still serve as a valid proof of concept for the unique combination of features that make Load.In an innovative and unique offering in the market.

In moving from the Prototype to Real World Product stages of development, one of the greatest risks Load.In faces is not properly collecting and utilizing user feedback. The Load.In prototype will feature minimal feedback mechanisms to ensure that valuable feedback is not lost. The second major risk with Load.In concerns the Load Plan. If a user chooses not to follow the instructions of the Load Plan, it ceases to be valuable to the user. To mitigate this risk with minimal effort, the Load.In Prototype will include a feature to rewind the Load Plan to correct any packing mistakes. In order to mitigate any security risks involving personal data, the Load.In Prototype will store all data locally on the user’s phone, rather than store it on offsite servers, and will additionally provide an option to delete the data.

In summary, the Load.In Prototype will serve as a proof of concept for the Real-World Product. In doing so, it will implement basic versions of the Move Inventory, and Load Plan features. It will also contain expert tips and the necessary search features to find them inside the app. The Test Harness will include tools to adjust truck dimensions and create randomized lists of box sizes for testing and demonstration purposes.

Several challenges have been anticipated for the prototype stages of development. The ability to adjust critical data such as truck size and move inventory from the test harness is necessary but may prove difficult. Calculating the number of trips is possible within the scope of the Prototype but will require careful design of Load.In’s internal systems to accomplish. Creating a feature that allows the user to search for tips and tricks is possible at a basic level but optimizing that search to be as useful as possible may be challenging. Finally, because Load.In is an application that depends heavily on computer vision algorithms, there may be several edge cases in those algorithms that need to be considered, and each may require unique solutions, complicating the development process.

# 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 work stations.

**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 work stations. 

**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.

**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 work stations.

# 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>