Load.In Product Specification

Jason Moran

Old Dominion University

CS411W

Janet Brunelle

3/28/2021

Version 1

Table of Contents

List of Figures

1 Introduction

Relocating to a new home is a stressful life event that presents Do-It-Yourself movers with a variety of challenges. These challenges include logistics planning, packing, dealing with fragile or heavy items, managing moving costs, enduring long-distance travel, time management and efficiently loading the moving truck. When loading a moving truck there are many weight distribution considerations one must make in addition to efficiently fill the and keeping track of the location of every box (White, 2020). Do-It-Yourself Movers must also 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 renting a moving truck (Manwaring, 2020; Meyers 2018). Figure 1 illustrates that the cost difference between Do-It-Yourself and Professional moving grows with the size of the move.

As a result of this huge price disparity, only 21% of moves utilize the help of professional movers (Wood, 2020) – 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 (Wood, 2020). This fact, when taken into consideration with the fact that most movers do it themselves, reveals that millions of Americans struggle without the expertise and skills of professional movers every year.

1.1 Purpose

Load.In is an Android application 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 enables Do-It-Yourself movers to perform their move efficiently and effectively by providing the expert knowledge of Professional Movers at a low cost.

Load.In has multiple features that assist the Do-It-Yourself mover through the move process, including packing, finding a moving truck, effectively loading the moving truck, and finding boxes once they have been moved. Most notably, Load.In generates plans for truck loading. These plans are generated using a Computer Vision algorithm and 3D models created from images taken by a smartphone camera through a process called Photogrammetry. This algorithm balances efficient space usage, weight distribution, protection of fragile items, and unloading priority. Once generated, the Load Plan can be reviewed throughout the move – providing insight as to which boxes are already unloaded, on the truck, or have yet to be loaded. As a result of the Load Plan generation algorithm, users are given information on the number of trips a move will take with a specific truck size. These insights provide Do-It-Yourself movers the unique ability to minimize either their cost or the time spent in the moving process without sacrificing safety or efficiency. Without Load.In, only rough estimates are used to determine truck size and hidden fees can drastically change the final cost of relocating – making the final cost unclear. Load.In provides expert tips and tricks in the packing process – serving as an information hub and resource to Do-It-Yourself movers throughout the moving process.

1.2 Scope

A prototype version of Load.In with limited capabilities have been developed in order to prove the feasibility of the innovative aspects of the solution. This prototype version contains a simplified set of features, with some real world product features cut entirely, and has also been produced with a smaller architecture.

The Load.In prototype demonstrates a simplified version of the real world product. While the real world product creates and utilizes 3D representations of any object, the prototype only support boxes. The prototype has the ability to view a Move Inventory which provides information about each box input into the application. The prototype also features a simplified Load Plan that does not support weight, fragility or unloading priority. Because of the information generated by the Load Plan and Move Inventory, the number of trips required to finish a move is available in the prototype. In addition, packing tips and expert articles are available in the prototype version. A basic feedback feature is included in the prototype.

Together, these features represent the core workflow of Load.In. While the prototype version is not as fully featured or sophisticated as the real world product, it still serves as a valid proof of concept for the unique combination of features that make Load.In an innovative and unique offering in the market.

Figure 1 illustrates the scope of the Load.In prototype in comparison with the features of the real world product.

Figure 1

A Comparison of the Prototype and Real World Product Feature Sets

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

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 models 50 mile moves where travel time does not prohibit multiple trips during 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). In addition, this case study only models moves that utilize a moving truck and do not employ professional movers. When studying the efficiency of a move for the case study, the number of participants in the move is considered, while extremely unusual items may be disregarded when judging the effectiveness of Load.In.

In order to gauge the effectiveness of Load.In, moves matching the description above are simulated with and without the use of Load.In. 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 through a series of interviews.

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 the product matures, Load.In needs to be adapted to support packing and loading of unusual items. Overall, Load.In's technology has potential for productization wherever space optimization is necessary.

1.3 Definitions, Acronyms, and Abbreviations

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.

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 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, 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 – TODO FINISH FILLING OUT

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 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.

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

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

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.

1.4 References

Moran, J. (2020, March 28). *Load.In Product Description Version 3.* Old Dominion University.

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_aUTUps&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

1.5 Overview

The remainder of this document will cover the inner workings of the Load.In prototype, including the architecture and tools used to create the prototype, the functional components of the prototype, and the prototype's external interfaces. In addition, the specifications describing the functionality of the prototype, including functional requirements, performance requirements, non-functional requirements, assumptions, and constraints will also be covered.

2 General Description

2.1 Prototype Architecture Description

The Load.In prototype was produced using an Ubuntu 16.04 Virtual Machine development environment. The user interacts with an Android App GUI. This GUI also contains Test Harness elements which are used for testing and demonstration purposes. The Load.In database is contained in a Docker container and accessed using MySQL. A second Docker container stores the Load.In Web API which utilizes Apache Tomcat as a platform.

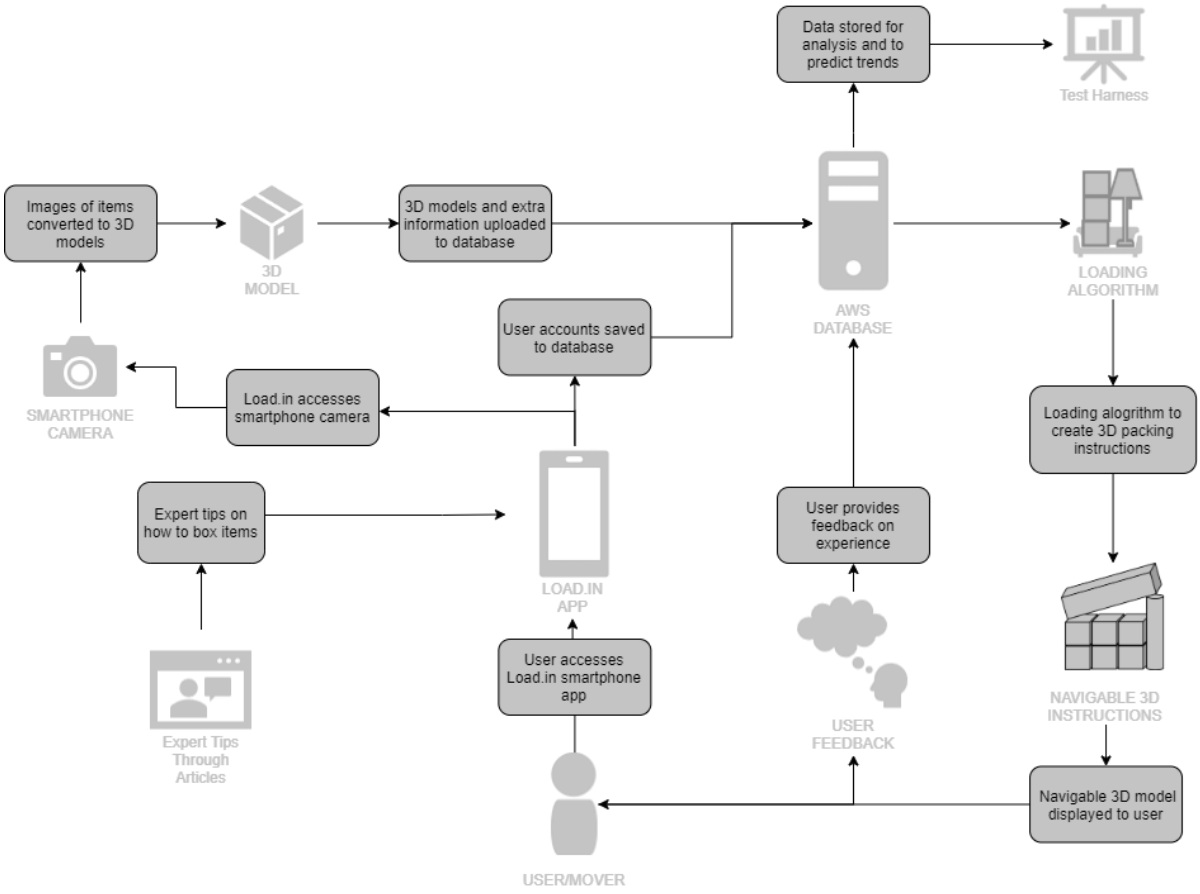
2.2 Prototype Functional Description

The Load.In prototype has several functional components. The interactions between each component are explored in detail in Figure 2.

- The Smartphone Client: This is the user interface that the user interacts with on their android device, with which they input data into the system and view the results of Load.In's various algorithms
- Web API: Used by the smartphone client to send and receive data from the database.
- Database: Used internally to store user data as well as data of the user's move inventory and their feedback.

Figure 2

The Major Functional Components of the Load.In Prototype



2.3 Prototype Development Challenges

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. Most importantly, 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 and lengthening the development process.

2.4 External Interfaces

Load.In requires certain external interfaces in order to function properly. Load.In requires an Android device with a built-in camera in order to provide input to it's photogrammetry enabled features. In addition, the device Load.In is running on must be able to communicate via the internet using TCP/IP, DHCP and DNS connections. For the prototype, certain external communications, such as Rental Truck Vendor Synchronization, will be simulated on the device for simplicity.