Lab 1 – Load.In Product Description

Chris Miller

Old Dominion University

CS 410

Professor Janet Brunelle

11 December 2020

Version 1

Table of Contents

Tables and Figures

1. Introduction

The average American moves almost 12 times in their lifetime. In 2019 nearly 31 million Americans moved, which is nearly 10 percent of all Americans moving every year. Moving is expensive; the average cost of an interstate move is $4,100 and an in-state move is $980. This average cost includes both do it yourself (DIY) and professional movers. For many, this means that hiring professional movers is just not feasible because of the high cost (Wood, 2020). Moving comes with many difficulties in addition to just the cost—from the packing of boxes and loading them onto the truck to the potential loss or misplacement of important items when a move is finished. The final and most valuable cost to movers is time. DIY moves take more time than hiring professionals.

All these problems culminate in one large issue for Americans: moving is either extremely expensive or extremely time consuming and tedious. To help solve these issues, an application for planning a DIY move is a solution that cuts costs and saves time for the mover. Load.In looks to solve these problems by providing a platform for movers to plan, catalogue, and gain expert knowledge on how to move more efficiently. Load.In allows the user to take pictures of their boxes and through photogrammetry produce 3D models of those items, access expert tips and tricks on how to pack and load different items, and provides a step-by-step packing solution to guide the mover through every step of the process—all without ever worrying about where any given item has gone.

2. Load.In Product Description

Load.In is a mobile application that can be used by any person or group of people to plan a move. Load.In aims to help reduce the cost of a user's move as well as decrease the time and difficulties that comes with a move. By taking pictures of their items, users have a 3D generated

plan of both loading and unloading their vehicle. This plan also contains suggestions on what rental truck sizes are the most efficient for moving the user's items. This process begins with the ability to access expert tips on how to pack items into boxes. Whether a user needs help with packing fine china or a ceramic vase, Load.In has different tips and tricks on how to begin packing for a move. The next step in the process is for the user to take pictures of their items to be packed. This allows Load.In to generate 3D models of the user's items and create a plan on how to load them based on the size, shape, and fragility of the user's different items. Another key component is that because all the boxes are being catalogued, a user will never have to worry about where important items are. Load.In maintains an inventory of what box is located where inside the truck including the contents contained within said box.

Regarding rental truck companies, Load.In acts as a powerful ally in analytics gathering and partnership to provide deals and information to the customers. By tracking data such as truck sizes and move distances, Load.In can provide rental truck companies with data on how large a fleet should be at a given location as well as how many of each truck size is needed and when. In addition to analytics gathering, Load.In can provide a platform for these companies to advertise directly to users who need their service. When the user's load plan is generated, the user can see an advertisement for the local rental company that has the exact truck size the user needs.

Load.In is an all-in-one solution to a DIY move. Keeping track of items, generating a plan to load and unload the truck, and providing tips on how to pack makes the lives of the users so much simpler. By cutting out the expensive costs of professional movers, Load.In provides a middle ground where DIY movers save both time and money that would otherwise be lost from having professional movers or planning the move completely by themselves.

2.1 Key Product Features and Capabilities

Load.In provides several different capabilities for users to simplify their move. The biggest feature is the 3D load plan generated on the app. This works through photogrammetry—accessing a user's smartphone camera and taking pictures of the items to be packed. With each picture the app determines whether an accurate model can be generated, and if not, it has the user take another picture. Once all the items are photographed and modeled the algorithm takes these 3D models and prepare the loading plan. This algorithm considers the fragility of items as well as weight and size to determine the safest and most effective way to pack the items. This generally means that things like furniture are packed first and other items stacked on top of them. The load plan suggests a truck size and points the user to where such a truck can be rented. If the user specifies a truck size or chooses a different truck, the plan is regenerated for that new vehicle. The load plan is meant to be a step-by-step manual on how to load and unload the truck. It specifies a location and order of how to pack items to keep them safe and make the job faster.

With the user taking pictures of all their items, Load.In maintains a log of all the items. This means that the user will never have to worry about which box has which items. The utensils the user packed are labeled as designated by the user—meaning they can take that box straight to the kitchen in their new home. This inventory means that not only is there no worry about misplacing items, but the important items that you may need right away, such as a computer, will be easy to find and unpack. In addition to the log of items, Load.In has a cost estimation feature generated from the information the user provides. By taking the load plan, the truck size, and start and end locations of the move, the app calculates an estimated total cost for the move. Load.In works with rental truck vendors to provide the best estimates and deals to its users. This is done through scraping the web and pulling in data from the rental companies to gather truck

sizes, costs, and availability. With all of these factors, Load.In shows a tangible cost benefit to the user over hiring professional movers.

In order to help a user begin boxing their items, Load.In will provides expert tips and have a live chat. These two services help with packing difficult and fragile items to keep them safe. The chatbot and live help are used to either guide a user to articles about the topics they want help with or provide live service help to the user. This could be for questions as to how the app itself works or something as simple as removing the legs of a couch. The tips and chatbot are there to improve the user experience and help answer questions when they arise without having to wait long periods for live customer service.

Load.In uses all this data that from the users to generate analytics on moves throughout the country. The important pieces of information that Load.In gathers include location data, meaning a move's start and end locations, and the total amount of trips and time it took a user to complete that move. Data on the move itself is collected including the cost of gas, the cost of renting the truck, the cost of any other supplies such as boxes, as well as the time spent loading and unloading the truck. The specifics on a user's inventory are kept until the move is completed, and the less personal data will continue to be stored. Data such as how many fragile items were packed, the weight of the entire move, and the dimensions of the different items or boxes are stored. All this data will be used to generate predictive analytics about the overarching meaning of these statistics. Generating predictions on where people are moving from and to, how far they are moving, and how much they are moving are valuable assets that can be either sold or used to make a more efficient service to Load.In users.

The final functions of Load.In are to gather feedback from the user. This is rating their experience with Load.In and how it helped their move. Giving feedback on how a move was

helps the product be improved for future users. Another way in which Load.In works to improve is using a heat map to track what users are accessing the most. It tracks clicks on the app as well as when the app is closed so Load.In knows exactly what features are being used the most or the least and what needs to be improved upon.

2.2 Major Components (Hardware/Software)

Figure 1 is an illustration of the Load.In Major Functional Component Diagram. This illustration is representative of how the Load.In application functions, with each user using a mobile device which displays a GUI to interact with specific functions. Each function in turn communicates with a remote AWS Database and produce an output which is presented to the user. Most of these functions are accessible through the Load.In app; however, some are reserved for admins to access things like analytical data on the website rather than the app itself.

The hardware that is required for Load.In to function starts with a relational database known as Amazon RDS. This is compatible with MySQL and Amazon Lambda which is used for connections on the back end of the system. The software is composed of a back-end API, user authentication, data storage, and algorithms in order to get the Load.In app to function for each user. The data storage and algorithms are a large part of the components as there is both a large amount of data to be collected as well as many different algorithms being used to generate all of the information Load.In hopes to provide to the users.

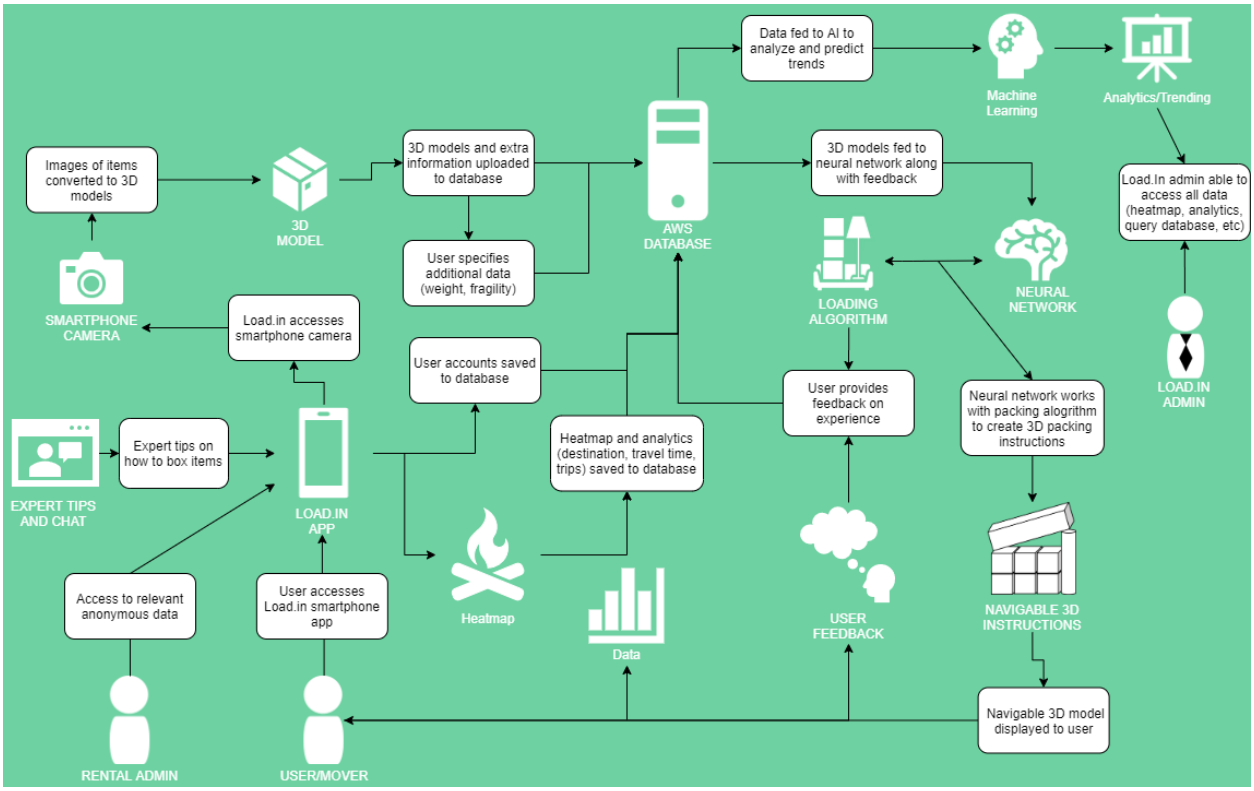

Figure 1, Load.In Major Functional Components Diagram. This figure shows the functional components of the Load.In application.

Load.In is as an application on Android smartphone devices ensuring the device allows access to the smartphone's camera. Many of the important hardware requirements are covered by most Android smartphones. The goal is to do as much of the job locally on a user's smartphone as possible. This means storing the photos locally—requiring more memory from the user. The other components needed to run Load.In includes a stable internet connection, either through Wi-Fi connectivity or cellular.

The Load.In website client is needed for applications outside of the specific move plan solution that is intended for the customer. The website is designed to work on current browsers and require very little in terms of the hardware on a computer as it is an information site and an analytical display for admins. The web API is written in Java and takes advantage of both AWS Elastic Beanstalk and Spring MVC with Tomcat. The vendor synchronization is also written in

Java and take advantage of AWS Lambda. The goal is to have the vendor synchronization work on a timed schedule so that it brings data from vendor sites in a timely manner.

The most important part of the Load.In system is the database. This runs on the cloud and needs to store large amounts of data including user profiles, analytics on the moves of the user, rental company information, and cost estimates. The database is the backbone of Load.In; much of the functionality and value comes from the database running smoothly.

3. Identification of Case Study

Load.In is being developed for people moving all over the country. Whether it is a move just across the street or a move 3,000 miles across the country, the goal is for the app to be useful in either situation. The case study for Load.In will be to look at a single family of 3 with a dog moving less than 20 miles across town. They need to move the contents of an approximately 2,200 square feet household which is the average house size US Census Bureau (2019). This move will also need a rental vehicle as it is too large just to fit in the family vehicles. A few of the mandatory items involved in the move will be furniture to get a more accurate picture of a typical move. Load.In will need to accurately tell the costs of this move and allow friends or family to collaborate through the app on the move. Really tricky or odd shaped items will be left out of this case study— meaning that the expert tips will not be necessary for determining the effectiveness of the app. The final goal for the Load.In case study is to have the app work well for a common local move but be able to see that it can both scale up and scale down to larger moves—whether in terms of distance or in terms of the number of items being moved.

4. Product Prototype Discussion

The prototype for Load.In will have some reduced functionality from the overall vision of the application. The features that are being developed for the prototype are as a proof of concept and are meant to help show what the app can potentially do. This means changing functionality such as the expert tips so that it merely shows the application can produce these tips rather than the implementation of real and valuable tips for a DIY mover. This model will be used to show how Load.In will work through demonstrating key functionalities that are key to the success of the application. The changes from the real-world product to the prototype are demonstrated in the following chart mapping how the functionality has changed.

Feature Description Implementation

Prototype Features Table

Feature	**Description**	**Implementation**
	UI	
Login Page	Create a landing page for the login screen.	Full Functionality
Guess User Interface	Ability for a new user to view the application without being logged in.	Full Functionality
Admin User Interface	Ability to have super user privileges when interaction with the application.	Full Functionality
DIY User Interface	Create a user role for the end user with normal privileges.	Full Functionality
	Authentication	
User Registration	Create an account inside of the load.in application.	Full Functionality
User Login	Authenticate with the server that an account exists.	Full Functionality
Reset Login	End user to be able to reset his/her own password inside of load.in application.	Partial:

	Move Inventory	
Furniture/Item Measurement	Ability to measure items using the phone camera.	Partial: Ability to manually type in dimension of boxes.
3D Model Generation	Generate a 3D Model based on item measurements	Partial
Box Locator Search Feature	Ability for the user to find the location of a box via search	Full Functionality
	Move Plan	
Load Plan	Generate a plan that will show users where and how to load boxes	Partial: Preloaded data inside of the database
	Logistics Planning	
Estimated Number of trips	Calculate the number of trips one will take based on the truck size.	Full Functionality
	Expert Help	
Packing Tips and suggestions	A list of tips will be suggested to the user on how to load the truck properly.	Full Functionality
Tips Search	Ability to search items a end user need additional help loading.	Full Functionality
Move expert articles	Expert level tips written by professionals.	Full Functionality
	Analytics	
Feedback data	Ability for the end user to provide feedback for a move.	Partial: User can only give a thumbs up or down.

Figure 2, Load.In Prototype Features Table. This figure shows the functional components of the real-world Load.In application versus the Load.In prototype.

4.1 Prototype Architecture (Hardware/Software)

The Load.In prototype is being created on a virtual machine using Ubuntu Linux operating system. The application will make use of Docker containers, that include Apache Tomcat, CFX for the web API, and MySQL. The user interface will be an Android app interface, and the prototype will also have a test harness interface. The Load.In prototype is looking to be functional with many different platforms that keep it steady. Tomcat is going to be used as the platform for the web API and MySQL will be the platform for maintaining data. The app itself will be an Android application which is how clients will interact with the other systems. The test harness will be a built-in way to create many different testing features as to ensure the functionality of the application. This test harness will be useful in demonstrating how the

application works in many different use cases.

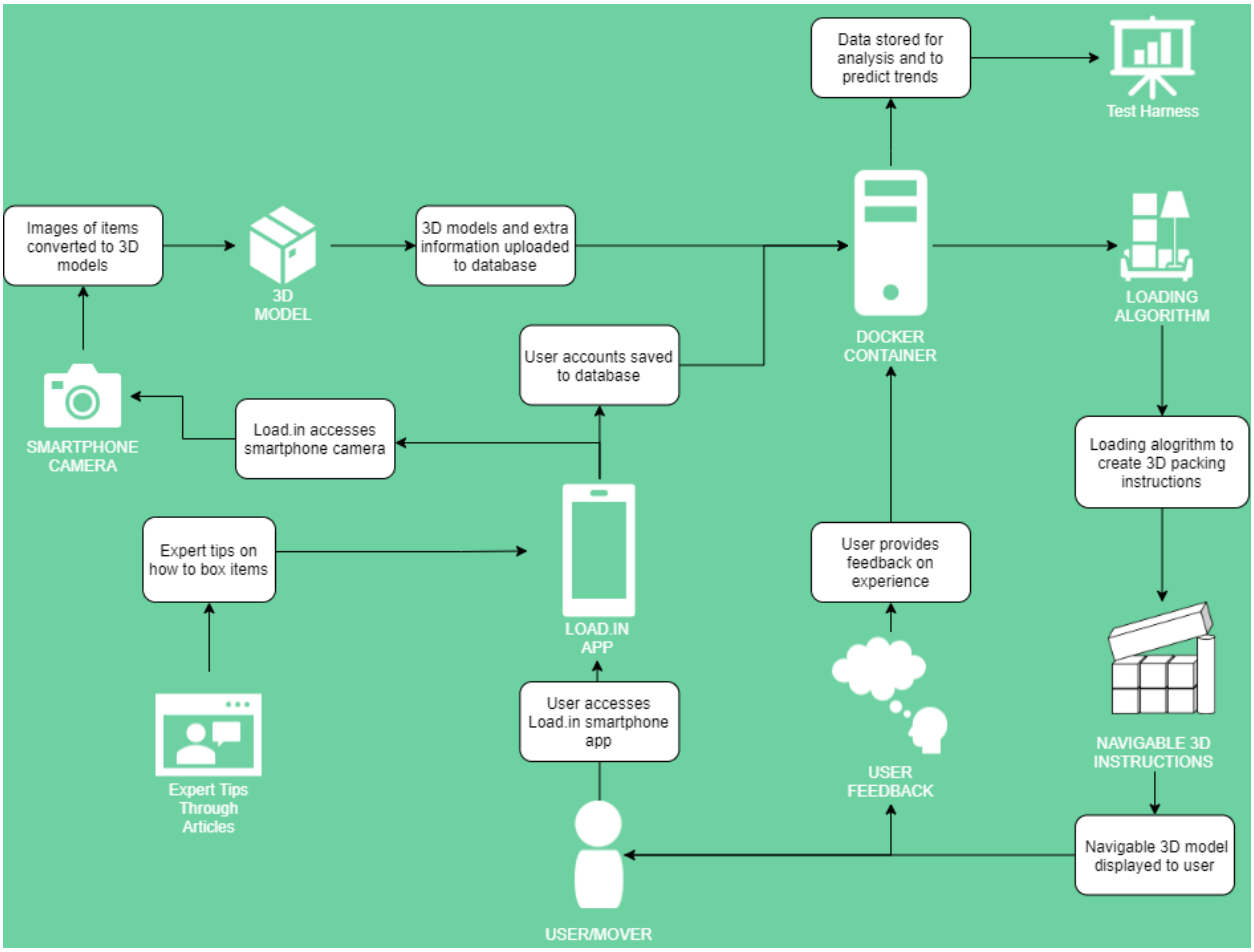

Figure 3, Load.In Prototype Main Functional Components Diagram. This figure shows the main functional components of the Load.In prototype.

4.2 Prototype Features and Capabilities

The Load.In prototype will have enough functionality in order to demonstrate a proof of concept that a full version of the application would be feasible. The prototype will demonstrate the ability to measure an item and generate that item into a 3D model. The items will be kept in an inventory that keeps track of where the items are located within the truck. Based off these models the app will generate a load plan which will guide the user step by step through loading the boxes onto a truck. The prototype will also include an estimation for how many trips a user

will need based upon the inventory and truck space available. The last two features the prototype will include are expert articles and a feedback feature.

These are the core features to what makes Load.In a unique solution to the problems of DIY moving. Some of these features such as the expert tips will be more of a proof of concept than creating a full catalogue of different tips. Load.In will have features that are meant to mitigate some of the risks involved in developing such an application. The feedback feature is meant to mitigate the risk of customers being unsatisfied with the different aspects of the app from how it measures items to how it plans a user's load. Another risk is the user not following the guidelines of the application. Loading boxes out of order or leaving items behind intentionally can alter how a user interacts with Load.In. To mitigate issues that could arise from such changes Load.In will have a feature to revert steps or reset a move. This way a user can be given a new plan based upon the changes they made. The security risks such as the pictures taken by the user, and the security of the user's data will be mitigated by the prototype storing photos locally on the phone, and the data only containing information for the purpose of the move. The data will also be able to be deleted at any time.

The goals of the Load.In prototype are to prove that the application can work in a real-world setting by providing expert tips and a plan to move from one location to another. This will be done through capturing a move inventory, generating a load plan based on that inventory, being able to change truck sizes, generating and searching for expert tips, and the ability to search for items within a load plan. In order to test many of the features the prototype will be able to generate a random inventory so there can be a variety of test cases.

4.3 Prototype Development Challenges

The main development challenges involved with Load.In specifically involve the many variables that will go into a move. This includes the test harness changing values in the database, changing the truck sizes, changing the move inventory, prioritizing results for the expert tips search, determining and setting multiple trips, and the handling of edge cases. The handling of edge cases is probably the most difficult when it comes to this application as it involves essentially every step of the process. Ensuring you cannot overpack a truck or have boxes that are too large for the truck itself. There are potentially problems with the boundaries of each variable and without careful implementation the prototype could fail to run correctly.

5. 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.

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.

5. 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