Lab 1 - Load.In Product Description

Paul Rodriguez

Old Dominion University

CS 411w

Professor Janet Brunelle

1 March 2021

Version 2

Table of Contents

Tables and Figures

1. Introduction

In 2019 an average of 31 million people moved across the United States, which was 9.8 percent of all Americans. A majority most of these moves were done by do-it-yourself (DIY) movers, (Wood, 2020). With the average American moving 11.7 times in their life, most people understand the process can often be tedious and time consuming (Wood, 2020). The do-it-yourself mover is often inexperienced with the logistical aspects of moving including cost estimation, proper packing and loading of boxes, and keeping track of their items' locations. Often a professional moving service can provide many benefits to the customer including packing assistance, loading, and transportation of all their items. However, with the average cost of an interstate move being $4,100 and a in-state move being $980, a majority of people are unable to afford a professional mover (Wood, 2020).

Due to all the issues that arise during the moving process, it seems necessary that an application is created to help the do-it-yourself mover during such stressful times. Load.In will provide the user with an easy-to-use program that allows them to organize their move by providing them with a plan and tips to maximize their efficiency and minimize the costs associated with moving. Through the use of Load.In the do-it-yourself mover is able to make their move as easy as possible.

2. Load.In Product Description

Load.In is a mobile application that is the solution for many issues that arise during the moving process. The goal of the application is to help the do-it-yourself mover save time and money while increasing the speed and efficiency of their entire move. A user creates an account that allows them to begin inputting information relevant to their move such as the start and end

location of their move and an estimation of items volume, weight, and fragility that are going to be moved.

Load.In provides the user with an estimation of needed trips, packing supplies, move time, and the size of truck needed. The user can take photos of their furniture and items which Load.In uses to generate 3D models of their items. All of these generated models are stored within the application's cloud database. After the user has entered the items that they have in their home into the application, a load plan is generated that shows the user how to properly pack their boxes and how to load them into their vehicle.

Load.In provides the user with the ability to see where their items are once everything has been loaded into the vehicle. If the user encounters an issue during the moving process, they can access a chat-bot system that can provide them with expert tips that pertain to their situation, and if needed they can get live support to help them.

Load.In provides the user with the ability to review the application, their move plan, and any help they received from the chat-bot or live move experts. On the back end of the Load.In system, anonymous analytical data is gathered from the moves, and this data helps increase the efficiency of the product and improve user satisfaction.

2.1 Key Product Features and Capabilities

The main feature of Load.In is the ability to generate a customized Load Plan that is created from the information entered by the user. Load.In achieves this by having the user generate 3D models of all the items in their home and providing measurements for large or abnormal furniture. After the user has entered all relevant information, the application creates the load plan which provides detailed instructions on how to properly pack their boxes and load their vehicle.

All of the items that are entered by the user are stored into a Move Inventory Database. This feature allows the user to generate a new load plan if they encounter an error without having to rescan all of their items. Once the user indicates that they have completed their move, the application will delete this database as to not store any user information.

Load.In provides rental estimations to the user including estimated number of trips, estimated time needed to complete their move, and estimated cost of rental trucks based upon their vehicle size. This information is generated using third party vendor information and past anonymous analytical information that has been gathered through the use of Load.In.

During the moving process if the user encounters a situation that is unusual, they can search through a database of expert tips based on keywords. These tips contain suggestions for packing and loading items that are not found in every household such as large or oddly shaped furniture. These tips are gathered from move experts that create articles for the application for situations they have encountered in their professional experience. If the user finds an article that pertains to their situation, they can leave a review based on how the article was able to help them.

Load.In provides the user with a Chat-Bot feature that allows the user to interact with an artificial intelligence that can help them search through the database more efficiently to find a relevant article faster. If the chat-bot feature fails in helping the user, then the user can be connected to a live support agent who can help them with their situation or error.

Load.In utilizes third party vendor synchronization that pulls data from vendor systems and databases and allows Load.In to properly estimate truck prices and availability based on user location. These third-party applications, a web scraper scrapes relevant data from the internet to help increase the accuracy of these estimations.

Throughout the entire moving process, Load.In gathers analytical data that helps lead to increased accuracy as the number of users increases. Some of the types of data that may be gathered include location data, move cost data, feedback data, and anonymous user inventory. The location data may include the start and end locations of a move and the number of trips needed along with the total distance traveled. Move cost data may include all costs associated with the move including gas, cost of rental vehicle, cost of supplies used during move, and time needed to load and unload the vehicle. Feedback data includes all the ratings provided by the user for the entire application experience and any individual expert tips they received during their move. Anonymous user inventory includes number of items, average size weight, and abnormal furniture that was encountered. All data is anonymous and only used for the analysis and improvement of any estimation algorithms being used within the application to improve the user experience.

The Heat Map feature of the application is used for developers to analyze the utilization of smartphones by the users. The Heat Map tracks what the users are accessing the most on the application and how often features are being used to help developers decide what needs to be improved upon. Some issues that the Heat Map helps solve are difficult to navigate user interfaces that reduces the user experience.

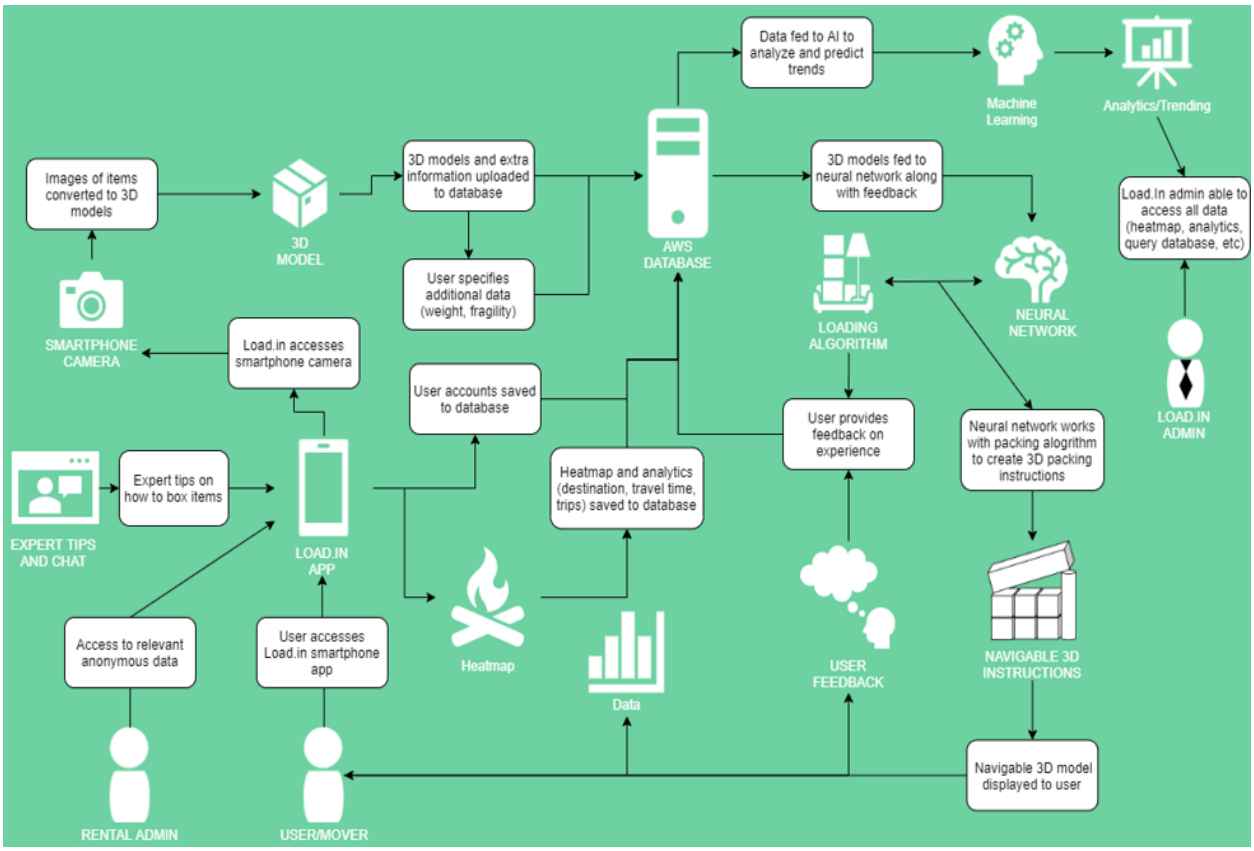
2.2 Major Components (Hardware/Software)

Load.In consists of a web application and a smartphone application. Figure 1 is the major functional component diagram (MFCD) for Load.In. The figure provides insight into how Load.In functions. The end user utilizes their personal smart device to access the mobile application to perform all of their 3D scans and Move Inventory input. All of this information is

passed into a database and run through the loading algorithm to provide the user with their load plan.

Figure 1

Load.In Major Functional Components Diagram



For the hardware, the most critical component for the success of Load.In is the server which is based in the cloud to reduce resource demand on the user's devices. The database uses Amazon RDS which is compatible with MySQL and Amazon's Lambda. The database needs to be able to store large amounts of information so that it may all be processed to provide usable output to the user. Along with this, some of the information that is stored in the cloud can be used by developers and administrators for analytics to provide a better user experience and service in the future. The majority of work falls upon the smartphone of the user which needs to

meet standard requirements. Some of these requirements include Wi-Fi connectivity, a 12-megapixel camera, 4Gb of RAM, and an 8 core processor. For the web application, it can run on a computer which uses Linux, Windows, or Mac OS. The Android application is programmed in Java and can be run on modern Android operating systems found on smartphones. The web API uses AWS Elastic Beanstalk and Apache CFX with Tomcat. The vendor synchronization portal uses AWS Lambda to be triggered on a schedule to draw data from various vendor websites.

3. Identification of Case Study

As Load.In is being developed to help the average do it yourself mover, the case study will be focused on a typical move that may be encountered using average variables collected from the analysis of move statistics across the country. This leads to a scenario where the move is within the same city and within a twenty-mile radius of the original location. The house size will be 2,200 square feet and will contain the items of a three-person family that has one dog U.S. Census Bureau (2019). The family will need to rent a vehicle for their boxes, which will allow the utilization of Load.In's estimation features. For items in the home, the family will have common furniture found in many households. To take advantage of Load.In's ability to share load plans among smart devices, the case study will include friends or family of the user helping them during their move. Since the case study is targeting the average scenario that the product will face, oddly shaped items will be excluded since they are only one factor in the problem that Load.In will help solve. The goal of conducting such a case study on an average home is that Load.In development team will be provided with vital feedback on how the application behaves given the scenario it will most likely be used in. Along with this vital information, the case study will demonstrate how the application can be scaled upwards or downwards in terms of distance of the move or the size of the home, vehicle, or items.

4. Product Prototype Discussion

The prototype for Load.In will have a reduced set of features from the overall functionality of the real-world application. The prototype will serve as a proof of concept and is intended to show what a fully functional application could do for the end user. Two of the features that will have changed functionality are the item measurement and 3D model generation. While the end user will be able to manually input their item dimensions and generate a rudimentary 3D model, they will still be required to manually enter more data rather than the real-world implementation that would automatically do this for them. Through the partial functionality of these two key features Load.In will still be able to provide demonstration of key functionalities that are vital to the success of the application.

4.1 Prototype Architecture (Hardware/Software)

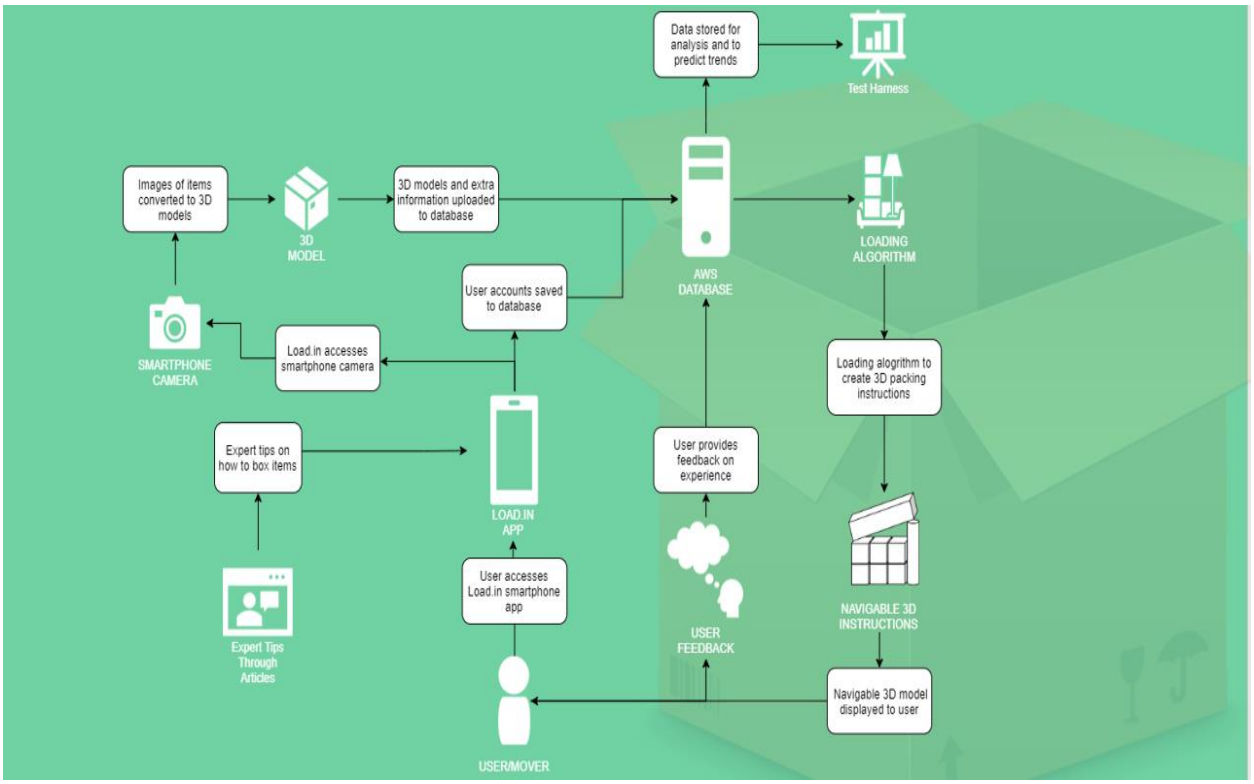
The Load.In prototype operates entirely on a single virtual machine using the Ubuntu Linux operating system. The prototype uses various Docker containers - one to be used for the Apache Tomcat and CFX programs for the web API. Another will contain the MySQL database that will be used to store all data. Apache Tomcat will be the server that hosts the CFX web API that provides the connection between the Android application and the MySQL database. Along with these services, a Test Harness will be developed to provide an interface for testing and demonstration purposes. The Test Harness will also be used in testing the edge cases that may arise throughout a typical move process.

Due to the differences in functional features between the prototype and the real-world implementation of Load.In, the Major Functional Components Diagram that is shown in Figure 1 has been modified to show the changes. One feature that has been removed completely from the prototype is the heatmap with other features such as the expert tips and 3D model generation

having their functional capabilities reduced. The features that will be present in the prototype are shown in Figure 2.

Figure 2

Load.In Prototype Major Functional Components Diagram



The major goal of the prototype is to provide a proof concept when used in the case study that is discussed in section 3. With the features that are shown in Figure 2 being present in the prototype, Load.In will be able to demonstrate the ability to provide a move plan to this family and show the effectiveness of the application in improving the overall moving experience.

4.2 Prototype Features and Capabilities

To provide a working product within the time frame given, the Load.In prototype will only include features needed to provide a successful proof of concept to potential users. As seen

in the Table 1, the main interface for the prototype will be the android application on the user's phone.

Table 1

Load.In 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.

Using the smartphone, Load.In will be able to create a 3D model of the item. If for some reason the user is not able to use the camera on their phone, they can manually enter the measurements of their items. Once the item measurements have been entered in the application, this information will be uploaded and stored into a database. The user will be provided with a load plan which will provide them with instructions to load all the boxes into the vehicle based on the available space. During the moving process if the user encounters an item that they are

unsure of how to pack they can use keywords to search for articles that will provide advice on how to pack that item. After the move has been completed, the user will be given the opportunity to provide fake feedback on their experience so that the Load.In prototype can be improved upon during future iterations.

Throughout the development and testing of the Load.In prototype the largest risk that arises is the security of user information. Load.In ensures the security of the user's private information, including names, photos, and location by storing all of this locally. All user data will only be stored for the purpose and duration of the move. Throughout the entire move process the data can be deleted at any time through the Test Harness. Another risk is the user not following the intended Load Plan provided to them by the application. Loading boxes in the wrong order, loading heavier items on top of smaller items, or leaving items behind can alter the user's interaction with the application. A planned mitigation for this is the application's ability to modify the Load Plan based on any actions the user takes.

Through the implementation and success of these selected features it allows the stakeholder to see the innovation that Load.In provides and showcases that the application can work in a real-world setting by providing expert advice and providing the stakeholder with a plan to move between two locations.

4.3 Prototype Development Challenges

The main development challenges involved with Load.In will be the complexity of the moving process and the number of variables that are involved. A key functional component of the application is the test harnesses' ability to update the values in the database and have the load plan updated to provide a plan for the user. One of the most difficult challenges to the development of Load.In is the handling of edge cases. The application needs to ensure that the

user is unable to overpack a truck or enter a box size that is larger than the truck can hold. The best mitigation for these problems will be constant testing and updating for the edge cases that arise as the prototype is being developed and more test cases are analyzed.

5. Glossary

3D – A three-dimensional form or appearance.

Administrator – Someone who will access elevated features of the Load.In system in order to maintain and detect issues.

Amazon Lambda – A serverless compute service that allows code to be run without the need for provisioning or managing servers.

Amazon RDS – A distributed relational database service provided by Amazon Web Services.

AWS – Amazon Web Services: a cloud platform on which Load.In's databases are hosted.

Android – A mobile operating system based on a modified version of the Linux kernel and other open-source software.

Android Client App – The client-side application for Load.In which runs on the Android platform.

API – Application programming interface: an interface for programs to share information and functionality with one another through a series of calls or connections.

AWS Elastic Beanstalk – An orchestration service offered by Amazon Web Services for deploying applications which orchestrates various AWS services including EC2, S3, Simple Notification Service, CloudWatch, autoscaling, and Elastic Load Balancers.

AWS Elastic File Storage – An AWS service that provides file storage with the ability to auto scale up with increased demand.

Apache CFX – A popular library for hosting web APIs.

Apache Tomcat – An open-source implementation of the Java Servlet, JavaServer Pages, Java Expression Language, and WebSocket technologies. Tomcat provides a "pure Java" HTTP web server environment in which Java code can run.

Chatbot – A feature within Load.In that provides information to users and guides them towards helpful articles and other resources interactively.

Cloud – A term used to describe several computing models such that a company or individual can purchase resources for hosting a variety of things in a centralized location accessible from anywhere in the world.

Computer Vision – a subclassification of artificial intelligence that involves computing information about the world from various sensory data such as images. Techniques of this classification are used throughout Load.In to observe real world objects.

Container Loading Algorithm – A type of algorithm that attempts to optimally fill a three-dimensional space with physical objects. Load.In uses this kind of algorithm to generate Load Plans.

CPU – Central processing unit: the primary component of a computer that processes instructions.

CSS – Cascading style sheet: a style sheet language that is used for formatting the layout of Web pages.

Do-It-Yourself (DIY) Mover – Non-professional movers who rent a truck for their move, and handle all packing, unpacking, and manual labor themselves. This is the primary end user of Load.In.

Expert Tips – Feature of Load.In that allows for a mover to search for helpful articles pertaining to a variety of useful information on how to accomplish various tasks during a move.

GHZ – Gigahertz: a commonly used unit when measuring computer processing speeds.

Guest – Someone who is accessing the Load.In system anonymously and has not registered for an account or someone who has registered but has not authenticated to the system at the time of access.

GUI – Graphical user interface: the aspect of a software program that the end user interacts with.

HTML5 – Hyper Text Markup Language version 5: a markup language used for structuring and presenting content on the Web.

Heatmap – A data visualization technique that shows magnitude of a phenomenon as color in two dimensions.

Java – A set of computer software and specifications that provides a system for developing application software and deploying it in a cross-platform computing environment.

JavaScript – A scripting language that runs in the browser and performs one or more functions to animate an otherwise static HTML document.

Linux – An open-source and community-developed operating system for personal computers and workstations.

Load Plan – A set of instructions on how to optimally load a container – generated automatically by Load.In from the boxes and furniture input into the system by the user.

Logistics Planning – A feature of Load.In that assists the mover with determining what rental trucks cost, how many trips the truck might need to take and whether the truck is available to rent based off proximity to the mover.

Mbps – Mega-bits per second: a unit of measurement for network speeds.

Megapixel – One million pixels: typically used to measure the size and quality of images.

Move Analytics – A feature of Load.In in which information gathered from previous moves are used to determine estimations for future moves as well as predict market trends for Rental Companies.

Move Inventory – A feature of Load.In that catalogs all boxes and items the mover intends to move.

MySQL – An open-source relational database management system.

MacOS – An operating system used on Apple's MacIntosh line of personal computers and workstations.

OS – Operation system: a collection of programs designed to provide a platform on a device to run other applications and typically provides a layer of abstraction from the hardware it interacts with.

Pixel – A small square of color that is part of a larger display screen or image.

Photogrammetry – A computational method of deriving three-dimensional information from images. This method is used in Load.In to construct 3D models of boxes, furniture, and other items from pictures taken from the end user's cell phone camera.

PNG – Portable Network Graphics: a common image file format that Load.In uses.

Professional Mover – Professionals who handle the physical labor of loading and unloading a moving truck as well as driving the truck to the destination.

Rental Administrator – A representative of a rental company who will access the Load.In system on behalf of the rental company.

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

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

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

SPRING MVS – An application framework and inversion of control container for the Java platform.

Test Harness – A set of special features used during the development of Load.In to enable testing and demonstration of the application.

Vendor Synchronization – A feature of Load.in that brings in truck sizes and availability of rental information from third party moving company websites.

Windows – An operating system developed by Microsoft for use on personal computers and workstations.

6. References

Andrew, P. (2020, January 26). Is your house the "typical American home"? *Hsh.* https://www.hsh.com/homeowner/average-american-home.html

CADCode Systems. (n.d.). *Optimizing & machining.* CADCode. Retrieved September 20, 2020, from https://www.cadcode.com/category/categories/optimizing-machining

Collins, T. (2018, April 20). A look into photogrammetry and video games. *Medium*. https://medium.com/@homicidalnacho/a-look-into-photogrammetry-and-video-games-71d602f51c31

Dube, E. (2006). *Optimizing three-dimensional bin packing through simulation.* Semantics Scholar. https://www.semanticscholar.org/paper/OPTIMIZING-THREE-DIMENSIONAL-BIN-PACKING-THROUGH-Dube/bb9986af2f26f7726fcef1bc684eac8239c9b853#references

Economy Moving & Storage, LLC. (2015, January 4). *How to properly pack and load a moving truck- movers Cincinnati* [Video]. YouTube. https://www.youtube.com/watch?v=rjmofUZOdwo&feature=youtu.be

Knoblauch, M. (2019, May 8). One in ten Americans would prefer a week in jail over moving. *New York Post*. https://nypost.com/2019/05/08/one-in-ten-americans-would-prefer-a-week-in-jail-over-moving/

Manwaring, K. (2020, November 24). *The average cost of moving truck rentals.* Move.org. https://www.move.org/average-cost-truck-rental/

Meyers, S. (2018, October 10). *How much does it cost to move a four (4) bedroom house?* Movers.com. https://www.movers.com/moving-guides/moving-four-bedroom-house-costs.html

Nat and Friends. (2017, April 18). *Google Earth's incredible 3D imagery, explained* [Video]. YouTube. https://www.youtube.com/watch?v=suo_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