Lab 2 – Load.In Product Specification Outline

Gregory Kukanich

Old Dominion University

CS411W

Professor Janet Brunelle

March 28, 2021

Version 1

Table of Contents

List of Figures

List of Tables

1 Introduction

Do it yourself (DIY) movers often lack the knowledge to handle the logistics of their move. Professional movers ease the physical burden of moving and handle logistics for the customer, but their costs are often very expensive (Wood, 2020). DIY movers must estimate costs, keep an inventory of the items being moved, pack all their items, and load the moving truck efficiently and safely, the last of which is an extremely important part of the moving process. Weight distribution plays an important role in truck and item safety when transporting one's belongings - an improperly loaded moving truck could result in damaged items or a motor vehicle accident.

Market research has shown that nearly 70% of movers choose to do it themselves (Wood, 2020). This is primarily a result of the high costs that professional movers charge for their services. For example, a four-bedroom household conducting an in-state move would have to pay nearly $3000, and if the move is out of state that cost nearly triples (Meyers, 2018). These prices are not affordable for most Americans, who therefore choose to move themselves. Further market analysis shows that there is a shift in the market towards software-based solutions for helping with the moving process. While there is currently some software available to help a person move, it is basic and helps with issues such as keeping an inventory for the move. No current software will help the mover to the degree that Load.In does.

1.1 Purpose

Load.In is a software-based solution that has an Android application that assists DIY movers by providing 3D modeled loading instructions for the truck and logistical support for the move. Users take photos of the packed boxes and any furniture they are moving. Load.In uses these pictures to create 3D models of all the items in the moving truck to give the user a full visualized plan of how to load the truck properly.

The Load Plan that is generated guides the user on how to load their moving truck while accounting for important aspects such as weight distribution, fragility, size, and importance of the item. Users can search the Load Plan to find items should they need to know where it is after they have finished loading the moving truck. Load.In provides tips and tricks for properly packing different materials and items, in the application to assist the mover. Load.In's time and cost estimates are important features that help the user understand how much time and money they can expect to spend on their DIY move. The goal of Load.In is to deliver a product that allows users to have the expert knowledge and experience of a professional mover without having to pay the high costs that they charge.

1.2 Scope

Load.In is designed to help Do-It-Yourself (DIY) movers handle different aspects of their move by providing support for logistics, inventory, and loading of the truck. The Load.In prototype will be created and tested to be a proof of concept of the key features and ideas behind the application. While the application has reduced functionality from what was initially envisioned, it will allow for further development to be done on the key features during the prototype process.

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.

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.

1.4 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_aUTUpps&feature=youtu.be

Team Yellow. (2021). *Lab 1 – Load.In Product Description*. Old Dominion University. Retrieved from, https://www.cs.odu.edu/~411yello/

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

This product specification details the hardware and software configurations, external interfaces, capabilities and features of the Load.In prototype. The information contained in the remainder of this document includes a description of the architecture of the prototype, the features that the prototype will contain, challenges to development, and the external interfaces of the prototype. The functional, non-functional, and performance requirements for the Load.In prototype will be detailed in a separate document.

2 General Description

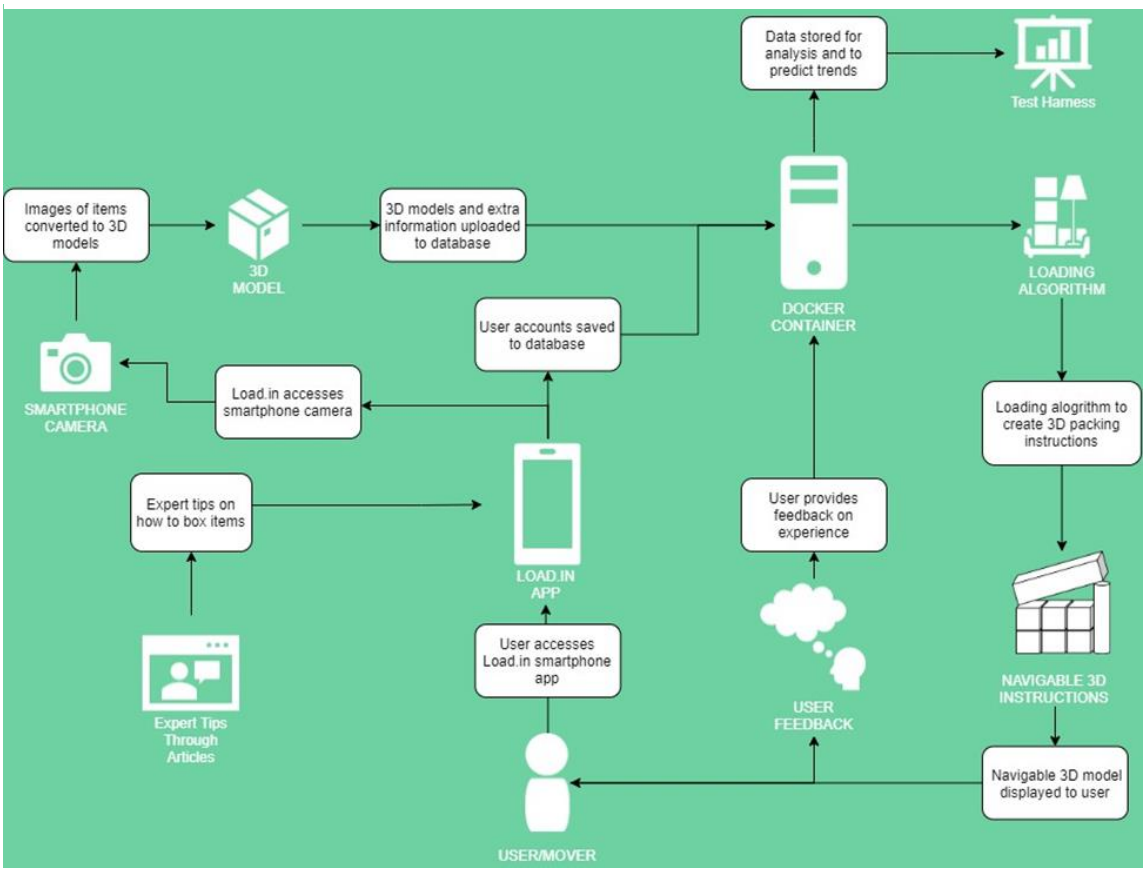
The Load.In prototype will be implemented as an Android application to serve as a proof-of-concept. The Load.In prototype will allow for users to add items to their Move Inventory and have a rental truck Load Plan generated for their items based on criteria such as size, weight, and fragility. The Load.In application will also assist users with certain logistics of their move such as rental truck costs, trip estimates, and distance estimates. The Load.In prototype will also assist users in the packing of their items with the Expert Tips. The Load.In prototype aims to demonstrate its usefulness for DIY movers.

2.1 Prototype Architecture Description

The Load.In prototype will consist of a virtual machine, Android application, web API, and a database. These components will interact with each other to allow prototype users to make an account, receive Expert Tips on packing their boxes, add boxes to their Move Inventory, generate a Load Plan based on their Move Inventory and the rental truck selected. The prototype will also provide logistic regarding the estimated distance, number of trips, and estimated costs for their move. The components of the prototype and their interactions can be seen in Figure 1.

Figure 1

Load.In Prototype Major Functional Component Diagram



The virtual machine (VM) used is provided by the CS department at Old Dominion. The VM is running Ubuntu 16.04, it will host the web API and database used for the Load.In prototype. The web API is built using Java on the Apache CXF services framework and will run inside of a Docker container on the VM. The web API will be responsible for handling authentication of users and allowing users to access or manipulate their data stored in the database. The database for the prototype is a MySQL relational database which will also run inside of a Docker container on the VM. The MySQL database will contain tables storing user's profile information, Move Inventory, and their Load Plan(s). The database will also contain tables for the Expert Tips, user feedback, and rental truck information. The Android application will serve as the user interface for the prototype providing users with the functionality needed to

complete a standard move. The Load.In application will also display the 3D rendered Load Plan to the user in a step-by-step instruction set to aid the user in packing the rental truck. The Android application will make connections to the web API which will provide users the ability store, access and manipulate data in the MySQL database for their move.

2.2 Prototype Functional Description

The Load.In prototype is being built to demonstrate a proof of concept of the important core features of the application. The prototype will provide users with an accurate experience to represent how the real-world-product version of Load.In would operate. The features of Load.In are listed in Table 1 along with how they will be implemented in the prototype.

Table 1

Prototype Features Table

Feature	**Description**	**Implementation**
	Authentication	
Login User Interface	Create a landing page for the login screen.	Full Functionality
User Registration	Create an account inside of the load.in application.	Full Functionality
Reset Login	End user to be able to reset his/her own password inside of load.in application.	Partial
User Login	Authenticate with the server that an account exist.	Full Functionality
DIY User Interface	Create a user role for the end user with normal privileges.	Full Functionality
	Move Inventory	
Furniture/Item Measurement	Ability to measure items using the phone camera.	Partial
Move Inventory Interface	The interface shall consist of multiple different screens to manage the inventory.	Full Functionality
3D Model Generation	Generate a 3D Model based on item measurements	Partial

Box Locator	Users shall be provided with a search bar on the Move Inventory Interface to allow them to search for a specific box by the box number or description, they will be provided with the location of the box.	Full Functionality
	Expert Help	
Expert Tips User Interface	The user interface for expert tips shall alert the end users when an expert tip is found.	Full Functionality
Expert Articles	The expert articles shall provide useful information to the end user based on keywords stored for each article.	Full Functionality
	Load Plan	
Generation Algorithm	The Generation Algorithm for the Load Plan solves placements of Move Inventory items in a given truck.	Partial
Load Plan Display Interface	The Load Plan Display Interface takes a given Load Plan and displays it to the user.	Full Functionality
	Logistics Planning	
Rental Truck Costs	Calculate the cost of renting a truck.	Partial
Move Estimates	Calculate the number of trips one will take based on the truck size.	Partial
Move Inventory Storage	The system shall provide a mechanism by which the user inventory will persist and be accessible anytime by the user.	Full Functionality
	Analytics	
Feedback data	Ability for the end user to provide feedback for a move.	Partial
	Test Harness	

Sample Move Inventory	The test harness shall include a preset move inventory that will be representative of an average move in the United States.	Full Functionality
New Truck Size	The New Truck Size feature in the Test Harness will allow for testing of the Load Plan with different sized trucks.	Partial
	Algorithms	
Container Loading Algorithm	This algorithm works on the back end to generate a load plan that minimizes space of the user's moving truck given a set of boxes.	Full Functionality
3D Model Generation	This feature will be utilized in order to efficiently capture real-world measurements of boxes, which will later be used as input into the Load Plan algorithm.	Full Functionality
Expert tips Algorithm	The algorithm shall take in keywords based on the end users input for box content description during the move inventory phase.	Full Functionality
	Database	
Relational Database	The relational database shall store all persistent application data.	Full Functionality
User Credentials	The system shall provide a mechanism by which the user credentials can be persisted and retrieved at any time for the purposes of authentication.	Full Functionality
	Web API	
Authentication	The web API shall provide both a mechanism for authenticating users to the android client but also to authentication the user when utilizing services for the web API.	Full Functionality

Password Rest	The web API shall receive a request to reset the password from the android application.	Full Functionality

Of the features listed some will only be partially implemented but will still demonstrate the usefulness of the feature. The prototype will feature basic authentication allowing users to create an account and sign-in. Once signed in users will remain logged into their account until closing the application.

The first features that will help in the move process are the packing tips and expert articles. While entering the box description users will be issued a prompt if the input matches a keyword in the packing tips index. This prompt will provide users tips on how to safely package the contents of the box. Prototype users also can search for help on how to pack items around their house and see Expert Articles on the items they are packing. The next important feature in the prototype is the Move Inventory. Prototype users can use their smartphone camera and take a picture of the boxes they are moving to input them into the Move Inventory. The application will use the images to generate measurements and a 3D model of the box. Should they have any issues with this, the user is able to input the measurements manually. This demonstrates the ability to generate 3D models of boxes even if there are issues with the phone's camera measurements.

The Load Plan is another feature that is featured in the prototype. Once all the 3D models of the boxes have been generated, the application will create a Load Plan using the Container Loading Algorithm and then show how to properly pack the boxes on the moving truck. If the user makes any mistakes during the loading process, they can go back a step of the Load Plan, and if the user does not like the Load Plan created, they have the option to generate a new one.

The prototype will also feature Logistics Planning providing trip estimation to allow the

user to see the estimated total distance for their move based on the number of trips needed and distance to travel to the new location. At the end of the move, Load.In has a feedback page to allow the user to provide feedback on their move experience. This feedback allows for improvement of the application by learning where the application may need small changes to improve the process.

2.3 Prototype Development Challenges

While the prototype for Load.In has been scaled back to allow for the demonstration of the more key concepts and features, there are still challenges during the development process. Moving involves many variables and it requires much more time developing Load.In to build an application that addresses and handles all these different variables. To handle a lot of this, there is a test harness. This will allow for the developers to set and change different values in the database of items such as boxes, truck sizes, and more when testing the different features of Load.In. Testing is an important part of the development process for Load.In as there are so many aspects of the move. If one thing goes wrong the whole move process can be affected. By having the test harness and repeatedly testing the application, the developers can address as many of the issues as possible.

2.4 External Interfaces

This section will identify the physical and logical interfaces that are used by the Load.In prototype. The specific characteristics of each interface type are detailed in the following sections.

2.3.1 Hardware Interfaces

The Load.In prototype will be implemented as an Android application, therefore it requires an Android smartphone with a camera and internet connectivity. The camera will be

used for box measurements and the internet connectivity is needed for accessing the Load.In server. The Load.In server is hosted on Old Dominions CS departments provided virtual machine and will utilize their hardware and network infrastructure.

2.3.2 Software Interfaces

The Load.In prototype will use SSL for the encryption of internet traffic. The prototype will use the Retrofit2 library for making HTTP requests to the web API, which will send queries to the MySQL database to edit data or retrieve data to return to the users.

2.3.3 User Interfaces

The Load.In frontend is an Android application, the application will display all necessary data to the user. The application will feature multiple different interfaces within to allow the user to access the different features that are implemented in the prototype.

2.3.4 Communications Protocols and Interfaces

The prototype will require internet access via TCP/IP connections. The prototype will use SSL to encrypt traffic from the web API to the Android application.

3 Product Requirements

The requirements for the Load.In prototype will be included in a separate document titled "Lab 2 Section 3 – Product Requirements". The document will contain the functional, non-functional, and performance requirements for the prototype. It will also contain the necessary assumptions and constraints for implementation.