Lab 2 – Product Specification Outline

Paul Rodriguez

Old Dominion University

CS411W

Professor Brunelle

3/28/2021

Version 1

Table of Contents

List of Figures

List of Tables

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 DIY 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 the issues that arise during the moving process, it is necessary that an application is created to help the DIY 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 DIY mover is able to streamline their move to reduce the stress and anxiety that it brings.

1.1 Purpose

Load.In is a mobile application that using expert knowledge provides a solution for the issues that arise during the moving process. The goal of the application is to help the DIY 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 the volume, weight, and fragility of the items 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 Load 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.

1.2 Scope

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.

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_aUTUps&feature=youtu.be

Rodriguez, P. (2021, March 22). *Lab I - Load.In Product Description*. Retrieved March 28, 2021, from https://www.cs.odu.edu/~411yello/deliverables/labs/prodriguez_lab1_v2.pdf

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 provides the hardware and software configuration, external interfaces, capabilities and features of the Load.In prototype. The information provided in the remaining sections of this document gives a detailed description of the hardware, software, and external interfaces of the prototype; the architecture of the prototype; the features that the prototype will provide, and their implementation level compared to a real-world product; the testing methods and challenges faced during development of the prototype; and the functional requirements of the prototype.

2 General Description

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.

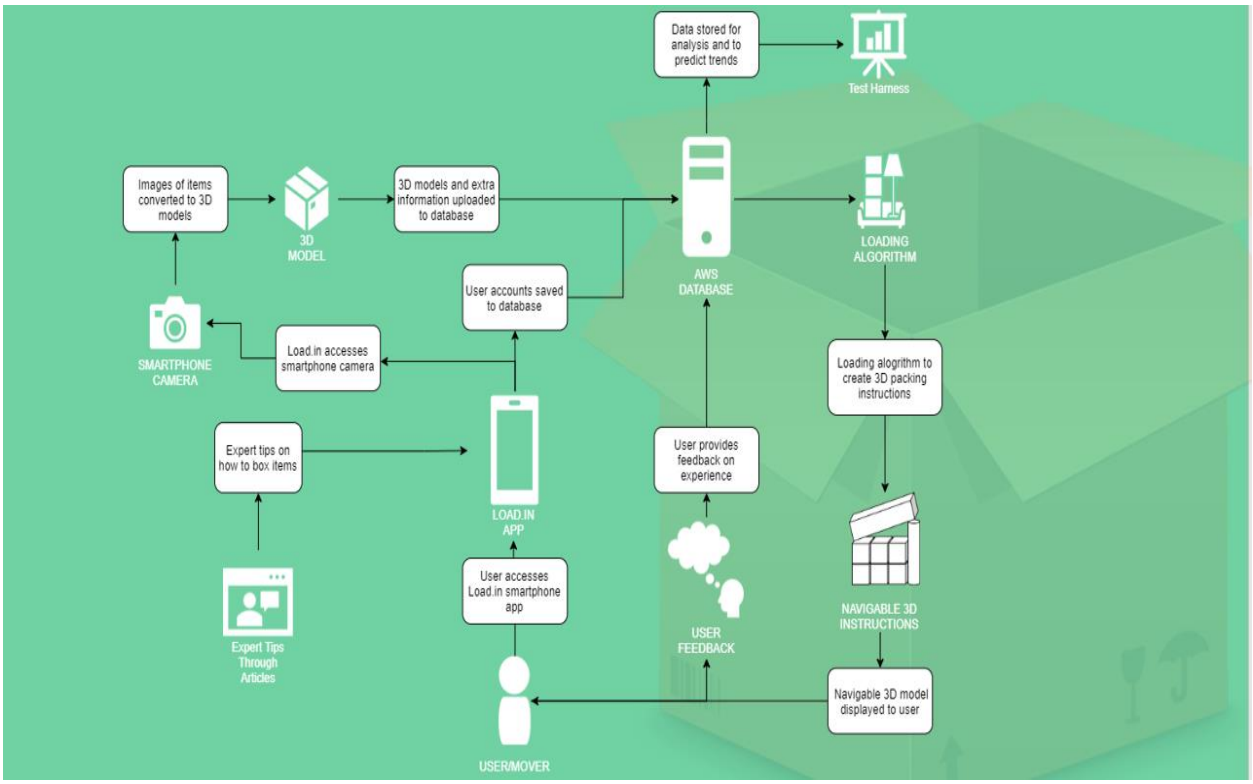
2.1 Prototype Architecture Description

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.

Figure 1

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 Load Plan to this family and show the effectiveness of the application in improving the overall moving experience.

2.2 Prototype Functional Description

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.

2.3 Prototype Development Challenges

The main development challenges involved with the Load.In prototype 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 the Load.In prototype 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.

2.4 External Interfaces

The Load.In prototype will rely on hardware, software, and networking components to provide features to the end user. The user's smartphone device will be utilized to run the Android application that will act as the front-end user interface. The user's smartphone device must have a camera in order to use the photogrammetry feature of the Load.In prototype. For software interfaces, Load.In's web API is built on the Apache CXF framework. The backend database is built using MySQL. The networking components of the Load.In prototype require that the Docker containers interface using TCP/IP, DNS, DHCP, and various open ports to support the HTTPS communication that is utilized by the web API.