Lab 2 – Load.In Product Specification Outline

Byron Aquilino

Old Dominion University

CS411

Janet Brunelle

28 March 2021

Version 1

Table of Contents

1 Introduction 3

1.1 Purpose 3

1.2 Scope 4

1.3 Definitions, Acronyms and Abbreviations 5

1.4 References 9

1.5 Overview 11

2 General Description 11

2.1 Prototype Architecture (Hardware/Software) 11

2.2 Prototype Features and Capabilities 13

2.3 Prototype Development Challenges 15

2.4 External Interfaces 16

List of Figures

Figure 2 Load.In - Prototype Major Functional Component Diagram 12

List of Tables

Table 1 Real World Product versus Prototype 13

1 Introduction

When it comes to relocation, not every American can afford to pay for professionals (Wood, 2020). Every move incurs expenses. These expenses include moving truck costs, packing material, gas, and insurance. In addition to these common expenses, professional movers incur labor costs as well (Wood, 2020). Out of the nearly 31 million moves that occur every year, a staggering 78.3% of movers move themselves (Wood, 2020). Compared to the cost of a "do it yourself" (DIY) relocation over a short distance, a professional relocation over the same distance is on average approximately seven times more expensive because of the additional labor expense (Wood, 2020). DIY movers have a strong incentive to want to move themselves.

However, the lack of professional management is not without its downside. Professional movers bring additional benefits to the mover such as expertly packing items safely and securely, loading a truck safely, and loading a truck efficiently. DIY movers do not know how to pack and load as well as a professional mover can. This lack of expertise can cause issues to a DIY move ranging from frustration and anxiety to damaged property and time wasted due to mistakes (Knoblauch, 2019). Additionally, the DIY mover also faces other difficult logistical considerations such as knowing how big of a truck to rent, how to organize his boxes, and keeping track of all his possessions.

Load.In is a software-based solution that helps the DIY mover acquire expert knowledge. Load.In has features based on Computer Vision, Artificial Intelligence, and 3D model generation to help the DIY mover plan, prepare, and execute his move successfully. This provides an approach that features the best aspects of a professional move with the cost effectiveness of a traditional DIY move.

1.1 Purpose

Load.In has key features such as Load Plan generation, Move Inventory, Expert Tips, and Logistics Planning. These key features address one or more aspects of the DIY moving problem. In addition to providing benefits to the DIY mover, Load.In introduces several key benefits to the rental company industry as well by improving advertisement of available products, reaching new potential customers, and providing insights into moving trends. The Vendor Synchronization feature scrapes data from rental companies into Load.In so that DIY movers can see what rental trucks are available within proximity of the move and help guide movers to the right choice of rental vehicle based on customized search parameters. Move Analytics, another critical feature of Load.In, provides valuable analytics for consumption by the rental companies so that they can better forecast moving demands and trends. The Feedback feature gets feedback from the customers on truck rental quality as well as how well Load.In performed.

1.2 Scope

The Load.In prototype will demonstrate the key features of the real world product such as the Load Plan, the Move Inventory, the Tips and Tricks, the Logistics Planner, and the Box Locator. The mover will register for an account in the registration section. They will then be presented with the option of entering their move inventory into the system. From there they can manage their Move Inventory such as searching for an item (Box Locator), removing an item, and editing an item. They will be able to take their Move Inventory and using the Logistics Planner, choose from different truck sizes. They will also be able to generate a Load Plan and walk through the steps which will show the mover how to load the truck.

Load.In's major components will be the Android application, the main database hosted on MySQL, and the web API hosted in Apache CXF. Apache CXF and the MySQL database will be hosted by the docker platform and will be hosted on a single CS department virtual machine.

The goals and objectives of the Load.In prototype will be to demonstrate the key features that make Load.In innovative by showing that a mover can use Load.In to have it successfully aid in the management of their move.

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

Aquilino, B. (2021, March 2021). *Lab I – Load.In Product Description*. Old Dominion University: CS411 Janet Brunelle.

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

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 outline will over the following topics: the general description of the prototype, the prototype architecture, prototype features, the development challenges involved with the creation of the prototype, and the external interfaces that the prototype will interact with. The prototype architecture will go into details about what platform the prototype will run on as well as the hardware and the major functional components. The prototype features will talk about each feature present in the prototype as well as the order that they will be demonstrated. The development challenges will talk about the specific challenges in solving certain aspects of the prototype including limitations or assumptions associated with the prototype's problem domain. The external interface section will discuss the presence or impact of several interfaces and what that means for the prototype.

2 General Description

The Load.In prototype attempts to take the most important features from the real-world product and demonstrate them in a way that still proves the functionality and innovation of the system while reducing the features' scope and complexity. The Load.In prototype includes features such as the Load Plan, the Move Inventory, the Logistics Planning, and the Expert Tips.

2.1 Prototype Architecture (Hardware/Software)

Load.In's prototype architecture is distinct from the real-world product's architecture due to the nature of the reduced feature set. Instead of using AWS, the Load.In prototype will be

hosted by a series of containers that will be deployed to a single virtual machine running Ubuntu Server 16.04 and Docker, which operates as the main platform for the containers. The prototype's main components will consist of the Android client application, the web API for brokering the communication to the database, the database hosted on MySQL, and a test harness running as a standard Java application. The web API container will host an instance of Apache CXF which runs on Tomcat and a Linux kernel. The MySQL container will host an instance of MySQL and will ultimately host the data that Load.In will utilize from the web API. Both the Android client and the test harness will interface with the web API in order to exchange data and be able to operate. Figure 2 shows the interactions of these components with one another.

Figure 1

Load.In - Prototype Major Functional Component Diagram

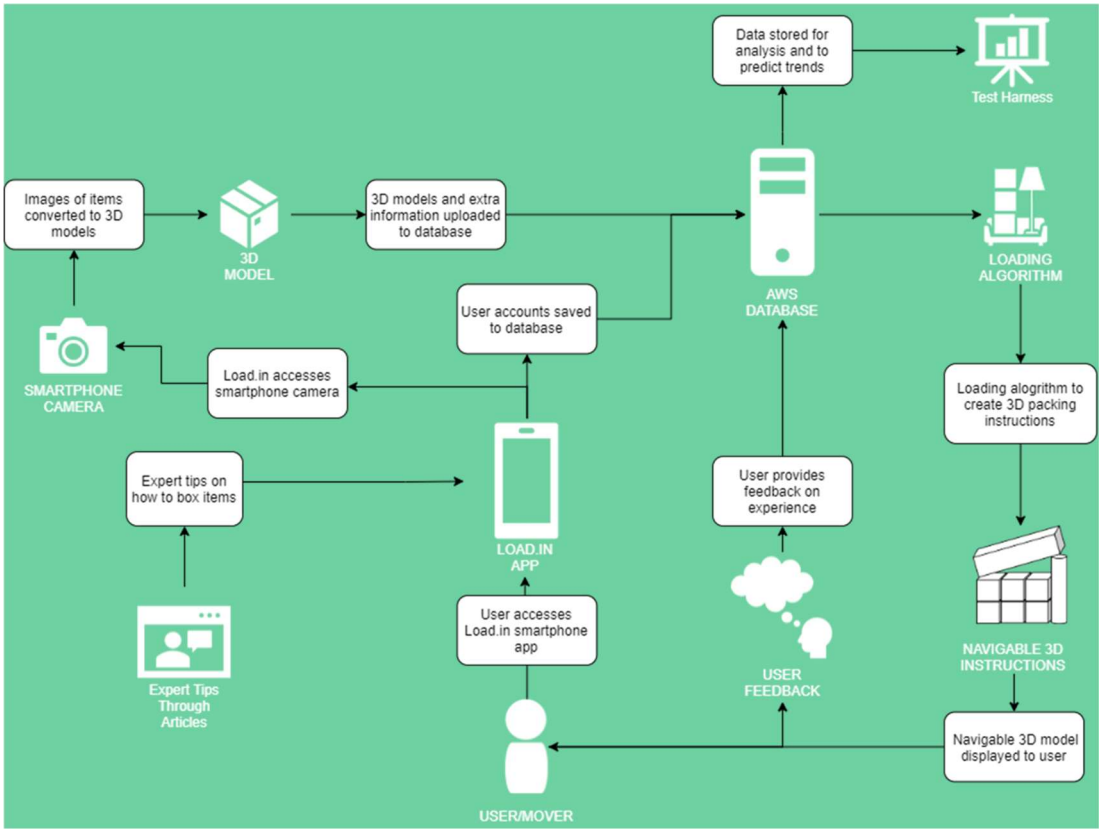

Figure 2 shows the reduced capabilities of the prototype as well. The smartphone client provides the interface for the mover for the interaction with the Expert Tips feature, the 3D model generation, Load Plan display and Move Inventory features.

2.2 Prototype Features and Capabilities

In order to demonstrate the functionality of the prototype, the Android client will serve as the main application and will host the critical features such as the Load Plan, Move Inventory, and the Logistics Planning. The test harness GUI will assist the Android application by allowing a test user to be able to generate data pertaining to a move, setup user accounts, load Expert Tips, and automatically load truck sizes or change truck sizes. This will simulate the other features that are not present in the prototype application such as the Vendor Synchronization. From the Android application the following actions will be demonstrated: measuring a non-box item using photogrammetry, entering in a box dimensions and content description, 3D model generation, locating a box from the inventory, generating a load plan, estimating a move's number of trips, getting packing tips, finding expert articles, and providing feedback for a move experience. These actions demonstrate the core features of Load.In. Table 1 shows a comparison of features from the RWP, also known as real-world-product, versus the prototype.

Table 1

Real World Product versus Prototype

Feature	Real World Product	Prototype
Move Inventory		
Furniture/Item measurement	Fully Functional	Partial
3D model generation	Fully Functional	Partial

Item weight	Fully Functional	Eliminated
Item fragility	Fully Functional	Eliminated
Box locator search feature	Fully Functional	Fully Functional
Move Plan		
Load Plan	Fully Functional	Partial
Truck unloading instructions	Fully Functional	Eliminated
Logistics Planning		
Estimated number of trips	Fully Functional	Fully Functional
Estimated time to move	Fully Functional	Eliminated
Estimated rental truck costs	Fully Functional	Eliminated
Expert Help		
Packing Tips and suggestions	Fully Functional	Fully Functional
Tips search	Fully Functional	Fully Functional
Move experts' articles	Fully Functional	Partial
Chatbot	Fully Functional	Eliminated
Live expert	Fully Functional	Eliminated
Vendor Integration/Data Import		
3rd party vendor web scraper	Fully Functional	Eliminated
3rd party vendor web API reader	Fully Functional	Eliminated
Box dimensions	Fully Functional	Eliminated
Truck sizes	Fully Functional	Eliminated
Truck availability	Fully Functional	Eliminated

Analytics		
Location data	Fully Functional	Eliminated
Move data	Fully Functional	Eliminated
Feedback data	Fully Functional	Partial
Heatmap	Fully Functional	Eliminated
Rental interest statistics	Fully Functional	Eliminated

There are risks that the prototype will have which it must address. There are two main risks from a customer perspective that the Load.In prototype must address. The first risk that the prototype confronts is the mover may not like aspects of his move. The implementation of the feedback feature mitigates this risk, and the feature will be present in the prototype the same it is in the real-world-product. Second, if the end user misses a step in the load instructions generated by the Load Plan feature, the user needs to be able to repeat one or more steps so that this issue does not impact the quality of the move. To solve this, Load.In's prototype will include a feature that allows the mover to back up or rewind any number of steps so that the display can replay steps in the correct order and the mover can make sure they have followed the instructions appropriately.

2.3 Prototype Development Challenges

There will be challenges related to the development of Load.In's prototype. One of the biggest challenges is the development of the test harness and the implementation of functions or routines that will allow for the change of values from the prototype's database such that the client Android application reflects the changes. One of these scenarios that needs to be demonstrated is what happens when the available trucks change on a mover after a mover has

selected his truck for a move. The prototype needs to be able to demonstrate to an audience the impact of such an event on the Load Plan and ultimately the number of trips for the mover. Another challenge is changing a Move Inventory to show the impacts to a Load Plan. For example, a mover might decide to add in other boxes to the Move Inventory, or the mover may have a wide variety of odd-shaped boxes.

2.4 External Interfaces

The Load.In prototype will rely on hardware, software frameworks, networking components to provide features to the movers. For hardware, the mover must utilize an Android smartphone device capable of running the application as listed under the prototype architecture section. It must have a camera to take photos or to use the computer vision features. It must have speakers to playback audio from the expert tips. For the software interfaces, Load.In will rely on the Apache CFX framework and MySQL. For the networking interfaces, Load.In will require that the operating system of the containers interface with TCP/IP, DNS, DHCP, and open port communications on the networking infrastructure for HTTPS communications to and from the web API. Load.In will require the use of SMTP for transmitting of emails for resetting passwords.