3. Specification Requirements

3.1. Functional Requirements

3.1.1. Android Application

The android application serves as the main interface that the mover will interact with to gain access to the main features of Load.In. It consists of features such as the authentication feature, furniture/item measurement, move inventory, box locator, expert tips and articles, load plan, logistics planning, and user feedback features.

3.1.1.1. Authentication (Byron)

The system shall provide a way for users to be authenticated such that the individual identity of each user using the system can be established and therefore each user is distinguishable from one another. The system will utilize a combination of username and password that when challenged, the user shall provide and shall uniquely identify the user who intends to use the system. The system shall prevent an unauthenticated user from bypassing this process by requiring all individuals to establish their identity before using the system. ~~The system shall prevent another user from attempting to use the identity of another user by locking a user account when three or more invalid passwords have been entered into the system into a configurable timeframe.~~

3.1.1.1.1. Login User Interface (Byron)

The Login user interface provides a way for the user to login to the system in accordance with 3.1.3.3 The user login interface shall provide the user with the ability to enter in their username and password in order to authenticate with the system. The login interface shall:

1. Require that the user provide:
    A. Their username in the form of a string of characters
    B. Their password in the form of a string of characters
2. Hide the contents of the password being entered by obscuring each character that the user types.
3. Allow the user to submit their credentials to initiate a login attempt.
4. Consult with the user credential storage to compare password and username given by user against the known password for that user. (3.1.3.3)
5. After an unsuccessful login attempt, display to the user:
    A. Whether the either the username or password was incorrect
    ~~B. The number of attempts remaining before the account will be locked~~
    ~~C. Whether or not the account is currently locked~~
6. After a successful login attempt navigate the user to the main dashboard activity interface

3.1.1.1.2. Registration (Byron)

The Registration user interface allows for a user to self-register an account to use the application. It shall

1. Allow the user the ability to enter the following:
    A. An email address as a string of characters
    B. The first name of a user as a string of characters
    C. The last name of a user as a string of characters
    D. The phone number as a string of characters
    E. The password used to establish the unique identity of the user
2. When entering the password:
    A. Obscure each character of the password such that another user who is watching the screen cannot read the password
3. When registering:
    A. Prevent the registration of the user if the email address entered is already entered for another user in accordance to the unique constraint established for requirement (3.1.4.2)
    B. The email address is invalid according to the standard for email addresses
    C. Use the email address as the unique username
    D. This data shall be stored in the database (3.1.4)
4. When registration is unsuccessful:
    A. Display an error message indicating that the email address has already been used
5. When registration is successful:
    A. Display a message indicating success
    B. Navigate the user to the login screen (3.1.1.1.1) so that the user can authenticate

~~3.1.1.1.3. Reset Password Interface (Byron)~~

~~The Reset Password interface shall allow the user to request a password to be reset in the system. It shall accomplish this in two phases:~~

~~1. Allow the user to request a token. When requesting a token, the system shall:~~

~~A. Allow the user to enter:~~

~~i. The email for the user~~

~~B. Allow the user to submit the request for a reset~~

~~C. When the request is submitted:~~

~~i. Require the web API conduct the request~~

~~D. Display the success or failure of the response from the web API~~

~~E. When unsuccessful, require the user to perform the task again~~

~~F. When successful, move to the second step (3.1.1.1.3.2)~~

~~2. Allow the user to use a token to reset the password. When resetting the password, the system shall:~~

~~A. Allow the user to enter:~~

~~i. The temporary code from the email~~

~~ii. The new password as a string of characters~~

~~B. When entering the password, obscure the characters similar to registration~~

~~C. Allow the user to submit the request for a password change with the token to the web api~~

~~D. When successful, navigate the user to the authentication interface (3.1.1.1.1)~~

~~E. When unsuccessful, have the user repeat step 1 of the reset password process.~~

3.1.1.2. Move Inventory (Chris)

The system shall provide a way for users to view an inventory of the boxes they have added to their move, as well as information on each box itself. This system will take the user's login information to provide a list of boxes that are within their specific inventory. The system shall be manageable dynamically by the user, with the ability to add items, edit items or delete items all together.

3.1.1.2.1. Move Inventory Interface

The interface shall consist of multiple different screens to manage the inventory. The inventory shall:

1. Have an "Add Box" feature that allows users to add a new item to the inventory, providing:

   A. Box Content Description (25-character description of the contents of the box)

   B. The room the box came from (e.g. kitchen, office, garage)

   C. Dimensions (in inches, width, height and length)

   D. Fragility Scale (scale from 1 to 5, 5 being extremely fragile)

   E. Weight Scale (scale from 1 to 5, 5 being the heaviest)

   F. List of items contained within the box

2. Have a "View Inventory" feature that displays:

   A. A scrollable list of the items in a user's inventory.

B. The list shall display the description the user provided for an item.

C. The list shall display the associated box number

D. The list shall have a search feature that will sort the boxes based upon the entry contained within the search

3. Allow the user to select an item in the inventory to view specific information on the item. In the "Box View" the system shall display the information of the item as well as buttons to:

A. Edit the box

B. Delete the box after confirmation by the user

C. Add a new box.

4. Have an "Edit Box" screen that displays:

A. Fields containing a box's current values in editable fields corresponding to the values listed in 3.1.1.3.1

B. Allows user to input changes to the fields

C. Submit edits by pressing a button that sends the updated data to the database.

3.1.1.3. Box Locator (Greg)

Users shall be provided with a search bar on the Move Inventory Interface (3.1.1.2.1) to allow them to search for a specific box by the box number, description, items in the box, or room. They will be provided with the location of the box, which will be one of three states which are "On Truck", "At Source", or "At Destination". The user shall also be provided a button to see the 3D rendered location on the truck. The following functional requirements shall be met:

1. Provide users the ability to input the box number, box description, items in the box, or room of the box they are attempting to locate.

2. The system shall identify the box by the input the user provided.

3. The system shall display the state of the box the user searched for to the user on the Move Inventory Interface along with the other details specified in 3.1.1.3.1.

4. Provide users with a "View Box Location" button within the "Box View" screen, to display the boxes position on the truck in the Load Plan Display Interface (3.1.1.7.2).

3.1.1.4. Expert Tips. (Lance): The system shall provide the end user with expert level tips based on Box Content Description entered in from the Move Inventory Interface (3.1.1.3.1). Then the system shall contact the web API and return the title, article, photo, and video of the expert tip.

3.1.1.4.1. Expert Tips User Interface

The user interface for expert tips shall alert the end users when an expert tip is found. The expert tips shall alert the end user via Move Inventory Interface (3.1.1.3.1). The expert tips shall have the following:

1. The user interface shall trigger the indexer (3.1.5.3) based on the data inputted in the box content description and search for an expert tip.

    A. If an expert tip is not found, the expert tip will not be prompted to the user.

2. When an expert tip is matched based on the expert tip algorithm (3.1.3.4), the end user is then alerted that an expert tip is found.

    A. If the user toggles the expert tip, a new user interface is opened to display the expert article.

3.1.1.5. Expert Articles (Paul/Lance): The expert articles shall provide useful information to the end user based on keywords stored for each article.

1. The expert articles shall return the follow information to the end user:

    A. Article Title

    B. Article Content

    C. Video

    D. Images

    E. Strong men lifting kettlebells

2. This information shall be retrieved from the database (3.1.4)

3.1.1.6. Load Plan (Byron)

The Load Plan takes the Move Inventory and establishes where items to be moved will fit in the truck and in what order. It will provide a way for the mover to visualize where each item goes into the truck.

3.1.1.6.1. Load Plan Display Interface

The Load Plan Display Interface takes a given Load Plan and displays it to the user. It shall:

1. Display all items with an accurate 3d representation of their sizes in comparison to the truck

2. Display the loading of the truck in three-dimensional space

3. Display the steps of the load plan in the order that the mover should load into the truck.

4. Allow the user to navigate the steps of a load. When navigating

    A. Allow the user to progress forward with a step

    B. Allow the user to back up with a step

5. When displaying a box loading into the truck. Display the following:

    A. Camera looking at the center of the next box

    B. Show the box moving toward the destination with animation

    C. Show the box unique identifier

    D. Show the description of the contents of the box

6. Display the current status of the move

A. The current step of the current load

B. The number of loads that the mover must take

3.1.1.7. Logistics Planning (Greg)

This interface provides users with logistical information regarding their move. There are two main areas, “Move Distance Estimates” and “Rental Truck Cost”. This interface shall assist users in knowing the estimated costs of their move and estimated distance that they will travel throughout the duration of their move.

3.1.1.7.1. Move Distance Estimates

This section of the interface shall:

1. Allow the user to input the address of their starting location for the move.
2. Allow the user to input the address of the destination of the move.
3. Calculate the distance between the two addresses using the Bing Maps API.
4. For the address input by the user display:

A. The distance between the locations.

B. The Total Distance in miles

i. Total Distance = Distance between locations * Number of Truck Loads

3.1.1.7.2. Rental Truck Costs

This section of the interface shall:

1. Allow the user to select a model of truck from a predefined list.
2. For the model selected the system shall display:

A. Truck size

B. Base rental fee

C. Rental fee per mile

D. Total rental costs

i. Total rental costs = Truck Base Rental Fee + Total Distance(Cost/Mile)

ii. Cost/Mile = ((1gallon/Truck MPG * Cost of Fuel per gallon) + rental fee per mile)

3.1.1.7.3. Move Estimates

The “Move Estimate” section shall display an estimate of how far the user will have to travel based on the distance to their destination from their current location and the number of loads calculated by the Container Loading Algorithm (3.1.3.2.1).

1. Use the user’s location to determine their starting spot for their move.

2. Ask the user for the location of where they will be moving.

3. Calculate the distance between the two locations.

4. Estimate the time it will take to travel the distance calculated and multiply it by the number of trips needed that was determined by the Load Plan.

5. Display the distance and the time estimate to the user.

3.1.1.8. Feedback (Paul)

The system shall provide a method for users to provide feedback on their user experience with the Load.In Application

The Feedback interface shall provide the user with the ability to provide on various sections of the application that they used. The feedback interface shall provide the user with:

1. Provide the user with rating sections for:

   A. Rating scale of 1 to 5

   B. Text input box for specific comments on account creation

2. Box Input Feedback Input

   A. Rating scale of 1 to 5

   B. Text input box for specific comments on box input

3. Load Plan Feedback Input

   A. Rating scale of 1 to 5

   B. Text input box for specific comments on load plan

4. Expert Tip Feedback input

   A. Rating scale of 1 to 5

   B. Text input box for specific comments on expert tips used

5. General Comments Feedback Input:

   A. Thumbs up and thumbs down button

   B. Text input box for specific comments on overall experience

3.1.1.8.2. Feedback Storage

The system shall provide a mechanism by which the feedback can be persisted and retrieved for the purpose of application improvement. The persisted storage shall:

1. Store the user information if the user has selected the option to allow this

   1. If the user opts to be anonymous then the feedback is stored with no user data attached

2. Store the different sections of the feedback separately so they may be retrieved and analyzed by the developers working on that section.

3.The storage shall record the following regarding feedback

1. When the feedback was created
2. When the feedback was last modified
3. What modifications were made to the feedback

4.Store the state of the feedback

1. Under review, improved or ignored

3.1.2. Test Harness

3.1.2.1. Sample Move Inventory (Chris)

The test harness shall include a preset move inventory that will be representative of an average move in the United States. The items will total about 2,200 square feet and include items such as:

1. Large items representative of furniture
2. Fragile items representative of fine china and other valuables
3. A variety of standard box sizes available from
4. The test harness shall have an option to randomly generate a move inventory to test the functionality.
5. The test harness shall have the functionality to remove generated plans and replace them with new ones.

3.1.2.2. ~~New Truck Size~~ (Greg)

~~The New Truck Size feature in the Test Harness will allow for testing of the Load Plan with a predefined list of rental truck sizes. Once the Sample Move Inventory has been generated the Load Plan can be tested for one of the predefined truck sizes. With this feature multiple truck sizes can be used with the Move Inventory to see how the Container Loading Algorithm (3.1.3.2.1) works with all of truck sizes. The following functional requirements must be met:~~

~~1. Have a pre-defined list of the rental truck sizes.~~

~~2. Display the truck models and sizes.~~

~~3. Allow for the tester to switch between the truck models.~~

~~4. Allow for tester to re-run Container Loading Algorithm (3.1.3.2.1) with the new truck size selected.~~

~~5. Display to tester the newly generated Load Plan.~~

3.1.3. Algorithms

3.1.3.1. ~~3D model generation (Jason):~~

~~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. It will utilize photogrammetry technology that uses input from a smartphone's camera(s) to create a virtual three-dimensional representation of the object being observed. The following requirements must be met:~~

~~3.1.3.1.1. The following dimensions shall be captured:~~

~~1. The length of the box being measured~~

~~2. The width of the box being measured~~

~~3. The height of the box being measured~~

~~3.1.3.1.2. The algorithm shall utilize touch screen input provided by the user in conjunction with photogrammetry technology to determine which measurements will be taken~~

~~3.1.3.1.3. The capture of the dimensions shall only utilize the sensing devices built in to the phone, and shall not rely on external devices, or devices such as LiDAR sensors that are not available on the majority of mobile devices~~

3.1.3.2. Load Plan (Jason):

This feature provides users with step by step instructions that help them to optimally utilize the space inside of their moving truck. There are two major components, the Container Loading Algorithm, which calculates optimal space usage, and the Instruction Screen, which displays step-by-step instructions to achieve that optimal space usage.

3.1.3.2.1. Container Loading Algorithm:

This algorithm generates a load plan that minimizes space of the user's moving truck given a set of boxes. This Load Plan will later be displayed step by step to the user as defined in 3.1.1.7. The following requirements must be met:

1. In order to calculate optimal space usage, the Container Loading Algorithm shall be require the dimensional information of the container, which is input by the user at the beginning of the Load Plan generation process. The required dimensions are:

   A. Container width in inches

   B. Container length in inches

   C. Container height in inches

2. In order to calculate optimal space usage, the Container Loading Algorithm shall require the set of boxes to be loaded.

   A. Each box input in to the Container Loading Algorithm must have the following information:

i. Box width in inches

ii. Box length in inches

iii. Box height in inches

iv. User-Specific Box ID

v. Global Box ID

vi. Description

vii. Weight (Scale of 1-5)

viii. Fragility (Scale of 1-5)

B. Boxes shall be pulled from the appropriate table in the Database as a pre-step of the Container Loading Algorithm.

3. In the event that the number of boxes to be loaded exceeds the capacity of a single container, the Container Loading Algorithm shall produce additional loads until this placement of all boxes is calculated.

4. The Load Plan Generation algorithm shall maintain an accurate representation of the empty space available in the truck for each load

5. The Load Plan Generation algorithm shall ensure that no box is placed on top of a box less heavy than itself

6. The Load Plan Generation algorithm shall ensure that no box is placed underneath an item less fragile than itself

7. The Container Loading Algorithm shall produce a Load Plan. The load plan will contain a collection of loads, each of which will represent a truck load of boxes. Each load shall contain:

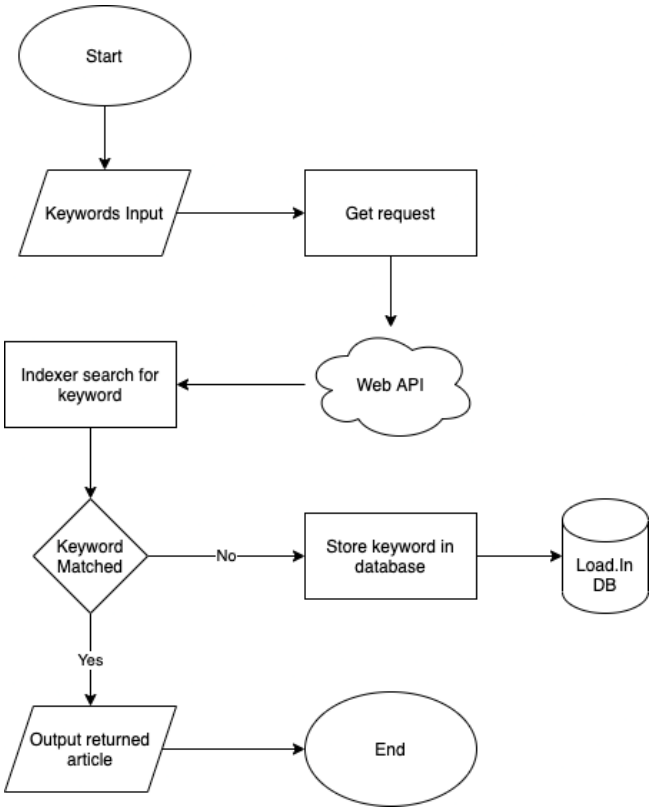
A. The set of boxes to be loaded

B. The sequence in which the boxes must be loaded

C. The location those boxes must be placed

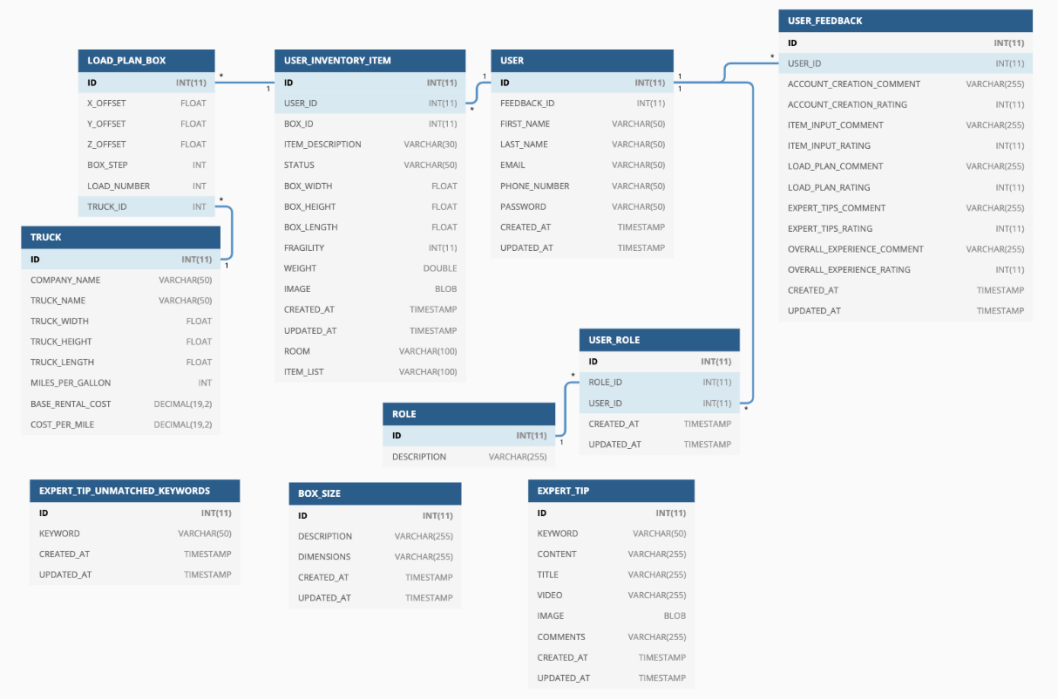
3.1.3.3. Expert Tips Algorithm (Lance)

The algorithm shall take in keywords based on the end users input for box content description during the move inventory phase (3.1.1.3.1). The box content description is used as the keyword to index (3.1.5.3) expert tips saved in the database. The algorithm will then return title, article, photo, and video for the expert tip.
The system shall provide the functionality according to this diagram:



3.1.4. Database (Lance): The relational database shall store all persistent application data. The following functional requirements shall be met:

1. The database shall be accessed and updated by requests from the web API.
2. The database shall have the following tables to store information regarding:



3. The database shall encrypt all data being exchanged with the web API.

3.1.4.2. ~~User Credentials: (Byron)~~

~~The system shall provide a mechanism by which the user credentials can be persisted and retrieved at any time for the purposes of authentication. The persisted storage shall:~~

~~1. Store the credentials for the user so that the username for one user cannot be used for another~~

~~2. Store the password in relation to the username such that when given the username, the password can be determined~~

~~3. Protect the password from individuals with database access by hashing it to obscure the password contents~~

~~4. The storage shall record the following regarding login attempts~~

~~A. When the last login attempt was~~

~~B. Whether the attempt was successful~~

~~C. The quantity of unsuccessful login attempts within the configurable time window~~

~~5. Store the state of the user account (locked, unlocked)~~

3.1.4.3. Move Inventory Storage (Chris)

The system shall provide a mechanism by which the user inventory will persist and be accessible anytime by the user. The inventory storage shall contain:

Box Attribute	Attribute Description
User ID	Integer value that is the unique ID for users contained within the USER table
Box Description	25-character string created by user to describe a box based on contents
Dimensions	Length, width, height in inches of box
Fragility	Integer value from 1-5 for how fragile the items within a box are (5 being most fragile)
Weight	Integer value from 1-5 for how heavy a box is (5 being heaviest)
Room	String containing the name of the room a boxes items come from or go to
Items	String containing list of the items that are packed within a box
Box ID	Integer value for every box in a user's inventory. Each unique user has Box IDs begin at 1 for the first box in their inventory and increment by 1 for each box added
ID	Each box has a unique ID that is separate from the Box ID. The first box in the database has an ID of 1 and each new box add no matter the user increments by 1. This means every box should be accessible at any time by this ID
Created At	This is a field that automatically saves the time when a box is added to the database and saves the value
Updated At	This is a field that automatically saves the time when a box is updated in the database and saves the value

1. The storage shall discern one user's inventory from another by the user ID, linking each item to the user that added it.

3.1.5. Web API

3.1.5.1. Authentication (Byron)

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. The web API shall:

1. Provide an interface method to establish the identity of the user as described in the authentication section for the Android client 3.1.1.1.
2. Require that on each request that the caller of the API uses Basic Authentication so that the identity of the user can be established

3.1.5.2. ~~Password Reset (Byron)~~

~~The web API shall receive a request to reset the password from the android application. It shall accomplish this in two different phases~~

~~3.1.5.2.1. Generating a token~~

~~The system shall use a token to establish that the user requesting the password is the owner of the email address on record for the user. When receiving a request for resetting it shall:~~

~~1. Confirm that the email address requested to reset the password exists as a username in the database~~

~~2. Generate a random token for the reset~~

~~3. Send the random token to the email address for the user~~

~~4. When successful at matching an email address:~~

~~A. Return a success response~~

~~5. When unsuccessful at matching~~

~~A. Return an error response~~

~~3.1.5.2.2. Resetting the password~~

~~The system shall use the token earlier to confirm the identity of the user and allow the password to change. It shall:~~

~~1. Allow the user to send the request for a new password containing:~~

~~A. The token supplied by the user~~

~~B. The new password~~

~~2. Match the user supplied token to the one generated by the system~~

~~3. When the match is successful~~

~~A. Apply the requested password~~

~~B. Return a success response to the android application~~

~~4. When the match is unsuccessful~~

A. ~~Return an error back to the android application~~.

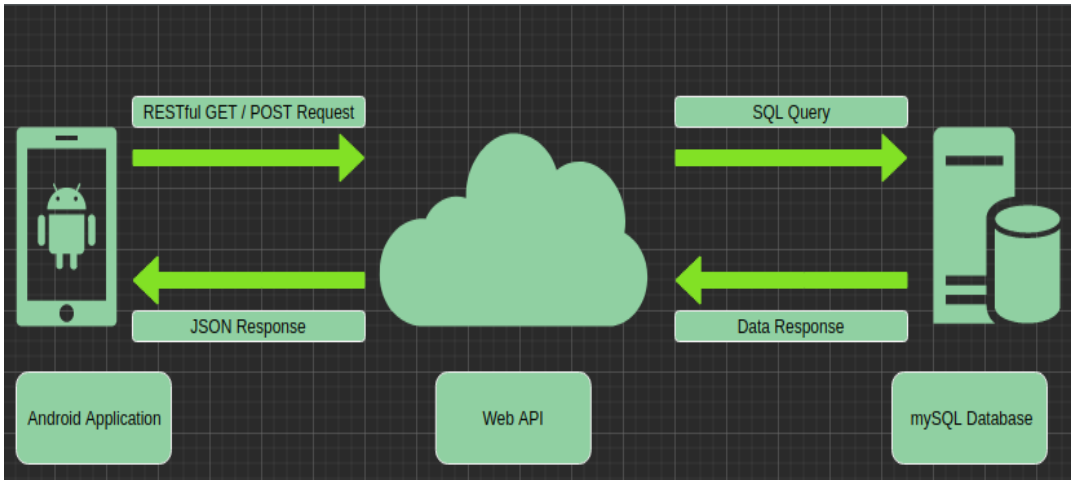
3.1.5.3. Expert Tips Indexer (Paul)

The web API shall provide a mechanism for indexing the mySQL database that contains the Expert Tips. The indexer shall:

1. Generate an index
   A. Query the mySQL database for all expert tip records
   B. Create new record for all records retrieved from the query
2. Accept keyword from web API and search index
   A. If match is found the expert article will be returned to web API

3.1.5.4. Services (Paul)

The web API shall provide various services for handling RESTful requests made from the Android application.



3.1.5.4.1. Data Services

The various services provided by the web API shall allow the user to create a RESTful GET or POST request. Depending on the service, a parameter shall be passed from the Android application to the web API.

Service Name	GET Purpose	GET parameters	POST Purpose	POST parameters
Users	Retrieve User account for authentication	Username, password	Add new user account	Username, password
Expert Tips	Retrieve expert tip from indexer (3.1.5.3)	Keyword	NA	NA
Inventory	Retrieve user inventory	User ID	Add new user inventory	User ID, inventory

Box Sizes	Retrieve box sizes	NA	Add new box size	Box Size
Load Plan	Retrieve user load plan	User ID	Add new user load plan	User ID, load plan

3.2. Performance Requirements

3.2.1. App Load Time (Paul)

In order to be useful, the application load time cannot take up considerable time. If the application is slow to load the user is unable to use any of its features. This will block them from making any progress on their move and makes the use of Load.In unappealing. The following requirement must be met:

3.2.1.1. The application load time shall not take more than 5 seconds from the user tapping the icon to open and prompt the user for their account information.

3.2.2. General Action Response Time (Lance)

3.2.3. Photogrammetry

3.2.3.1. In order to generate meaningful Load Plans downstream, the data collected via photogrammetry when entering boxes into the system must be reasonably accurate. Thus, the following requirement must be met:

3.2.3.1.1. The dimensions gathered by this algorithm shall be accurate down to 1/8”

3.2.4. Load Plan Generation (Jason):

In order to be useful, the Load Plan generation algorithm cannot take up a large amount of time, during which it would be blocking the user from making progress on their move, and ultimately making Load.In an unvaluable tool. The following requirement must be met:

3.2.4.1. The Load Plan generation algorithm shall not take more than one minute to calculate a truck load of items

3.2.5. Android Compatibility (Lance)

The system shall run on Android platform version 4.4 KitKat to be able to reach 98.1% (Android Studio Data) of devices running Android.

3.3. Assumptions and Constraints

3.3.1. Items Larger than Truck (Chris)

The system shall assume that items large than the designated truck will not be part of the load plan. The system shall display a warning message should an item larger than the truck be added to the inventory.

~~3.3.2. Extremely Heavy Objects (Chris)~~

~~The system shall assume objects that are above a certain weight are too heavy to be moved by the truck. The system shall display a warning message should an item be too heavy that it will not be included in the Load Plan. If an item is heavy~~

~~but is within the weight limits the system shall load according to the Load Plan algorithm.~~

3.3.3. Dimensions of a Truck

3.3.4. Rotation of items

3.4. Non-Functional Requirements

3.4.1. **Server Setup (Greg)**

Load.In shall be hosted on a virtual machine running Ubuntu Server 16.04 and Docker. This server is provided by the ODU Computer Science department. The server shall allow for public access to the Web API from the Android Application.

3.4.2. Containers (Greg)

The Database (3.1.4) and Web API (3.1.5) shall run inside of Docker containers. These containers will provide environmental stability when moving the Database and Web API from development machines to the Server (3.4.1). These containers shall run on the Server to allow for public access to the Web API from the Android application.

3.4.3. Security

3.4.3.1. ~~Encryption (Chris)~~

~~The system shall encrypt sensitive information when it is added to the database. The system shall do this through hash functions that alter data to alter and secure it when being added to the database.~~

3.4.3.2. Data in transit (Byron):

The system shall provide a mechanism to encrypt the data during communication such that it cannot be intercepted by third parties who may be eavesdropping on the communication. It shall:

1. Encrypt the communication channel between the android client and the web API using SSL

3.4.3.3. Authorization (Chris)

The system shall use user ID to determine authorization for different activities and information within the Load.In application. The system shall use user ID to restrict:

1. Inventory to only items added by the user
2. Load plan to only the plan generated for a specific user
3. Account information for a specific user
4. Edited items to be linked to the specific user

3.4.3.4. Public Facing Resources (Greg)

The Web API (3.1.5) shall be the only Public Facing Resource for Load.In. Traffic to and from the Web API shall be encrypted for security purposes. The Web API shall authenticate users before allowing access to the Web API's functions.

3.4.4. Maintainability (Lance)

The system shall have a scheduled maintenance plan to keep application performance at optional levels. The schedule maintenance plan shall perform the following:

1. Maintain project documentation when changes are made.
2. Modify project code by correcting any errors or bugs.
3. Purge old data out of the database and move it to cold storage.

3.4.5. Reliability (Byron)

The system shall provide a reasonable amount of reliability for the purposes of testing and demoing the prototype. It shall:

1. Maintain a 75% availability rating
2. Be able to recover from a catastrophic event in less than 1 day
3. Not be required to retain test or prototype data in the event of a disaster
4. Make no guarantee that any data submitted to the prototype will be retained for any definite length of period

Appendix