

Diving Denizens

A Two-Sided Marketplace App

Buy and sell used diving gear in Aus



Photo source: unsplash.com

About and why?

For people to sell used underwater diving equipment, there are usually two online places I have anecdotally discovered that people sell online:

- Social media platforms - like Facebook Marketplace.
- Dedicated marketplace websites - like Gumtree.

The key advantage of my app will be primarily around its centric business model to **exclusively list only diving equipment.**

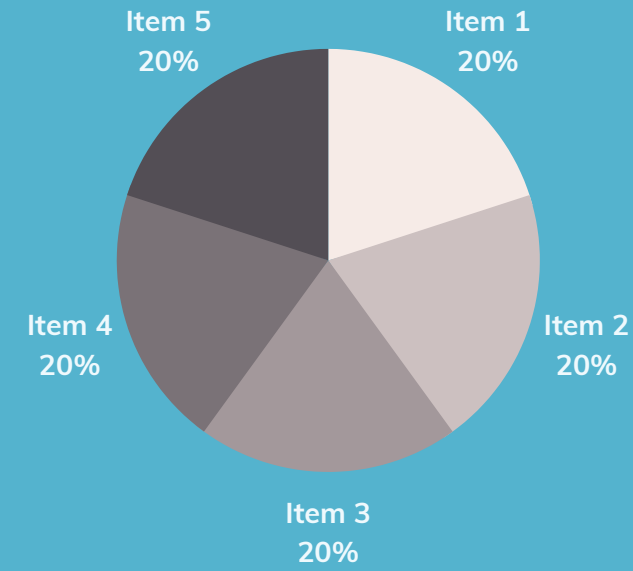
This means that the **buyer** can easily **filter and search by the countless sub-categories of diving equipment** to find what they need. This is especially important when there is such a relatively huge range of equipment types that is required for the average recreational scuba diver, especially.

Additionally, as a **seller**, a seller can opt to list multiple items individually, as it's quite common to choose to sell 'bundles' of gear, but the buyer may often want to only purchase a specific gear piece out of that bundle, which would normally mean the seller has to then modify their original listing with the now sold gear piece gone. The app will encourage the seller to list items individually as separate listings, as the seller will be able **to specify a 'discount' to be applied to an order if the buyer purchases 2 or more items from the seller in the one transaction.**



User Stories

<https://trello.com/b/KAQ2BD78/diving-denizens-user-stories>



INFO (DO NOT MODIFY) ...

All list titles are in the format of "As a <list title>, I..." For example, "As a visitor (i.e. not signed in), I" followed by either the word "MUST" in each card below, OR "WOULD LIKE TO" in each card below. Followed by details of the users' expectations.

"MUST" means what the application must deliver to the user and provide for the user for application MVP. (RED LABELED CARDS)

"WOULD LIKE TO" means what the application should or optionally deliver to the user and provide for the user for a greater user experience and for more product market competitiveness. (YELLOW LABELLED CARDS)

Only the "MUST" cards will need to be fulfilled for the application MVP.

+ Add a card

As a visitor user (i.e. not signed in) ...

MUST: Be directed to the landing page on root visit, and able to view the landing page without being signed in.

MUST: Be able to view all listings of items available for sale, but unable to purchase or negotiate price on an item without being signed in.

MUST: Not be able to sign into anyone else's account.

MUST: Only be able to securely sign in to my account if I have already created an account.

MUST: Be able to securely create a new account for myself, specifying my email address and valid password, as well as enter my contact and address details for shipping and delivery as a potential buyer.

MUST: Not be able to see personal details of others' accounts.

MUST: Be able to navigate via links on every webpage, back and

+ Add a card

As an admin role user ...

MUST: Be able to specify a listing fee that gets applied to all item listings for sale, and is automatically deducted from the seller's sale once an item is sold.

MUST: Be able to specify a delivery fee that gets applied to all sold items that require delivery/postage in Australia.

MUST: Be able to remove any item listings that violate Terms and Conditions or are otherwise inappropriate.

MUST: Be able to disable user accounts that violate Terms and Conditions or are otherwise inappropriate.

MUST: Be able to give any normal role users admin role access control as my discretion. (Role-based Access Control).

WOULD LIKE TO: Prevent and/or monitor for bot activity creating fake or spam user accounts or item listings.

+ Add a card

As a normal role user (in addition to all the 'visitor user' cards) ...

MUST: Be able to conduct all transactions securely, with strong privacy and confidentiality ensured.

MUST: As a buyer, be able to outright buy an item for sale, specify my billing/shipping details, choose delivery or pick-up, and make immediate payment using my chosen payment method.

MUST: As a buyer, be able to offer a price to the seller on an item (if enabled by the seller as 'negotiable'), specify my billing/shipping details, choose delivery or pick-up, and wait for acceptance or decline on the offer from the seller.

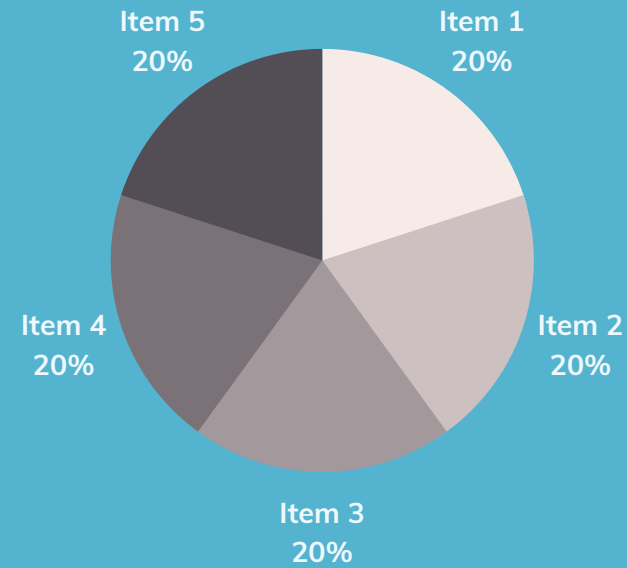
MUST: As a buyer, be able to make payment using my chosen payment method on an item that I made an offer on and was accepted by the seller.

MUST: As a seller, be able to make a listing of an item for sale, and specify important details, including photos, item condition, and description.

+ Add a card

MVP Implementation Plan

<https://trello.com/b/dduTSMRI/diving-denizens-mvp-implementation-plan>



A screenshot of a Trello board with the following structure:

- INFO (DO NOT MODIFY)**
 - All the following lists and cards (excluding 'Documentation') refers to the project plan to satisfy all the "MUST" requirements in the User Stories cards (LINK IN DESCRIPTION) for this app to reach MVP.
 - Trivial tasks such as generating the Rails models, views, and controllers, aren't included in any of the following lists, as that's just assumed as a 'To Do' requirement.
 - I will use a Domain Driven Development (DDD) approach without any real Test Driven Development (TDD) red-green-refactor cycle implementation. As such, manual testing will be conducted throughout completion of key, self-identified app functionalities, the front-end and back-end will be mostly be completed simultaneously. As such, ALL cards, including documentation, in all the following lists will have the inherit DUE DATE of 10/07/2022.
 - Note: Most of the 'To Do - Front-End' card activities require back-end configuration as well.
- To Do: Documentation**
 - Take screenshots of this Trello board every day that changes or progress are made.
 - README - Identification of the problem.
 - README - Why it is a problem that needs to be solved.
 - README - Link to deployed app.
 - README - Link to GitHub repo.
 - README - Description of marketplace app (SEE DESCRIPTION).
 - README - User Stories for the app.
 - README - Wireframes for the app.
 - README - ERD for the app.
 - README - Explanation of the high-level components in the app.
 - README - Detail of any third party services used in the app.
 - README - Description of projects models in terms of relationships.
- To Do - Front-End (UI/UX)**
 - Add detailed comments to the views, where appropriate.
 - Add Bootstrap and use it throughout for styling, customising where needed.
 - Add custom CSS.
 - Implement Trix editor for ActionText rich text for use of the seller to create their item listings.
 - For any data entered on any of a form's fields, check its validity server-side and send a UI message/alert to the user on whether or not form submission was successful.
 - Make the root route the website home page/landing page. Accessible for all users, signed in or not.
 - Add a view for all listings of items available, that can be filtered by category/categories.
 - A view with a form for a new user to enter their personal details, along with username and password, when creating a new account.
 - A view that shows a seller's rating and their reviews.
- To Do - Back-End (incl. Database)**
 - Add detailed comments in all custom methods and classes, where appropriate.
 - Implement Devise for user authentication.
 - Configure Devise minimum password complexity requirements.
 - Configure role-based access control (RBAC) for two roles - admin and normal. All new user accounts are to be assigned to the 'normal' role by default. Possibly a gem to handle roles.
 - Use the custom RBAC system to configure authorisation. Namely ensuring no 'normal' user can access the 'admin' user-only pages. Principle of 'least-privileged access'.
 - Point models and database.yml to PostgreSQL including specifying username and password credentials for both development and production environments.
 - Create GitHub repo with default base Rails structure.
 - Set up CI pipeline for Heroku to GitHub repo.
- To Do - Testing**
 - Draft test cases to use for manual testing, specifically testing for app interaction and response on edge and corner case data, including invalid data.
 - Appropriate Manual Testing by myself throughout all key feature implementations. This will be flagged as 'Done' when all MVP functionality and UI is deemed working.
- Doing**
- Done**

ERD Draft

