Software Project Proposal

Contents

Aims/Objectives

Our aim is to find areas where we can increase Usability, Accessibility and Automaticity in cataloguing applications, as well as lure older companies into the digital age of cataloguing. Our objective is to solve issues that current cataloguing applications have and add a solution to solve for the reason why some older establish companies are not making use of digital cataloguing applications. Reason for our aims and objectives is because legacy or new companies who have not yet integrated digitally sit at a disadvantage amongst companies who have, which effects company's success in an unnecessarily negative way.

Planning

Proposed Idea

A web-based product catalogue application. A simplistic catalogue creation application that follows the culturally accepted display layout that popular online shops use. The application main feature would include tools to populate their catalogue with rich product information quickly and easily from content suppliers such as Icecat. The catalogue would be tailored to the company's brand and shared via a URL.

Aims and objectives

To create a web-based catalogue creator application, which is simple enough for non tech savvy users. The app should tackle some of the issues current applications have, in particular:

- Cost to smaller companies and new users, we want to utilize different income streams, to relieve the burden of cost from the users.
- Ease of product listing and updating, as this is a big issue in cataloguing for both digitally involved companies and none digitally involved companies.

- Insufficient and missing product information for customers, we want to create a method in which companies can present all their products details without the time it takes to do so being increased.
- Complicated layouts, we want to steer away from custom layouts that customers aren't familiar with and promote template designs that are culturally accepted layouts from apps of similar purpose.

SWOT Analysis

Strengths:

- Auto Populating: Having an application that can automatically draft a product page with all its specs and details. This saves a lot of time in creating large catalogues, or catalogues that change frequently.
- Runs independently from a user's current ERP, this is a strength when the user can't redesign their entire ERP or website.
- User beginner friendly: A very simplistic and basic application, easy for non tech savvy users.
- Constant Design: Each product is displayed in the same manner within a single style template, which makes it easy for viewers to find what they are looking for.
- Extension: A chrome extension could further enhance the applications usability, having an extension that when viewing the content suppliers, a quick add button to add to their catalogue.

Weaknesses:

- What could be lacking, things that we might not do very well
- We could lack skills in a certain area, we could look at 'upskills'.
- Product Range: This would be limited to the content supplier's database, if a product were in the database it would need to be added before including in the catalogue.
- Features: Existing catalogue creators offer many custom design features which users can adjust to be exactly how they want, which won't be in this app.

Opportunities:

- Here look at the other catalogue makers and see what other stuff they are doing with their apps.
- Multiple Content suppliers: Creating a search feature that can search multiple content suppliers if one supplier doesn't have the product registered.
- Analytics: The possibility of capturing sessions from views and present this in an analytics page for the user to see who is looking at what.
- Reviews: The option of including reviews from a review source or having the option of the viewers to review the products within the catalogue.
- Sales: Creating the option of viewers to add products to a cart or quote request directly from the catalogue.
- ERP Integration: Having the ability to integrate the catalogue with the users ERP system to gather and present stock levels, as well as pricing.

Threats:

- Look at the weaknesses, threats can hinder from them.

- 3rd Party sources: If a supplier stops their service, as the content supplier decides to not provide their API as an open source.
- Down Times: If our server crashes or needs to be updated, this would affect all the company's catalogues, by not being able to be viewed.
- Existing applications: Fully featured existing applications can implement the autocomplete functionality in their apps, which would make our app the underdog.

Specification

Within this application we will be including:

- User account creation and login system. We will be utilizing a basic database that will hold each user's login data with passwords hashed. As our app security isn't a focus, we will at least protect the user's passwords.

- An interface for the user to create a personalized theme and add products to their catalogue. We will create a theme that is a part of their catalogue that is personalized for each user, as a user logo and colours. As each company would need to be identifiable through their catalogue.

- A categorized index page for the catalogue navigation, automatically created. We will gather the category tag directly from the content supplier. This would help navigating the catalogue for customer users.

- SQL lite databases storing login information as well as data involved in the user's catalogue content. We aren't going to focus to much on database design in this section though will utilize a few tables to achieve our goal.

- API integration with a content supplier, depending on how we deliver the project and if we struggle to fully integrate the API we might need to pre fabricate a database with content. We will create a free subscription to the content supplier's API.

- We will include one or two additional freemium design features that could be Included. This will be done to create an income stream for a free application.

We will not be including:

Multiple content suppliers, we will just be utilizing one content supplier

- This would be beneficial as some content suppliers will have products that other content suppliers don't have.
- Cost of implication won't be that great, though will further complicate our application.

Custom products, that users can include products that are not on icecat.

- Including a feature that users can include custom products that might not be registered products or aren't on content supplier's APIs is a necessity. Though this is already available on other applications and implementing this doesn't help our objectives.
- Cost of implication will be great, as this is an entire feature on its own.

A cart and a method of purchasing through the application

- Including this feature would be an added benefit to the application and could be integrated as a freemium feature.
- Cost implications would be high.

Site analytics

- This feature is available in some existing applications and could benefit company users in determining their catalogues success. Would be a good idea for a freemium feature.
- Cost implications won't be too high to gather the data, though the cost to implement a dashboard and database would be high.

Data normalization

- We won't be attempting to restructure the content supplied beyond a method that works. This is a necessity for company users, to assure content is all useful.
- Cost implications would be very high.

Managing complex products

- We won't be attempting to create features for kits and assemblies. This is a necessity for company users, to further customize their products.
- Cost implications won't be too high here, though too high for this project.

Integrating payment methods

- We won't be integrating the freemium pay methods, just a basic method of including the features.
- This would benefit the application in creating an income stream.
- Cost implications won't be high, though unnecessary for this project.

Searchable products and NLP processes

- This is a feature that would greatly benefit the customer users in navigating the catalogue.
- Cost implications would be very high here, having a search bar that isn't fully optimal is annoying, and having a search bar that is, is costly.

Scope

Our focus is to include features that existing applications don't have opposed to including already available features. Within this scope we will briefly indicate how we will deliver each, as well as why we included a deliverable if not yet noted. Within the non- included deliverables, I will state potential beneficial future functionality and their cost Implications.

Development Activities:

- Auto Populating: Having an application that can automatically draft a product page with all its specs and details. This saves a lot of time in creating large catalogues, or catalogues that change frequently.
- Runs independently from a user’s current ERP, this is a strength when the user can’t redesign their entire ERP or website.
- User beginner friendly: A very simplistic and basic application, easy for non tech savvy users.

- Constant Design: Each product is displayed in the same manner within a single style template, which makes it easy for viewers to find what they are looking for.
- Extension: A chrome extension could further enhance the applications usability, having an extension that when viewing the content suppliers, a quick add button to add to their catalogue.

Limitations:

- What could be lacking, things that we might not do very well
- We could lack skills in a certain area, we could look at 'upskills'.
- Product Range: This would be limited to the content supplier's database, if a product were in the database it would need to be added before including in the catalogue.
- Features: Existing catalogue creators offer many custom design features which users can adjust to be exactly how they want, which won't be in this app.

Requirements

Legacy or new companies who have not yet integrated digitally sit at a disadvantage amongst companies who have, which effects company's success in an unnecessarily negative way. (These companies will be the stakeholders).

A categorized index page for the catalogue navigation, automatically created. We will gather the category tag directly from the content supplier. This would help navigating the catalogue for customer users.

They will have an interface for the user to create a personalized theme and add products to their catalogue. We will create a theme that is a part of their catalogue that is personalized for each user, as a user logo and colours. As each company would need to be identifiable through their catalogue. User account creation and login system.

An interface for the user to create a personalized theme and add products to their catalogue. We will create a theme that is a part of their catalogue that is personalized for each user, as a user logo and colours. As each company would need to be identifiable through their catalogue.

Possible hidden stakeholders

- Retailers, in the case the business is a wholesaler, retailers could use the data for their cataloguing. As the app would be a reference-based app, and not copying the data from the content supplier, there shouldn't be any legal ramifications around this.
- Content suppliers are stakeholders that are aware of usage of their API though any change in their business or API access directly effects all other stakeholders.
- Usage of the applications ability to access content suppliers API in unintended methods. As users accessing the content providers API though the app for methods not intended can cause issues in the relationship between content supplier and the app.
- Server hosts. Issues with hosting companies effect the app as well as all the users, if the server goes down for maintenance, then all business user's catalogues won't be live.

Literature and Market Research

Issues companies are facing:

- Keeping product listings up to date and extending on the catalogues. Time is the main issue here, even with 5 minute for a product listing people still can't find the time to do it or do it properly. The main reason here is most companies don't have a dedicated person who handles this, they delegate it as a task to some employee.

- Normalizing supplier or third-party data. Time again here is the issue, having large amounts of data then having a single person decide what data is relevant to present. Or data that is to technical or not technical enough.

- Managing products listing across multiple sales channels. Syncing product and stock with online shops, catalogues and differently located branches and shops, companies limited on time faces these issues, as well as poor ERP (Enterprise Resource Planning) systems.

- Cost to create maintain and print. The cost associated with staff or a graphic designer to create a catalogue, publish it online and as well as a print ready version. This is a problem that has been solved with cataloguing systems though syncing product and recreating their catalogue on such system brings forth the issue of time.

Issues customers are facing:

- Companies hiding the real price of products, concealing additional costs. This is an easily solved issue by just being transparent.

- Complicated layouts, with non-consistency. Causes come from catalogue applications steering away from common practices.

- Missing product information. Not offering detailed information opens doors to customers not knowing exactly what they purchasing and greater possibilities of returns. This could be caused by the time issue or suppliers not supplying enough information.

- Too many technical specs Customers are different, some don't need to know all the technical specifications while another do. Bombarding a customer with a long list of specs can be overwhelming. This is caused by printed catalogues and catalogues not including 'view more details' tabs.

Motivations

Identifying the communities of users

- Cataloguing and shopping by catalogue is used by everyone, the groups of people that use these applications can differ from the business users and the customer users. As highly technical business users might create cataloguing presentations for customers that aren’t technical at all. It is rare that a non-technical business user would design a technical catalogue for highly technical customers, it would be more so the case that the catalogue could just be complicated.

- Catalogue creation in SMEs is most often done by the staff who are associated with sales and marketing, though these fields aren't required to be technically competent by nature, unlike the roll of a designer. In conclusion a catalogue application directed for SMEs would be more beneficial being directed to the lowest side of technical competent people.
- Customer use of cataloguing applications, whether it's a catalogue for large corporation or tiny companies should be on the same technical level. Being towards the lower technical side would be more beneficial in terms of a universal design
- The users I would like to cater too are the 'Passive users'

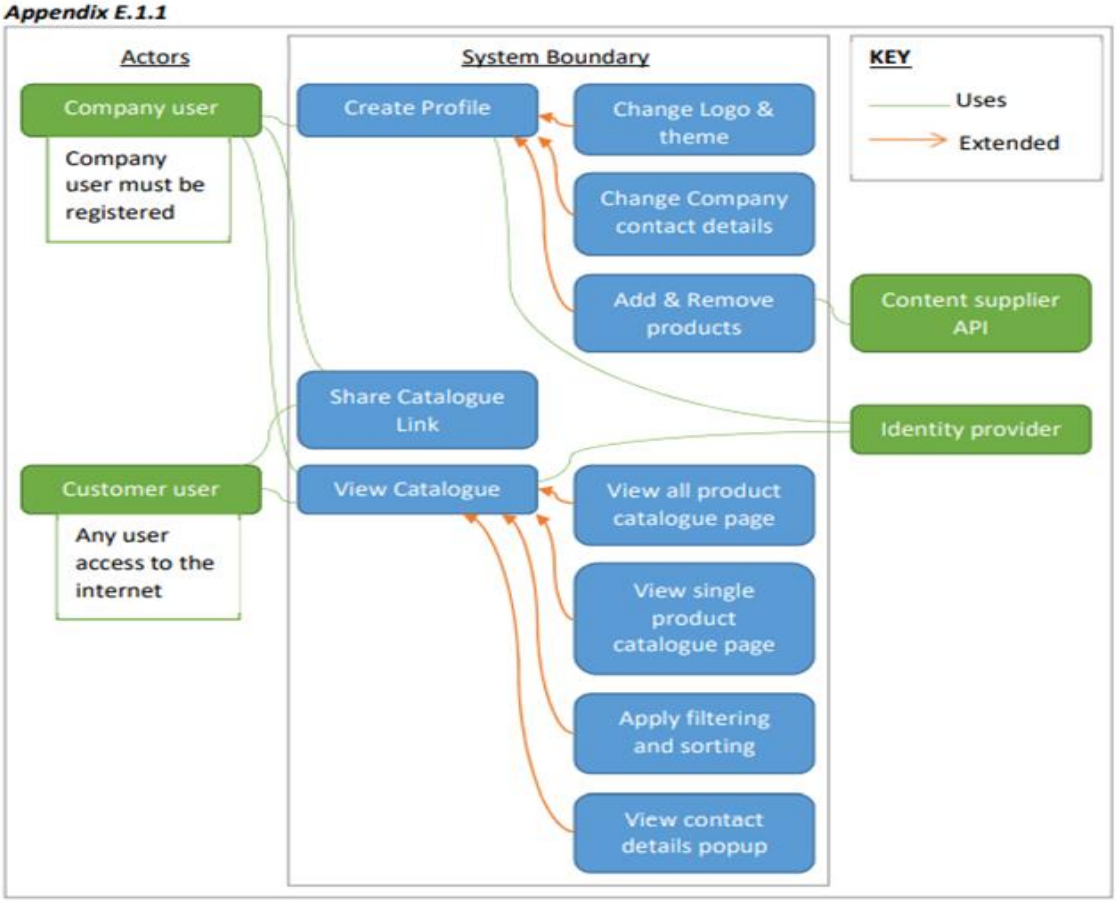
Prototyping

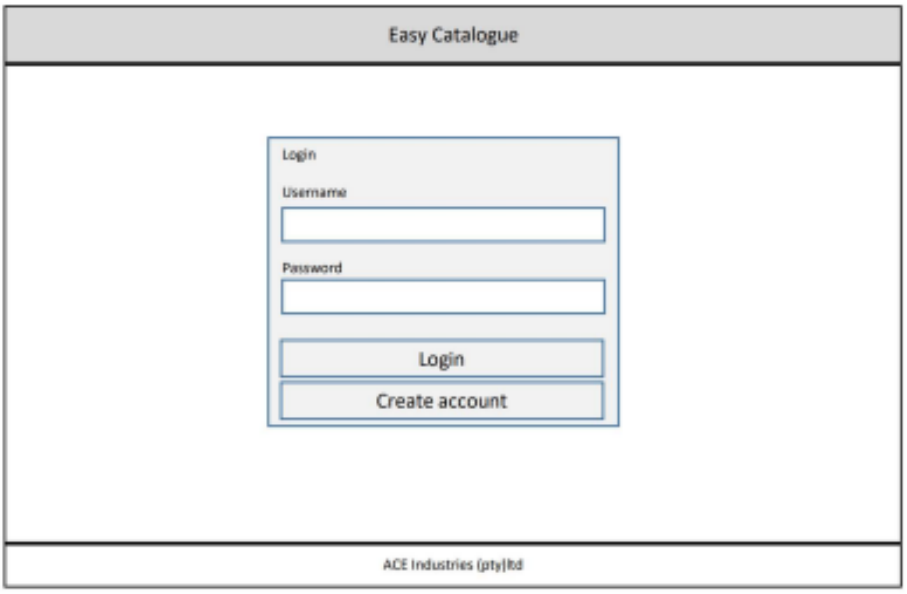
Theme customization features:

The idea is a simple login page for the company users creating the catalogue, then having a dashboard to include a logo and two-colour themes, and company details. When adding products, they will search the product on icecat, then past the URL of the product or the code in the add product box, and then our app will go get the products data and create the product page in the catalogue.

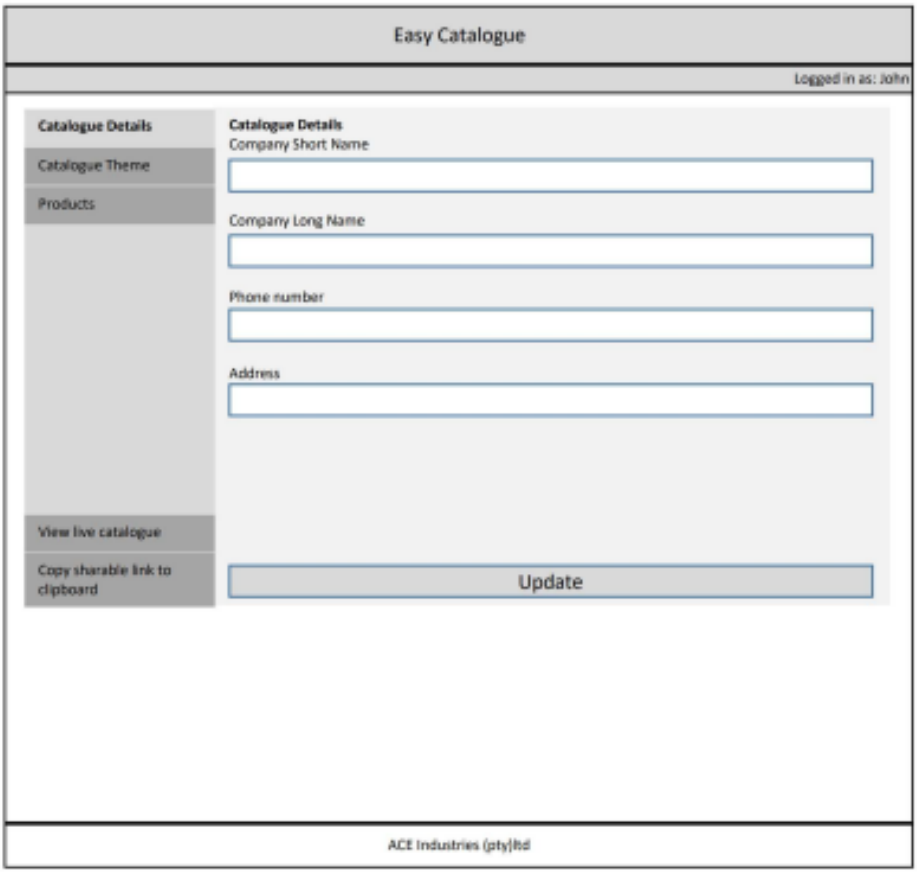
We haven't included options of like filling in extra details and customising the text, so currently it's as whatever comes from icecat is going to be in and they can't change it. We can do this differently though, though just want to keep the app simple, but we can see maybe we add boxes where they can adjust the received product data. See wireframe below:

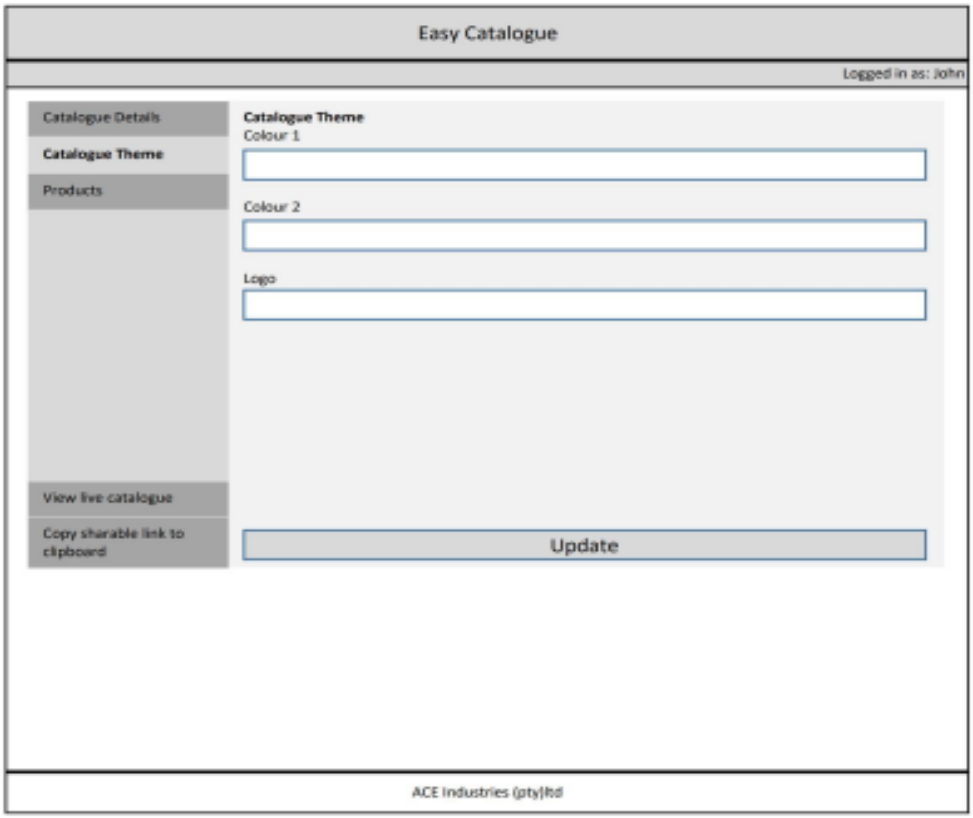
Appendix E.1.1



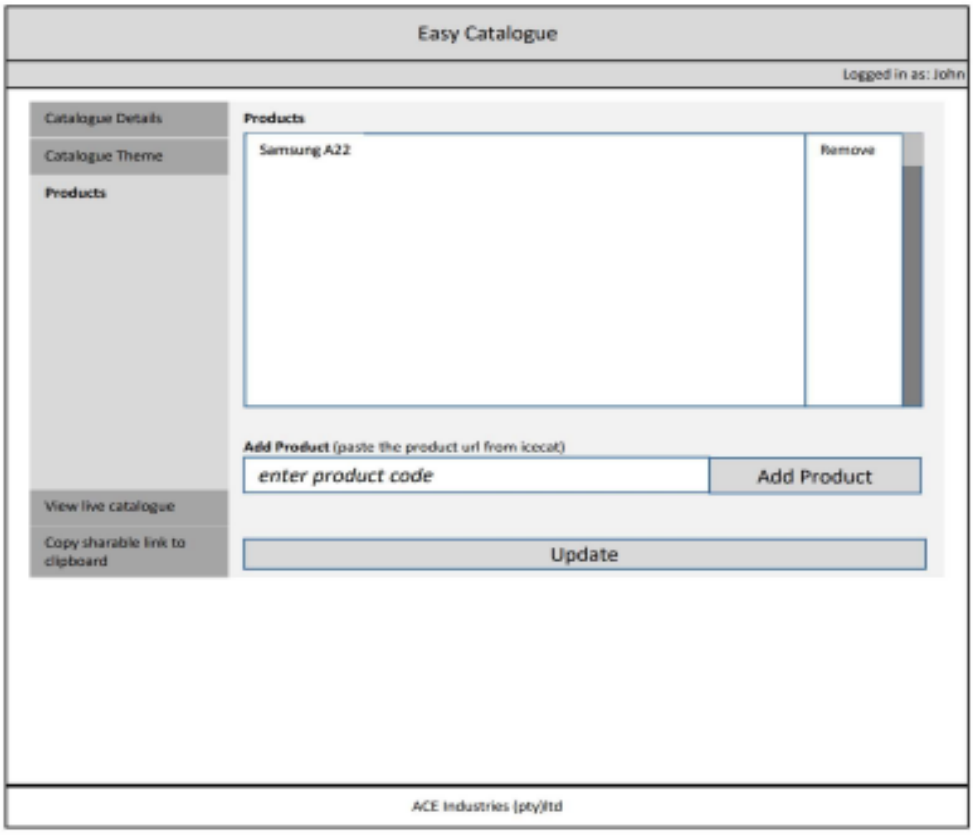

Easy Catalogue
Login
Username
Password
Login
Create account
ACE Industries (pty)ltd

Login page


Easy Catalogue
Logged in as: John
Catalogue Details
Catalogue Theme
Products
View live catalogue
Copy sharable link to clipboard
Catalogue Details
Company Short Name
Company Long Name
Phone number
Address
Update
ACE Industries (pty)ltd



Details page

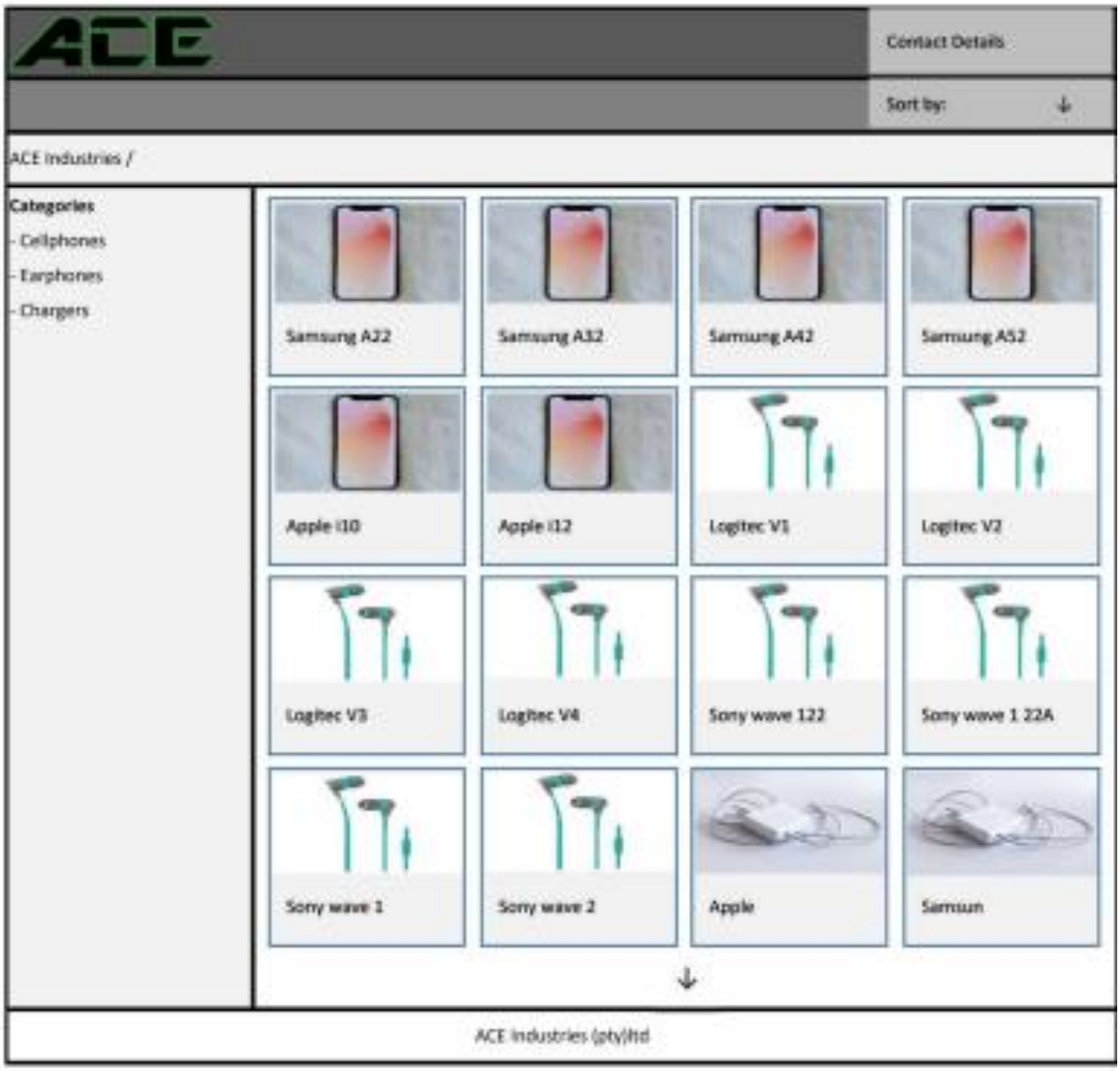


Details page

Assumption Testing

Below would be the catalogue that customers view, so will be two pages, one with all the thumbnails of the products then a page for the product.

Sorting the products, we have only included ‘category’, ‘brand’ and ‘date listed’. Though it will get very complicating if we include options like comparisons. Also having a search bar would be nice though also will be tricky to do, so we have left it out in the prototype. The format of the catalogue is like an online shop, reason here is ‘heroics’, one of the issues with catalogues is that people do weird designs that customers can’t find what they are looking for, so one of the points on this design was having a ‘culturally’ accepted layout that most people are already familiar with.



Catalogue page – cluster products

Catalogue page – single product page minimal details

Analysis/Outcomes/Evaluation

Quantitative data, from technical and non-technical users

Iterative user testing on mock-ups for business and customer users. Here we will evaluate the time that is taken to complete functions of the app, we will be able to determine how complex or complicated the app is depending on if technical users took long to do a task, then is complicated, if technical users took quick though non-technical users took long then is complex.

- Questionnaire on mock-ups for business and customer users. Here we will evaluate frustration levels of the layout, and if users wanted to do something that they weren't able to do.
- Questions such as, for 'task 1' were the buttons and steps in the correct places, or did you struggle to find them.

- Questions, was there something that you wanted to do that you could figure out how to do? Qualitative data, from technical and non-technical users.

Iterative user testing

- Questions such as, for 'task 1' were the buttons and steps in the correct places, or did you struggle to find them.
- Questions, was there something that you wanted to do that you could figure out how to do?
- Questionnaire on mock-ups for customer users. Here we will gather information on the customer's experience.
- Questions such as, was there to much information? When looking for further information did they find out everything they needed?

How we going to present our findings

- Stake holders effect and risk
- Quotative findings
- Stake holders
- Summary of qualitative data