Comp9323 Project 6 Brief

Summary

Project Description

Charity op-shops have long been selling second-hand goods to raise funds for their respective charities. But like most physical stores, charity op-shops have struggled to operate and fundraise in the COVID19 times, due to social distancing and lockdowns.

As such, charity op-shops are wanting to digitalise. And they would like your help with building a cataloguing solution for listing all the unique books and clothes that are donated to their stores.

In addition to building a React Native cross-platform mobile interface, this project would involve:

- The creation of a cloud-based relational database for managing listing & user access;
- Ecommerce Integration to list items on eBay, Amazon and beyond for greater market exposure;
- Improvement of the book cataloguing efficiency by auto-generating the book details. This could be done via barcode (ISBN) scanning & integration with various book APIs.

Possible extension tasks include creating an offline mode to queue listing and deleting requests, creating a web-store front, and integrating with a delivery logistic management system.

Cross-collaboration

Since there are 2 teams working on CircEx in the 2020 T2 COMP9323 course, the CircEx team is planning to split up the development roadmap.

Each team will be developing software that could exist as a standalone product. But by giving different focuses to each team, CircEx would combine the different features developed by the 2 COMP9323 teams to create a more comprehensive cataloguing software for the op-shops.

> To help combine the software from the 2 teams, CircEx is recommending specific frameworks and languages to be used for the front-end and back-end of the cross-platform mobile development. Specifically, CircEx is recommending the use of React Native for the frontend, and python-flask for the backend.
>
> While there is no hard requirement for the frameworks, CircEx asks both Project#6 teams to agree on using the same frameworks, such that the codes are compatible when CircEx combines them.

Focuses

There are 2 possible focuses for the teams. Team 1 could build the app for cataloguing books, while team 2 builds the app for cataloguing clothes.

Books

Team 1 will be redeveloping CircEx's book cataloguing Android MVP in a cross-platform framework (like React Native), and building on it as well.

Specifically, this book cataloguing app allows op shop workers to list & checkout **books** online via barcode (ISBN) scanning. Book details could be autogenerated from API integration with ISBN databases. And the app would list the items on the op-shops' own sites or even on other e-commerce sites like eBay as well.

When charity op-shops asked to accelerate the cataloguing software development, CircEx started by repurposing code from past Android projects to quickly build an MVP. But due to the varying device's requirements from the op-shop community, the op-shops would like your help to redevelop CircEx's MVP in a cross-platform framework.

[A video link for the existing android MVP will be included shortly]

Clothes

With cataloguing clothing, there is no standardised ISBN-like system to index & lookup the details and prices of clothes. But with AI, the description of the clothes could be auto-generated and the prices could be estimated, helping to speed up the cataloguing process. (See below for more details)

As such, CircEx hopes to assign the group with stronger AI development abilities as team 2, who will be given a clothes-cataloguing focus and an optional AI component in their extension tasks.

Building MVPs

From the Lean Startup methodology, the term minimal-viable-products (MVPs) describes a simplified version of a product with only the essential features. While additional features could improve the user experience, an MVP itself is still usable.

In CircEx's opinion, the essential features of the cataloguing app include an eCommerce API integration for storing and listing items, and some basic auto-description functionality to make the cataloguing process sufficiently quick. Along with the front-end, an app with these two features would constitute the MVP.

Logistically, in the first half of T2, Project#6 teams will be focusing on building these MVP features. Subsequently, in the second half of T2, while the Project#6 teams start developing the non-essential features, the CircEx business team can start field-testing an MVP. This quicker rollout helps charity op-shops to get back on their feet quicker. By field-testing alongside development, the CircEx business team can also give feedback and learnings to the Project#6 teams to help improve the design.

Phase 1 development - MVP

- Frontend
- Intermediate API Setup
- E-commerce Integration With One Site (e.g. eBay)
- Basic Auto-Description Functionality

Full feature Roadmap

The 2 Project#6 teams should write their code in a modular fashion for easy integration.

	Team 1 - Books	**Team 2 - Clothes**
React Native Frontend	• A Book Cataloguing Interface	• A Clothes Cataloguing Interface
	• An Administration Portal `Note: The CircEx team will provide sketches for the layout designs in the React Native frontend. And it is recommended that the 2 teams follow the designs for easy integration.`	
Intermediate Flask API backend	• A User Access Management System (From 1 Team)	
Basic Auto-Description Functionality	• ISBN Scanning & Book API Integration	• An Adaptive Category & Description-Tag Drop-down List
Ecommerce Integration	• EBay Integration	• Amazon Integration
	Additional • Integration with Other E-commerce Sites (e.g. Gumtree, Facebook Marketplace, Depop)	
SQL Setup	`Note: Each team will set up its own AWS databases. But CircEx will also help liaise between the 2 teams, and help standardise the SQL design & function structures for easy integration.`	
Advance Auto-Description Functionality	N.A.	• Clothes Auto-Description AI `#AiExtensionTask`
Auto-pricing	• Competitor Price Search (Using API & bots)	• AI Pricing Algorithm `#AiExtensionTask`
Extension Tasks	• An Offline Mode (To Queue Listing And Deleting Requests) • A Web Storefront • API Endpoints For Integration With Other Frontends • Integration With A Delivery Logistic Management System • A More Advance Auto-Pricing System For Books & Clothes	