|  |
| --- |
| Personal Project |
| Sharing App |
| Books, music, games, everything |

|  |
| --- |
| Pradeep Anumala & Parag Parate  6-9-2018 |

# Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Author | Comments |
| 9-Jun-18 | 1.0 | Parag Parate | Created |
| 10-Jun-2018 | 1.1 | Pradeep Anumala |  |

Table of Contents

[Version History 1](#_Toc516330777)

[Overview 4](#_Toc516330778)

[Idea Behind This Project 4](#_Toc516330779)

[Stakeholders 4](#_Toc516330780)

[Items Application Can Cater 5](#_Toc516330781)

[Business Model Challenges (Just a discussion point for now) 5](#_Toc516330782)

[Trust 5](#_Toc516330783)

[Pricing 6](#_Toc516330784)

[Genuinity of Product 6](#_Toc516330785)

[Risk Matrix 6](#_Toc516330786)

[Functional Requirements 7](#_Toc516330787)

[Login 7](#_Toc516330788)

[Additional User Details 7](#_Toc516330789)

[Location Co-ordinates 7](#_Toc516330790)

[Contact Number 7](#_Toc516330791)

[Email Address 7](#_Toc516330792)

[Landing Page 7](#_Toc516330793)

[Search 8](#_Toc516330794)

[By User 8](#_Toc516330795)

[By Item 8](#_Toc516330796)

[By Location 8](#_Toc516330797)

[Map Interface 8](#_Toc516330798)

[Transactions 8](#_Toc516330799)

[Putting Up Item for Sharing 8](#_Toc516330800)

[Showing Interest in Item 8](#_Toc516330801)

[Social 8](#_Toc516330802)

[Chat 8](#_Toc516330803)

[Call 8](#_Toc516330804)

[Methodology 8](#_Toc516330805)

[Use Cases 9](#_Toc516330806)

[Reference 10](#_Toc516330807)

# Overview

Share2care is an online portal that helps users exchange, sell and donate items. These items can include books, music, dresses, games and any other personal belonging. The initial version will target books. However, application will aim at a generic architecture to allow inclusion and exclusion of any number of components. This document will describe functional requirements and call on application stack. It will also speak about top level architectural design of solution but will not get into technicalities of design. That should be discussed in technical design document.

# Idea Behind This Project

There are many items that practically do not have much value once they’re utilized, like novels and fictional books, study guides that are specific to an academic year, games that are played and completed, etc. Beside such items, there can be items that are lying dormant that one can lend in exchange of some other item or at a nominal charge to someone who needs it. This application aims at providing such platform to users to put up items for exchange, donation or sale and look up for anything in particular they need.

This application can be practically useful for people with somewhat following requirements -

* Someone who wants to temporarily exchange/borrow/lend stuffs
* Someone who wants to barter one or more items in exchange of a mutually agreed good bargain from another user.
* Someone who needs an item but doesn’t want to spend too much on it.
* Someone who temporarily needs an item to get work done.
* Someone who wants to put dormant items at home to use.
* Someone who wants to donate stuff to someone deservingly needy.

# Stakeholders

Before getting on with design, it’s important to identify stake holders of this application. Identifying end users will be particularly helpful as we can step into their shoes and look for their requirements, problems they face in related area and how best application can be designed to solve those problems. Developer can also identify people in their circle that fall in this category, talk to them to refine requirements or take feedbacks and suggestions and even share the application with them to test.

List of identified stakeholders –

1. Developers
2. End Users
   1. Students who can share academic books
   2. Book readers wo love to read from hard copies
   3. Parents who can’t afford or don’t want to shell much on academic books
   4. People who are looking for a rare book that is not available in market
   5. Govt. schools who can list required books with quantity and people who want to donate in such cause.
   6. Someone who wants to sell off books they’re no more in need of. Sell it on a fixed price or auction/bargain.
   7. Selling used books might sometimes be difficult. So, user can barter them in exchange of any other listed commodity.
   8. Gated communities where trust is easily established and things can be exchanged easily.
3. Technology vendors
4. App store

Any item other than books should identify stakeholders in similar fashion.

# Items Application Can Cater

1. Books
2. Games (PC, PSx, Xbox)
3. Movies & Music (Virtually dead for hard copies)
4. Dresses? (Especially the partywear which people (read girls) don’t prefer to wear for the 2nd time)
5. Gadgets. E.g. iPod shuffle in exchange of fitness watch.
6. Camera lenses (a large number of people look for specific lens for short duration for a specific purpose photography, specially coz lenses are pretty expensive)
7. Musical instruments. (for short term for a particular show)

# Business Model Challenges (Just a discussion point for now)

## Trust

Lending any personal belonging to others can be a huge risk, especially when involved parties do not know each other. This is a huge problem that needs to be addressed before solution designing as this can form *point of failure*.

Practical examples can be drawn from business that are involved in such activities –

* Banks & Financial Institutes – who lend loans against a collateral. They also do a background check of applying candidate and evaluate their credibility using CIBIL score.

This application will involve random people putting up goods for sharing in some way. Letting end users handle trust is not a great deal for application as people would not want any kind of overhead while using an application.

So we need to identify equivalent of what these institutes do but in digital context.

Collateral Equivalent - ??

Background Check Equivalent – Social profile? Need a better factor.

CIBIL Score Equivalent – CIBIL score starts with base points for a new user and is then evaluated based on user behavior over period of time. Application can maintain user’s credibility score, but may take time to build it over period of time. This can form a base for trust over longer duration but will not be of great help for new users.

Can we draw user’s credibility from their social profile?

* Any other type of institutes in similar transaction?

**Possible Solutions**

* We can consider tagging an item with a price tag against required credibility based on its price. Only users that (Fuck! This is that episode from **Black Mirror**) have credibility equal or higher than this required score can trade.

Can people buy credibility? People can buy credibility against a predefined price-credibility scale. They can then trade items that lie within their credits. If person fails to return item to lender, lender can encash borrowers credits to recover his losses and borrower loses his credits.

Do we refund in case of successful transaction? We can, and user can retain credits but this is prone to abuse (by building credit points and duping someone in single transaction). Rather we can retain credit deposit and person lending can always be assured to getting item or his money back.

Even lender can cause a scam by claiming he did not receive back his item or it was damaged or not the same as one that was borrowed. *How do we handle this?*

* Percentage based rating: Initially every user will have 0% rating with a “new user” tag. After every transaction, the parties involved in the transaction would rate each other. There will be a questionnaire. Each question will have some weightage. Based on these answers the users will get the rating.
* Deposit approach: When a user puts an item for lending, the user mentions the hourly/daily rent and the deposit amount too. For e.g. Assume the cost of canon 1.8 f lens is 10,000 INR. The lender can charge 500 INR a day and a deposit amount of 5000 INR.
* Hybrid approach: Use both the above methods – percent based rating and deposit approach.

**Do we consider this model or let end user decide on trust?** Pros and cons are nearly listed above.

## Pricing

Pricing is left to the users.

## Genuinity of Product

It’s the buyer’s/borrower’s responsibility to check the genuinity of the product. The application do not claim any responsibility.

# Risk Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| Transaction | Challenge | Comments | Risk |
| Exchange | Need way to remove manual step of marking transaction complete | Allow end users to settle transaction outside system | Low  None? |
| Borrow/Lend | Trust. Can’t commit for a fair transaction from either or both ends | Refer to possible solutions listed above. | Risky to implement |
| Buy/Sell | Pricing, Genuinity | Let end users settle transaction. Application do not claim responsibility | Low risk |
| Donate | Genuinity of one receiving donation |  | Low risk of genuinity can be est. |

# Functional Requirements

This section outlines functional requirements in different areas of application. Focus should be on keeping manual intervention to minimal while designing solution.

## Login

Application will consider both web and mobile format. So login mechanism should have a common design to allow login with minimal effort consistently in either or both interfaces.

Web interface should allow integration with Facebook and Google authentication. Do we consider providing sign up feature? If yes, we allow bigger user base as it would include people without social accounts. But how big is that chunk? This will be at cost of not having user’s social identity (but even if we have it, what is its authenticity). I’ll suggest sticking to social account. [Open for debate]

Mobile interface should use same social accounts to allow access to users. Underlying entities would be same for both interfaces, so data should always be synchronized.

## Additional User Details

Following details in addition to social login will be needed for transactions. Should check which of these details can be captured with login.

### Location Co-ordinates

Co-ordinates will be a crucial factor for application to get user’s current location and find nearby matching searches/users. Co-ordinates in web application can be found by tracking IP address while mobile interface can get them using GPS location. Need to check how browser interface in mobile will track location (IP or GPS).

Application should have provision to capture these details and keep them dynamic until a transaction takes place. Location co-ordinates should be noted when interests are flagged or transaction is initiated.

### Contact Number

Web interface should not require contact number as we can’t make phone calls from computer. Is it needed in any case?

Mobile interface can use feature to call but will it need manual key in mobile number?

### Privacy

After signing up, the user provides his address and contact no. The mobile number should be verified by sending a verification code to the mobile. The users will also choose their preferred method of contact like email or phone? They can also choose if they want their mobile number to be displayed to other users or keep it private.

### Email Address

Email address will needed for sending notifications. Can be obtained from Gmail login. In case of Facebook?

## Landing Page

Landing page

## Search

Search

### By User

User

### By Item

Item

### By Location

Location

## Map Interface

Map

## Transactions

Transaction

### Putting Up Item for Sharing

Sharing

### Showing Interest in Item

Interest

## Social

Social

### Chat

Chat

### Call

Call

# Methodology

Agile – Scrum.

# Use Cases

# TBD

* Do users post pics of the items that they are transacting? I don’t think it’s needed for books,game dvds etc. But they are needed for

# Reference

The user would be able to logon to the portal using third party credentials like facebook,google etc

A user has 2 set of books at any point of time, those are (for time being) referred to as have\_books and want\_books. Have\_books are the books that user has and want\_books are the books the user is interested to buy/exchange. In some special cases the want\_books set can be null eg. User just want to donate his/her books.

How the process works?

1. User logs on using fb / google etc (register)
2. User provides additional information like mobile no. and communication address
3. User adds the books to have\_books set (books the user has )
4. User adds the books to want\_books set ( books the user wants )

(Here ends the user task)

1. Based on the location of the user, the system identifies with who other users the books can be exchanged.

Eg. Let there be 2 users Akshay and Pradeep who are neighbours to each residing at approx 0.5 km distant to each other. Akshay adds “Angels and Demons” in his has\_books and “The Davinci code” to the want\_books. Pradeep already has “The Davinci code” in his has\_books and “Angels and Demons” in his want\_books. Based on the co-ordinates, the system starts at Akshay’s address and moves away from the point ( as a circle ) until it identifies the match and displays the results. If there is no exact match , then based on the author,content it suggests some other relevant books.

For eg: suggests to exchange for “Java Cathy Siera” instead of “Java Head first”

1. Once the user (say user1) finds that there is a match and they can mutually exchange, the user1 can set a flag of interest. If the other user ( say user2) accepts the request, then a mail will be sent to both with the contact no.s asking them to exchange.
2. Once the physical exchange of books is done. Both the users have to edit their have\_ books and want\_books. Users would do it because they would get unnecessary requests.
3. The portal also contains user reviews of all kinds of books (similar to imdb ).

Novels to read before you die , user rating of each book on a scale of 10

Things that should be thought of:

1. Suggest the portal to a friend via facebook?

(A mail will be sent to user friend saying “ Your friend ‘abc’ has suggested you to go through this portal”)

1. Both android and web application should be in place while launching
2. Animation video that shows the process
3. Users rating each other after physical exchange of the books or after each getting mail to exchange

(Rating can be given based on: does user really have the book? Is it in a good condition? Did user respond quickly? Was the process smooth in collecting the book from him?

Rating encourages the people to keep books in good condition and keep exchange process easy.)

Desired Technical Features

1. Look and feel, easy navigation
2. Google map. The map points to the user location + 0.5 km radius. As the user zooms out, the map shows the available books in the location.
3. Dynamic typing: User can either enter ISBN or partial title string to add books to the sets

Open Technical Issues

1. What technology will be used for the development of application?

Java or python or oracle db or nosql ?

1. How the map thing would be implemented?

Modules

1. Add books
2. Lend books
3. Search
4. Reviews
5. How it works
6. My Profile
7. Peer rating

Points to consider –

1. Authorization (?)
2. Security
3. Cloud – AWS
4. Tech Stack.