System Requirements Specification

# 

# **1. Table of Contents**

[**1. Table of Contents**](#_za0xjl36jdfj) **2**

[**2. Problem Statement**](#_yihs077afgss) **4**

[**3. Overview**](#_8f5tuwc7ryo6) **5**

[3.1 Background](#_er4hetg25z9i) 5

[3.2 Overall Description](#_m91jwt6hf48z) 5

[**4. Investigation & Analysis Methodology**](#_996gs33nnz77) **7**

[4.1 System Investigation](#_juu3imlw0uvp) 7

[4.2 Analysis Methodology](#_i7rtyrdyskn6) 7

[4.2.1 Feasibility study and requirements elicitation](#_t7n07h5pm4qn) 7

[4.2.2 System analysis and requirements specification](#_dk74ppb7ftqf) 8

[4.2.2.1 Perform an analysis of the problem using object-oriented techniques](#_jh8izr65t673) 8

[4.2.2.2 Scope and Limitations](#_x9hpuykds95w) 8

[4.2.3 Object-oriented design using UML](#_bckuxmsua2x9) 8

[4.2.3.2 Use Case Diagram](#_rg6kf6pq6nnr) 11

[4.2.4 Prototyping](#_a0ukafv9zxur) 11

[**5. Constraints**](#_f2n3vodel6h) **12**

[5.1 Scalability](#_78gvwmc5xl6h) 12

[5.2 Credibility](#_6y4sp2j2qwhu) 12

[5.3 Project Schedule](#_hisdt7moya2f) 12

[**6. Operational Requirements**](#_japtv1pedhzx) **13**

[6.1 Help Desk Support](#_km5um06mdqvz) 13

[6.2 Application Services and Technical support](#_9a22ycy8bfxc) 13

[6.3 Administration Features](#_dxnh9ru35brw) 13

[6.4 System hardware fail over and routine back up](#_mxqtp7clpti0) 13

[6.5 Audit Trail](#_g2huuynwznio) 13

[**7. Functional Requirements**](#_89isvcxn5pqd) **14**

[7.1 CRUD for User account (Authentication)](#_lfn1c58j0e5) 14

[7.2 CRUD for Products (sellers)](#_54dfmtlio60i) 14

[7.3 Sort Products](#_z97trra1k170) 14

[7.4 Favorite Products](#_hcsdgwt83td0) 16

[**8. Input Requirements**](#_yzzzv7adie6t) **16**

[8.1 User account](#_3ti68li7usqm) 16

[8.2 Product sorting](#_xbfrc9rwcq9j) 16

[**9. Process Requirements**](#_c0403g2hdj15) **17**

[9.1 Database transactions](#_u7rnutjeqk1l) 17

[9.2 Data integrity](#_tx39rxq6g5tm) 17

[9.3 Data validation](#_mpk6nbhjtj4l) 17

[9.4 Performance](#_ideqboph375i) 17

[9.5 Data repository](#_ihyhmcxohtjm) 17

[**10. Output Requirements**](#_59dlnj2i4vc5) **18**

[10.1 Product purchase summary](#_d9ez9xvuthyw) 18

[10.2 Chat message](#_o3lt5ux07xdm) 18

[**11. Hardware Requirements**](#_6gpdzv3jlfkm) **19**

[11.1 Network](#_ww7dwrmkwhmp) 19

[11.2 Computers](#_5n0dciu8nvp3) 19

[11.3 Mobile Devices](#_o0c0f1kq8t4p) 19

[11.4 Production support systems](#_khh7w5b2vw9d) 19

[**12. Software Requirements**](#_ber44jf50m2m) **19**

[12.1 Client Operating Systems](#_i1vwy228ynmn) 19

[12.2 Client Application](#_3ej7p9g60zek) 19

[12.3 Network system](#_3rm0v5a6runs) 20

[12.4 Mobile Operating System](#_nlaud8v5mnbn) 20

[12.5 Licenses](#_l6nobpemmawx) 20

[**13. Deployment Requirements**](#_yqspq6mjarid) **21**

# 

# 

# 

# **2. Problem Statement**

The trend of fast fashion has led to an increase in amounts of textile waste, usually disposed of via incineration or landfill. With the rise of awareness about the environmental impact of fast fashion, thrifting - buying and selling second hand items, has become an increasingly popular means of shopping amongst Singaporeans[[1]](#footnote-0). This practice of buying second hand items is also fast emerging in electronics and interior design as well.

Therefore, MerchantDice seeks to be a one stop platform where buyers and sellers can come together and exchange goods conveniently via an online space. This reduces the need to travel to a physical thrift store in order to be able to thrift shop. Thereby allowing our users to enjoy the benefits of thrifting while still being able to enjoy the convenience of eCommerce. Lastly and most importantly, thrifting in the online space reduces the need for physical stores to be set up, thereby further reducing the carbon footprint as we can reduce the use of electricity that is needed when a physical store is set up.

# 

# **3. Overview**

## **3.1 Background**

Thrifting has many benefits, which include mitigating the environmental impacts of fast fashion. Fast fashion has led to clothes that are often in good condition being left unused, resulting in unnecessary waste production. The solution is thrifting. Thrifting has allowed people to sell their pre-loved items to others, thereby reducing the amount of unused clothing circulating the industry.

In addition, thrifting allows both the buyer and the seller to enjoy benefits. The buyer is able to earn profit from selling his or her clothes while the buyer is able to enjoy cost savings. This is because pre-loved items are usually sold at a cheaper price than brick and mortar retail shops. In other words, buyers might be able to buy branded goods in good condition at an affordable price range.

Despite its benefits, thrifting in Singapore still remains inconvenient as the number of thrift stores are still relatively low. As such, the locations of thrift stores do not incentivise thrifting to people who may not live in the area.

While, many are starting to see the allure of buying and selling pre-loved items. However, there are little known avenues for such exchanges to take place. In addition, many thrift stores are not well known to the public. Hence, the general public may not be well versed in thrift shopping habits.

## 

## **3.2 Overall Description**

MerchantDice is a web application that aims to facilitate the process of thrift shopping. In doing so, buyers and sellers can come together to exchange goods conveniently via the online space. Buyers can use the platform to look for goods that suit their needs and wants easily at a low price. And sellers can use the platform to sell pre-loved items. In addition, there is a chat function that enables all users to communicate with each other. This can be used to discuss details about the transaction or for any other purposes necessary.

# **4. Investigation & Analysis Methodology**

## **4.1 System Investigation**

A non relational database, MongoDB, is used to store product information and user information. From the web application user interface, requests are made to the backend to retrieve product information. When the user decides to make a purchase, the frontend communicates with the backend databases to ensure a smooth transaction. It will update the Products table as well as the purchase history in the User database.

If the buyer wants to view his or her purchases, he or she is able to view the transaction history. This is made by retrieving relevant information from the backend.

## **4.2 Analysis Methodology**

### **4.2.1 Feasibility study and requirements elicitation**

To ensure that the objectives of the project are met, a development and implementation team composed of people who have knowledge about thrifting and how it can be conducted in an online space will be formed. A meeting will be organized to determine the key functionalities that must be implemented in the web application in order to make thrifting an easier process.

A feasibility study will be conducted to determine which functionalities are achievable in the given timeframe. The study will also take into account the priorities of the users in defining the user requirements and deciding the key functionalities to implement.

### 

### **4.2.2 System analysis and requirements specification**

#### **4.2.2.1 Perform an analysis of the problem using object-oriented techniques**

An external view of the enterprise model will be developed using Unified Modeling Language (UML). This System Requirement Specification documents will form part of the documentation for the project.

Some desired features of the system include:

* Purchase and/or sell Products
* The ability to create and authenticate accounts
* Users able to chat with other users
* Users can modify product reviews
* Sellers can upload products to the database
* System able to retrieve and display products
* System to sort and filter products depending on user input

#### **4.2.2.2 Scope and Limitations**

Analysis methodology will involve business analysis, requirement analysis, data analysis, process analysis, (web) and application architecture:

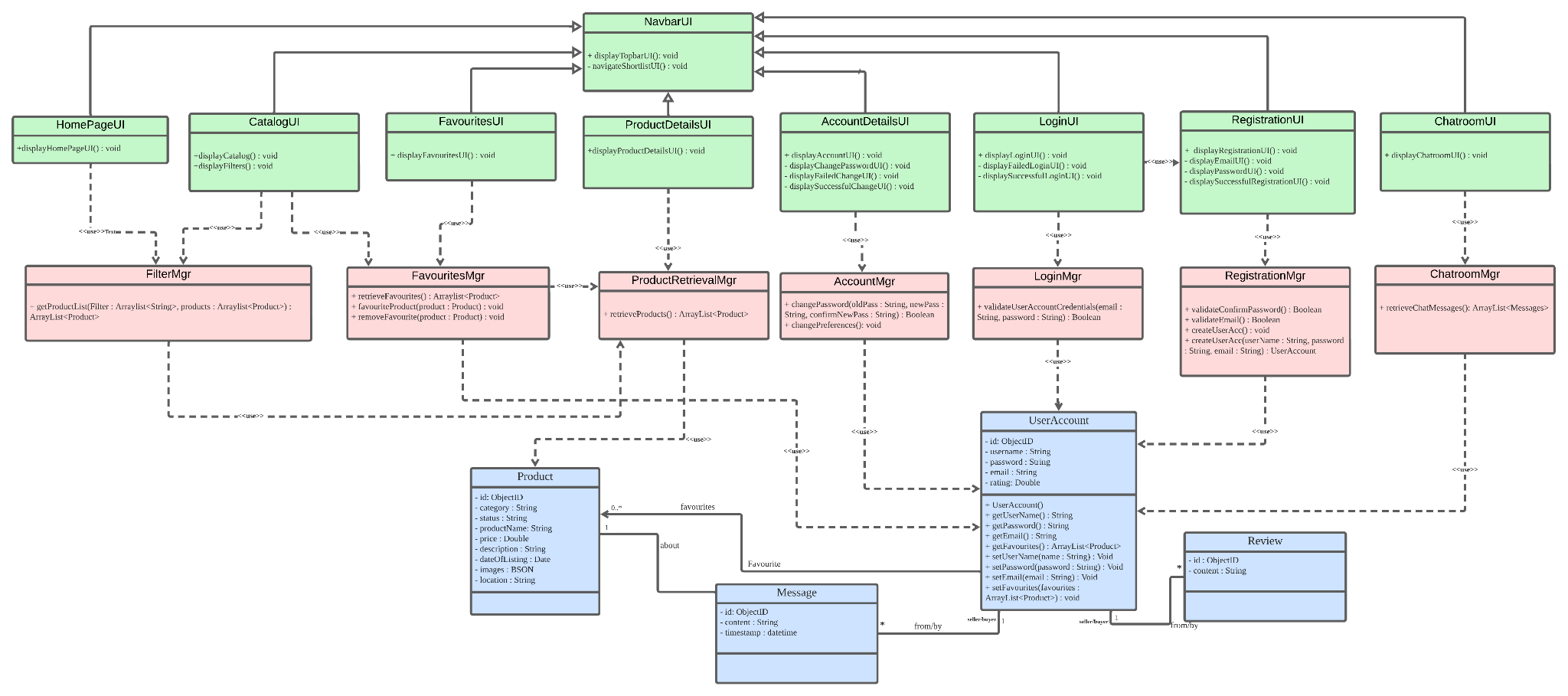
* Business analysis – State the business rules, business system interfaces, business function, business ownership, sponsorship and associated project budget requirement
* Requirement analysis – System I/O description, user requirement definition, functional and security requirement
* Data analysis – Involve data collection process, data validation, data storage, manipulation and retrieval
* Process analysis – Data/process flow analysis, process decomposition and system interfaces
* Application architecture – Analyze application information structure, usability, user interface design, interaction and application implementation.

### 

### **4.2.3 Object-oriented design using UML**

A detailed object-oriented design for the online shopping system will be developed. UML will be used again for the graphical representation and documentation of the design. The system will primarily concern itself with the product exploration process. At its core, a user will be able to view product listings in real time by the MongoDB back-end system. In addition, the system will allow users to search, sort, favorite and purchase the items. The class diagram clearly maps out the structure of the MerchantDice system by modeling its classes, attributes, operations and relationship between objects. The use case diagram will show how the user performs the different functionality of the application as well as the interactions with the database actors to achieve the desired outcome.

**4.2.3.1 Class Diagram**



#### **4.2.3.2 Use Case Diagram**

### **4.2.4 Prototyping**

The prototype will be a working example of part of the system for demonstration and proof of concept purposes only. It will include the login, registration as well as the product catalog UI where users can view products listed in the database. The prototype will be presented to the implementation team.

# 

# **5. Constraints**

## **5.1 Scalability**

The Products database is easily scalable horizontally. Given that MongoDB is a non relational database, it allows for increased flexibility in making changes to the fields we need to store. However, it is not easily scalable vertically as it will definitely use up more resources.

## **5.2 Credibility**

The MerchantDice application does not take into account the credibility of the registered users and gives full trust that they will not use the platform for any malicious purposes such as scamming attempts (selling damaged goods etc.). The team thus has to find a way to build a stronger customer confidence to tackle this issue.

## **5.3 Project Schedule**

There is an approximate time frame of 3 months to build a fully functioning web application that will enable buyers and sellers to conduct transactions online.

# 

# **6. Operational Requirements**

## **6.1 Help Desk Support**

MerchantDice provides support through the implemented chat system for problems such as, slow or sluggish system response time, incompatible browser features, application errors, system downtime inquiries, account lock-out assistance, etc.

## **6.2 Application Services and Technical support**

Programmers and application developers from the MerchantDice team will have access to source code to address bugs or make system enhancements as and when deemed necessary. Network and database administrator support is also required to maintain a 24/7 system uptime.

## **6.3 Administration Features**

System security and access levels are provided in the online shopping platform. There are varying levels of system access and functional authority. Each user can only modify product records limited to his/her own product data. Only authorized system administrator(s) have access to all user registration records.

## **6.4 System hardware fail over and routine back up**

Computer operations center will handle system hardware tasks such as data tape back-up, hardware maintenance, fail over, scheduled system patches and maintenance.

## **6.5 Audit Trail**

System audit trails are an inherent part of all product purchases. Among others, all transaction records will capture what action was taken, when (time-stamp) the transaction occurred and who made the transaction.

# 

# **7. Functional Requirements**

The online registration system is a “self-service style” system that shall initially address the student registration needs.

## **7.1 CRUD for User account (Authentication)**

Users can register for new account, delete account, update of user details

* System must allow the registration of new users
* System must allow for the deletion of users
* System must allow for the update of user details which entail their username, name or password
* System must allow for registration of new users through their google accounts[Third-Party Integrations (Google)]
* Internal database

## **7.2 CRUD for Products (sellers)**

* Users who want to sell should be able to:
  + Specify the price and name of the product
  + Upload an image of the product
* Show product listing based on selected genre/category
  + Products should correct filter the genre
* Allow customer to search for products
  + Products of a similar type should be shown in the web page, segregated by pages/a ‘view more’ option.
  + Prices of products should be displayed there, for easy comparison

## **7.3 Sort Products**

* System must be able to sort the products by date posted, price , seller‘s rating [Allow customers to sort the products].

## **7.4 Favorite Products**

* System must allow users to favorite their products such that they can view it anytime.
* System must be able to store the purchase history of their users such that the users can view it anytime
* Users must be able to give rating/feedback to sellers they have previously bought from
* Users must be able to message sellers based on a product he/she is selling
* System must be able to record number of clicks for each user per category
* System must be able to record number clicks per product

# **8. Input Requirements**

## **8.1 User account**

Each user is uniquely identified by their email address which is provided during the registration process for the application. The user must know this. This key maps to all his/her shopping and chat record information in the system.

## **8.2 Product sorting**

The view of the product listing can be sorted by the product type, price or name, which are inputs provided by the user in the application. This presents the user with a filtered view of all products in the system.

# 

# **9. Process Requirements**

The following are among the inherent requirements that the system must be able to handle.

## **9.1 Database transactions**

The system must be able to create, read, update and delete records from the database system where necessary.

## **9.2 Data integrity**

The product information displayed has to be uniform across all users using the application.

## **9.3 Data validation**

Data error from the user’s end and from the back-end database-processing end must be gracefully handled. There will be data validation and error-handling routines as part of the online registration system.

## **9.4 Performance**

Must resolve locking issues and handle concurrent use of the system on a 24x7 basis. Send, receive and display user messages to assist the overall user experience.

## **9.5 Data repository**

The MerchantDice system will maintain the MongoDB database as the main repository of data.

# 

# **10. Output Requirements**

## **10.1 Product purchase summary**

After a user purchases an item, the user must have a view of the item being purchased as well as its product details. The database will be able to display past purchase histories.

## **10.2 Chat message**

After each message sent, the user must have a view of the current chat screen with the other user. The database will be able to display all past messages.

# 

# **11. Hardware Requirements**

## **11.1 Network**

Global network infrastructure (wired and wireless)

## **11.2 Computers**

Mac, Unix and Windows computers

## **11.3 Mobile Devices**

Android and iOS mobile devices

## **11.4 Production support systems**

Web server computer(s) and related hardware support (back-up tapes, redundant drives, UPS, etc.)

# **12. Software Requirements**

## **12.1 Client Operating Systems**

UNIX (any flavor)

MAC

Windows

## **12.2 Client Application**

React and Java Script compatible browser:

Google Chrome

Microsoft Edge

Mozilla Firefox

Safari

Opera

## **12.3 Network system**

Network software and protocols in order for systems to communicate:

TCP/IP

HTTP

HTTPS

FTP

## **12.4 Mobile Operating System**

Android

iOS

## **12.5 Licenses**

Valid licenses are required to run software from third party vendors:

To use application development tools

To use web server, application server and database software in development, test and production mode

# 

# **13. Deployment Requirements**

1. Niinimäki, K., Peters, G., Dahlbo, H. et al. The environmental price of fast fashion. Nat Rev Earth Environ 1, 189–200 (2020). <https://doi.org/10.1038/s43017-020-0039-9> [↑](#footnote-ref-0)