COMP 3059 – Capstone Project

Team 06- Sprint 03

Keda Clothes

High-End Fashion Rental Application

Team Members:

Manjot Sidhu - 101165757

Benjamin Jenkyn - 101184985

Matthew Mukherjee - 101190394

1.0 Introduction

This Systems requirements analysis and design document contains the information required to create the databases for the application. It breaks down the functional and nonfunctional requirements as well as the plans which are in scope, which are we plan to add to the initial phase, as well as out of scope, which are to be added in the future. This document also includes many diagrams which provide a clear path forward for the creation of databases required for the application.

1.1 Purpose

The Keda app is an app which will be designed as a platform and community for the rental of high end and limited release fashion and streetwear. This app will provide a safe community which guarantees authentic items and includes a secure payment portal for the safety of the users who chose to rent as well as those who chose to rent out their valuables. It will include the reviews and reporting system to make sure the community remains a safe and trusted place for users to continue their business. Our app has many plans to expand in the future but for now our main focus is starting up as the main player in a market we feel has been left largely untapped.

1.2 Scope

The Keda app will focus on making sure that high end streetwear, sneakers and fashion are attainable to the masses. Our target market is retailers looking to make recurring profits from their purchases and clients wanting to wear coveted fashion without the need of a purchase commitment. The app will make sure that transactions are safe, items are legitimate and that a safe community is built within the app. Our goal is to begin the launch on iOS for now and expand in the future.

In Scope	Out of Scope
The ability to pay through the app instead of in person or outside of the app	Currently the app is limited to Toronto but in the future we would like to be available in other geographic regions in Canada
The app will only be available to residents of Toronto and iOS users	Currently we will be having authenticity checking done by our team, in the future we would like to implement a more robust system of external staff members to handle a higher number of order verifications
The app will have a system to verify authenticity of all items listed on the platform	Presently our app will not provide personalized recommendations to the users although in the future this is an implementation.
A feature that will allow users to see the items they have rented as well as the duration that the customer has left to use it.	Currently the app provides no price guidelines to renters, in the future we would like to make our app more accessible to users who might not have a deep understanding of the fashion market by using machine learning to suggest a rental price based on previous listings and trends
The ability to search the entire catalogue of items listed on the app	The app is currently only for rental of fashion although in the future we may look into the possibility of adding the sale of clothing as well.

The app will have the ability to send messages between the client and the renter to inquire about the items as well as negotiate prices	Presently this app does not have a
The app will have the ability to post items for rent where the renter will include the item name, description and price. The app will also include the ability to modify the listing after using an edit function.	The app does not have the ability to suggest any items which may go well as an outfit. In the future it is a possibility that the app will have the ability to suggest complete outfits using human curated suggestions from stylists as well as possible machine learning.
The app will include the ability for both the renter and the client to share their experience of the rental to help others make their decision.	In this scope there is no way for the app to find out what users would like listed as rentals. Down the line we may look to adding the ability to have clients request items that they would like to rent out.
The app will have the ability for users to create support tickets if any issues arise with the app of with a rental	In the current scope we do not have a dedicated team for support queries. Should the app take off we would like to be able to hire more people to handle customer support queries.
The app will have the ability to track shipments using 3rd party tracking numbers provided by the renter	Currently the app does not handle the shipment of items, we rely on third parties. As we may decide to have our own employees that will handle shipping in the city so as to cut cost and maximize profits.

2.0 System Overview

The System Overview section introduces the system context and design.

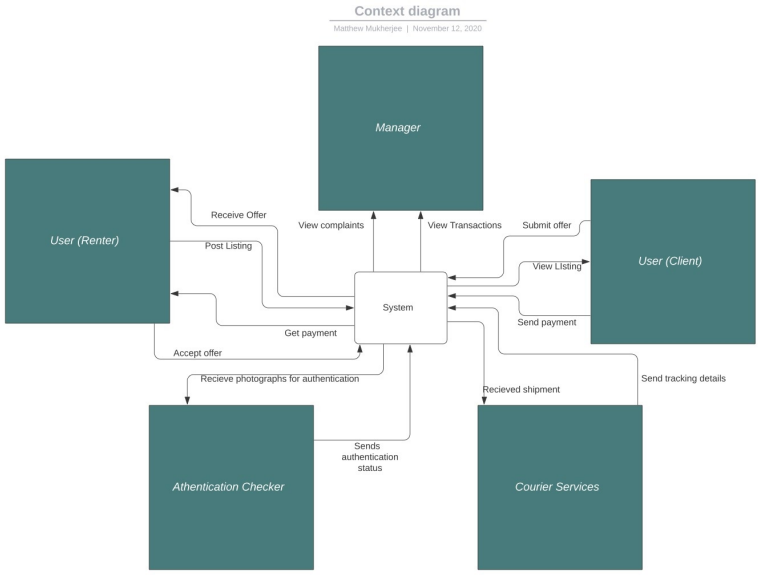
2.1 Project Perspective

In the world of fast fashion, trends can change at a moment's notice and keeping up with the highest end brands can get expensive. To add to this expense a lot of rare and highly sought after items often get purchased and resold for a much higher price than they were originally being sold for. For many people the price of these items is too high and as a result people may not get to wear the items that they want, or they wind up spending too much money and end up in an unfavourable position over an item that they will not use for very long.

On the other end of the spectrum we have resellers who are people who buy and sell rare fashion items, often sneakers, for a profit. The profits that people make are a one time deal and the amount they make on an item is invariable and depends on how popular the item will be, which can vary wildly.

Our app will be a self-contained system, with the supplementation of APIs, that will allow users with rare items to rent out clothing, sneakers and accessories to users that wish to wear them without the need for a commitment to purchasing an expensive item. We will offer a safe, secure and trusted platform for the community to come together and share fashion trends for all to wear.

2.2 System Context



The purpose of this system is to satisfy the demand for rare and high end fashion for those who wish to wear these items and are willing to rent them. This will be beneficial to those who cannot afford to own these high end clothing but still want to wear them for special occasions, or just to show off. It will also be beneficial for those who want to wear these clothing items while they are popular without needing to commit to a purchase, so they can always keep up to date with the latest fashion.

The third party suppliers who will be renting out the items they own will also benefit as they are able to rent out their clothing and make a steady profit long term instead of a one time profit.

Renters will be able to list items for rent through our app by providing information such as the item name, description and pictures. The users will make the transactions through the app and the owners of the app will get a percentage of the profits. The client will submit an offer to the renter which the renter can view, accept, counter or reject. For the payment we plan to use Shop Pay by Shopify and Apple pay as it will make transactions much easier for us as we will not need to make our

own payment method. For shipping we plan to use third party services such as FedEx and UPS as they are already established and it will make things much easier.

To make sure the application contains only authentic items, an authentication verification specialist will review pictures and inspect to verify authenticity of the item. Once an item is verified as authentic the listing goes live on the app. The clients will be asked to leave a review of the experience with the last rental after the item is returned to let others know if the experience with the renter was positive or negative.

2.3 General Constraints

Initially the system will be limited to iOS devices, primarily iPhones, as it will be easier to natively develop for one platform as opposed to two and the majority of our users will likely be in possession of an iPhone. As cost and time are our biggest restraints, we will not be developing a website or android app until we are able to secure more funding and if we have managed to gain a reasonably sized user base. We also plan to begin production of our app in Toronto as it will make distribution much easier. It will also reduce the amount of clientele that we will need to deal with initially so that our small team does not get overwhelmed. If we are successful in Toronto we may then expand further and potentially hire more people, however, this is outside the scope of the current project. The app will not dictate the market value of the items listed and will rely on external sources and the individuals creating the listing to determine the price.

2.4 Assumptions and Dependencies

The following assumptions have been defined for the Keda application:

iOS Users: We assume that most people in the high end fashion world will be utilizing an iPhone as it is a phone that is often seen as “premium” and “high end”. To that end we will be utilizing Swift to code the application meaning that it will be exclusive to iPhone. We would like to, in the future, expand to the android app store as well as make our own website however for the scope of this project we will only be utilizing iOS. Someone who is very up to date with fashion trends wants to have a photo shoot but does not wish to purchase the articles they can use our app to rent it out. Since they are inclined to purchase high quality and higher end items, they are most likely to be iPhone users.

Authentic merchandise: Those that will be renting out clothing will upload photos of the merchandise in order to allow the customer to verify that it is legitimate. In cases where the legitimacy of the product is difficult to verify or the product is an item which is frequently faked, the item will be sent in to us in order to verify the legitimacy of a product at which point it is approved and the item will be updated in the database to “verified”. In a situation where a potential renter wants to rent out their rare merchandise through our app. In the process of creating the listing they notice that photos of the product from different angles are required for verification and upload them accordingly. In some cases the app will alert the listing creator that the item must be verified in person in order for it to be listed as the item is very frequently faked.

Accurate descriptions: It is up to the owner of the clothing to provide an accurate description of the item that they want to put up for rent. This description should include type of item (i.e shoes, hoodie, purse, etc.) the condition of the item, any requirements for handling the items and anything else that the renter feels is

important to include. In a given scenario where a potential renter is preparing to make a listing. After taking the necessary photos they add a description of the item which includes all of the above requirements. If the description does not include any of the necessary information then the listing will not get posted.

Contactless payment: All transactions will be done within the app and there will be no meet up to exchange cash or in person handling of cash along with not sharing personal information or details. In a situation where a potential renter wants to proceed with renting out merchandise to the client however the client is insisting on meeting up in person to exchange the merchandise and pay the renter in cash. Due to safety concerns, this is not compliant with the app police and is grounds for termination of the offending accounts.

Credit card hold: Will ensure that the client has enough funds to cover for any damages that may occur to the marchandise during the rental period. A potential client wants to proceed with a rental transaction and has negotiated a fair price with the client. Authorization holds a fund on the available balance until the transaction is clear.

The following internal and external dependencies have been defined for the Keda application:

Internal Dependencies: Teamwork and communication with effective communication for productive results outputting quality work. Should be able to engage effectively with stakeholders to ensure deadlines are being met for a content clientele, therefore every team member needs to be sure of their responsibilities and be able to carry them out. Training and software knowledge that the team acquires to complete the project, while being able to pick up

new technologies and adapt. Time and cost of equipment and upgrading new softwares.

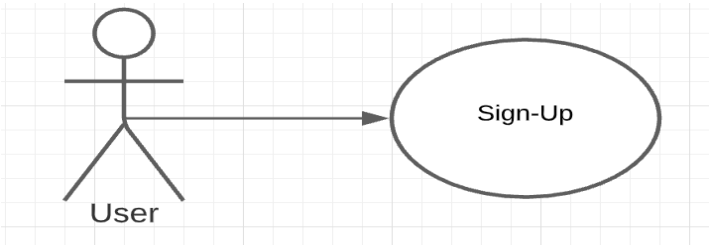
External Dependencies: Time and cost the financial constraints as a new tech startup. It will be crucial to budget expenses and cost to ensure all funds are going towards the appropriate need, and if need be then funds will be reallocated to where the resources are needed. All aspects of the app need to be developed to the highest quality to ensure we are above or atleast to par with our competitors. along with ensuring that our clients approve of the overall application by being satisfied with the customer service. To ensure clients stay content with the application our team will ensure to stay on top of all technological advancements and implement or develop new features while checking in with the project coordinators approval.

3.0 Functional Requirements

This section describes specific features of the software project. If desired, some requirements may be specified in the use-case format and listed in the Use Cases Section.

3.1 Functional Features and Use Cases

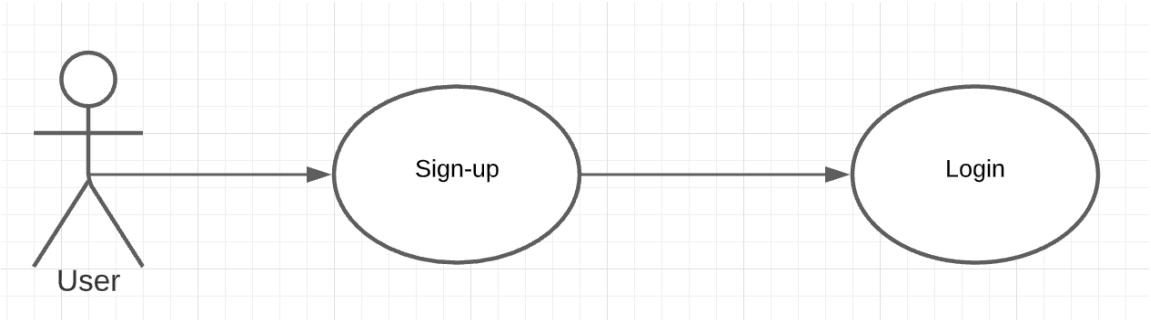
1. ***Sign-up:***



Introduction	User who downloads the app can create an account
Inputs	● Login screen will have a button to create an account ● Takes user to account creation where where they will have to input the information below ○ Username ○ Email ○ Password ○ Phone Number ○ Shipping Address ○ Billing Address ○ Security Question and Answer ○ Promo code (if applicable) ● Page will have a create account button at the bottom to confirm account creation along with a captcha
Processing	● If all information is valid, including the captcha, the system will store the user's information in the users table
Outputs	● If information is invalid then there will be error messages displayed accordingly such as ○ Email invalid

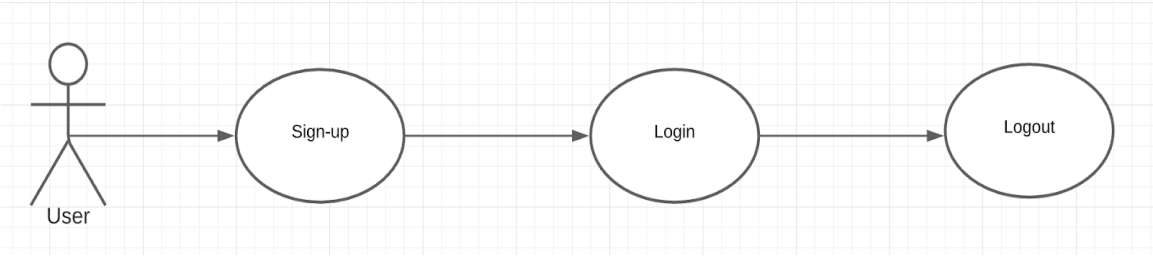
	○ Phone invalid ○ Captcha invalid ○ Address invalid ● The user will get a text and email with verification codes to confirm both are valid. ● Account will be created

2. ***Login***



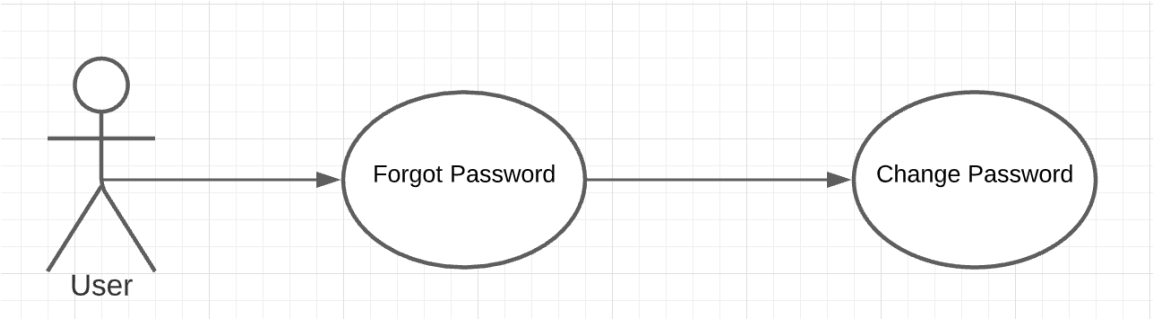
Introduction	Once users create an account they will log in and be able to post items as well as rent out items from the app.
Inputs	● Once users click login they will be taken to the login page where they will enter their username and password ● There will be a login button at the bottom of the login page
Processing	● The username and password will be searched in the users table to verify if the credentials are correct ● If credentials are incorrect the app will give error messages such as ○ Username invalid ○ Password invalid
Outputs	If all information is correct the user will be taken the the catalogue page of the app

3. *Log Out*



Introduction	Once users create an account they will log in and be able to logout of the application once they are done using it.
Inputs	• Once users click login they will be taken to the login page where they will enter their username and password • Log out button as well, if they user wishes to exit the application
Processing	• The session will be ended and the user will be redirected to the login page
Outputs	• User will be directed to the log in landing page

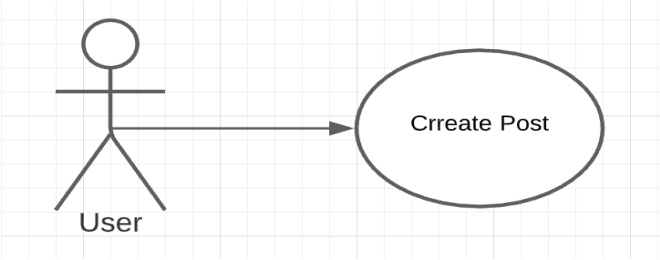
4. *Forgot password*



Introduction	On the login page there will be a “forgot password” button in case the user forgets the password they entered upon account creation.
Inputs	• The “forgot password” button on the login page will take the user to a separate page where there will be a form to enter the following information

	○ Username ○ Email ○ Phone number ● Users will then click the recover account button at the bottom of the page where they will be taken to a new page where a code that was sent to the email/phone number will have to be entered. ● User will then be asked to enter a new password
Processing	● The system will verify the requested credentials with the users table ● If the information entered is correct user will get a text message and email with a security code which they use to enter before creating a new password ● The user then enters a password which will be stored in the users table
Outputs	● If the password change is successful user will be notified that it is successful and they may attempt to login ● If the password change is unsuccessful then the user will be notified of the reason and they may reattempt the change.

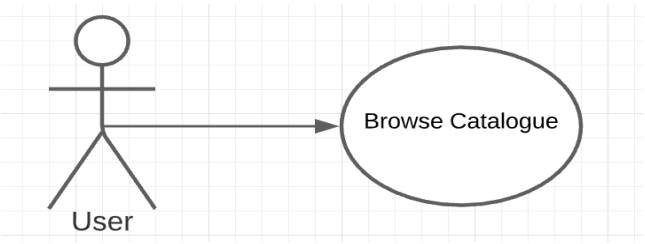
5. Create Post



Introduction	Users will have the ability to create a listing on the app
Inputs	● User will enter information such as title, photos, description, price and condition

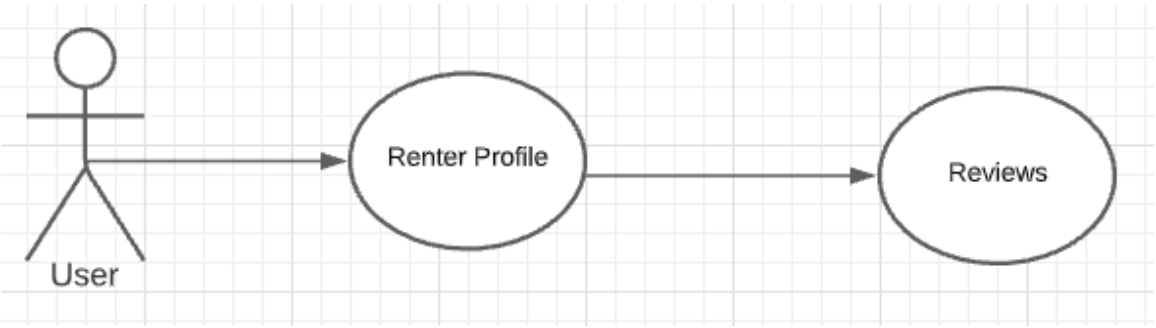
	• User will click submit listing when form is complete
Processing	• Listing will be created and added to the listings table
Outputs	• User will get a confirmation message that says 'listing successfully posting' • If unsuccessful, User will get an error message.

6. *Catalogue*



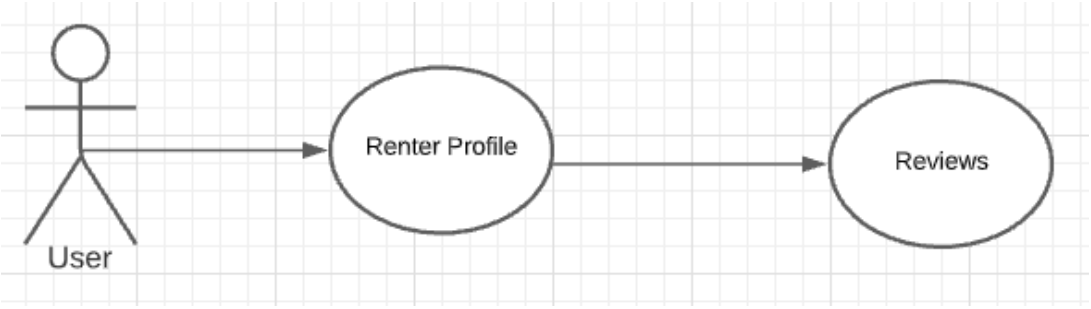
Introduction	A catalogue will contain all the listings including sections for the top listings, top rated renters, new items etc.
Inputs	• Users can select an item from the catalogue • User can sort by options such as, popular, related, price, type, size, gender
Processing	• System will open listing from listings database • System will sort listings from listings database
Outputs	• User will be taken to listing after selecting from the catalogue • User will be displayed catalogue page based on sort selection

7. ***Reviews***



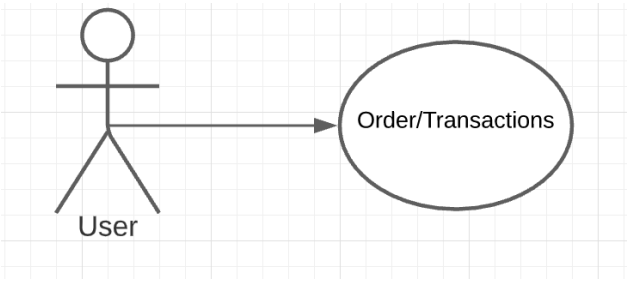
Introduction	Users will have the ability to submit reviews of their experience with a renter or client
Inputs	● After the transaction is completed the user will be asked to provide a review on their experience with the other user by use of a review form. ● User will be asked enter the following information based on their role in the transaction ○ Overall rating out of 5 ○ Value out of 5 ○ Condition out of 5 ○ Condition returned in out of 5 ○ Interaction out of 5 ○ A text box to enter a written review
Processing	● The system will take this information and add it to the reviews table
Outputs	● The user will be alerted that the review was posted successfully

8. ***In-app messaging***



Introduction	Users (renters and clients) will have the ability to message each other to discuss the listings and negotiate prices
Inputs	• Client will select the 'message' button on a listing which will redirect them to a messaging page • After initial contact is established users will be able to discuss the items including the condition, price etc and the renter will have the ability to modify the price based on the negotiation.
Processing	• Registered accounts will be processed into the users table • Messages will be handled through an external service
Outputs	• Clients will be able to receive messages • Clients will be able to send messages

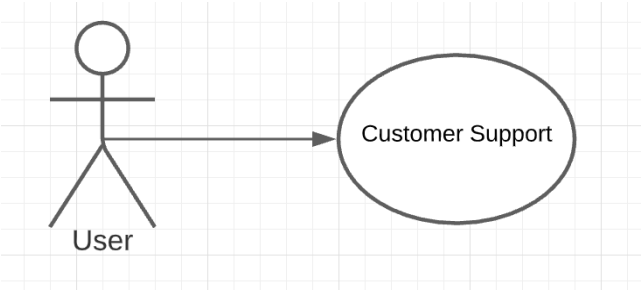
9. ***Orders and Transactions***



Introduction	Users will have the availability to proceed with checking out with

	their rental and proceeding with their payment interaction
Inputs	• Client will enter payment information into a ShopPay or a Apple Pay form User will click submit
Processing	• Payment information is processed through payment partners • Payment information is not stored on our database • An order number is generated and stored in the orders database • Other information about the order such as item, price and users involved in the transaction will be stored in the orders database
Outputs	• Shows order confirmation to the client • Renter is notified that payment was successful • If payment is unsuccessful client will be shown message to try another payment method

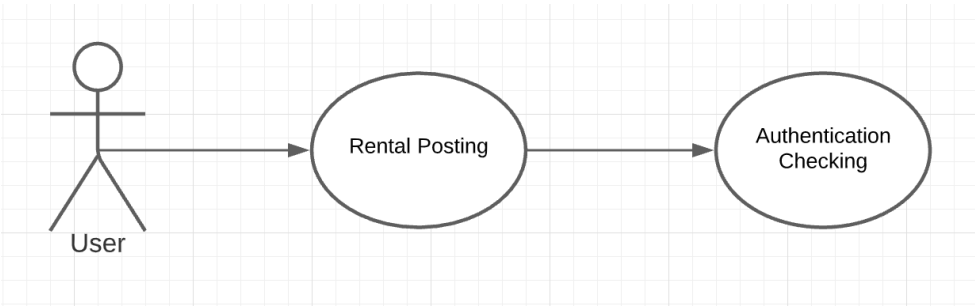
10. ***Customer Support***



Introduction	User will have the ability to express any concern or issues they are facing with the application or users
Inputs	• Select complaint type • Input description about the complaint • Select submit button to forward the complaint to

	management
Processing	• The complaint will be forwarded to the administrators email inbox • After review if the report was necessary action will be taken and the reports will be stored in the report database and tied to the user
Outputs	• User will get confirmation that report was created • User will get response after report has been reviewed

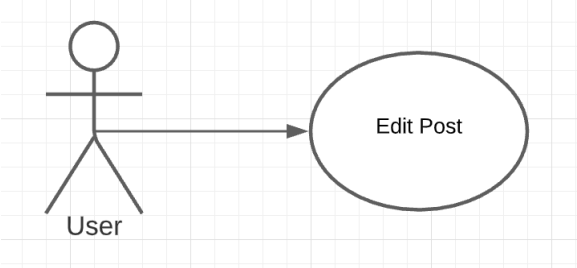
11. ***Authentication checking***



Introduction	The app will have the ability to to verify authenticity of merchandise
Inputs	• Renters will submit pictures into the app while creating a new listing. • Renters may be asked to provide additional pictures by staff if needed • Renters will click submit button
Processing	• Pictures will be uploaded to the Listings table to be views by our staff and also posting on the listing
Outputs	• User will get a notification saying images successfully uploaded • User may be notified if issues with authentication check arise and further inspection including physical inspection

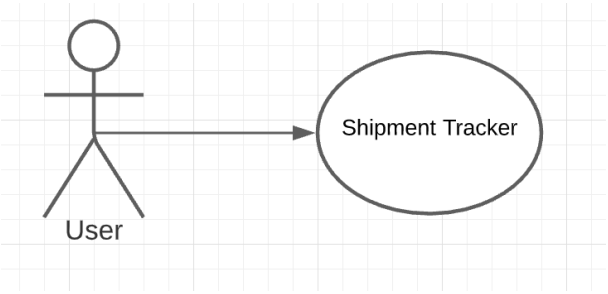
	is needed

*12. **Edit posts***



Introduction	The user will have the ability to making edits and adjust their post after they have uploaded it on the app
Inputs	• User can select edit post button on their post after it's been uploaded • User can select submit button to upload again
Processing	• Listing updates on listing
Outputs	• User will be taken to updated listing with the message post successful

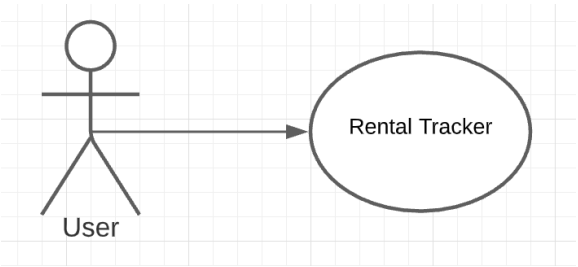
*13. **ShipmentTracker***



Introduction	The app will have the ability to read the tracking number and store it in the orders table to monitor rental time tracking.
Inputs	• Renter will notify the app when they begin fulfillment of the order

	• Renter will add tracking number into app after the shipment is sent with their transaction number
Processing	• We will let the third party shipping company handle tracker and use their apis for tracking on our app
Outputs	• Shipment status will display on order page

*14. **Rental Tracker***

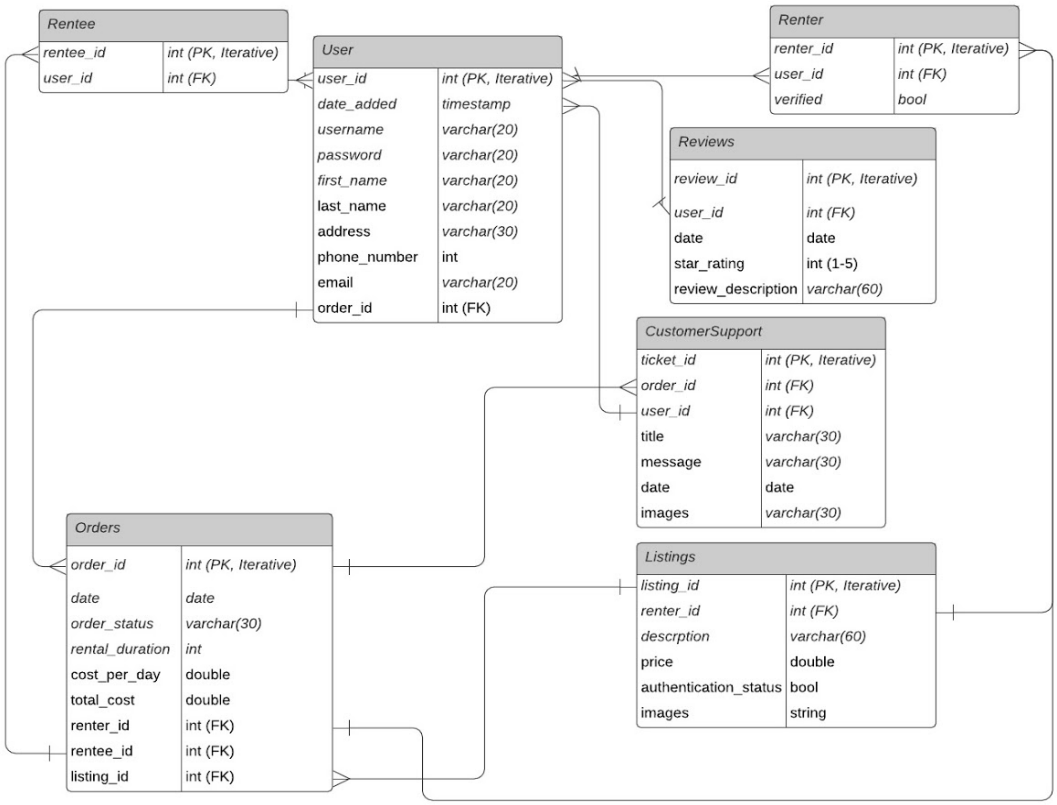


Introduction	Renters will be able to see the amount of time they have an item rented out for and when they should expect to receive it back
Inputs	• Rental will input that the item is shipped • Client will input that the item is shipped back
Processing	• System will update the listings table with availability changes • System will update orders table
Outputs	• The customer will be able to see the item being rented on the specified dates

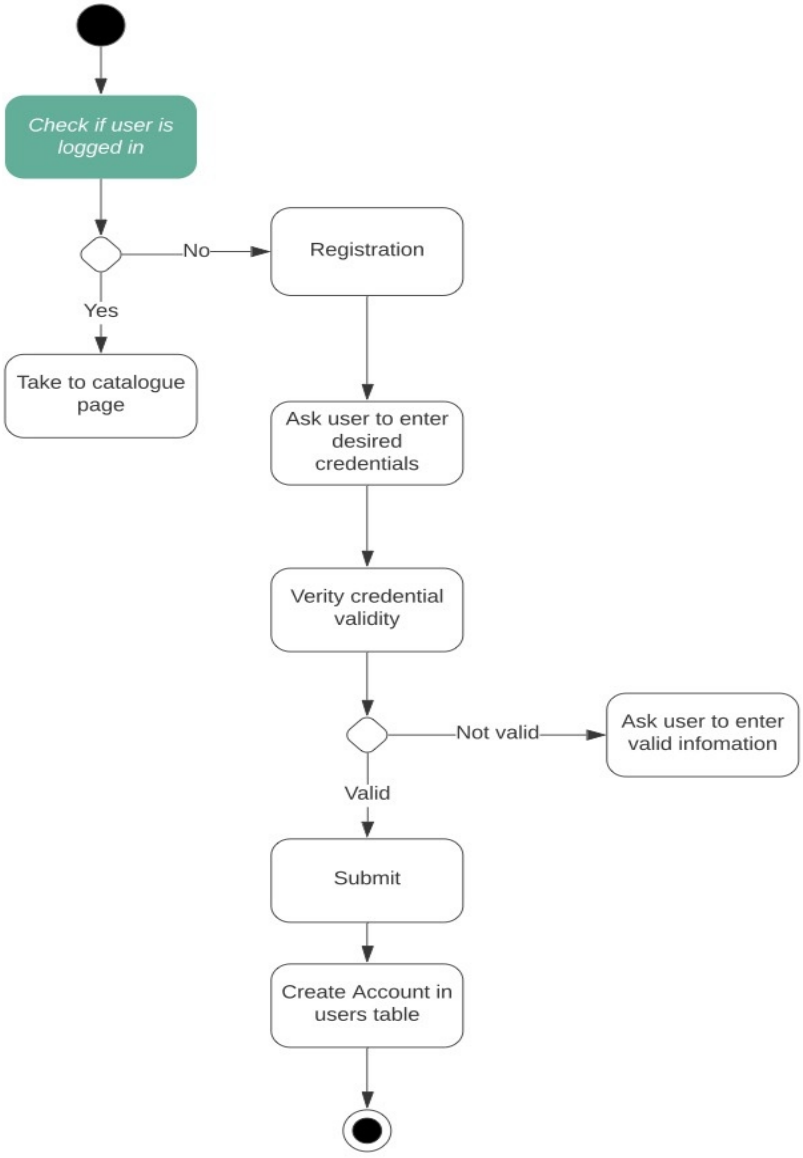
3.2 Data Modelling and Analysis

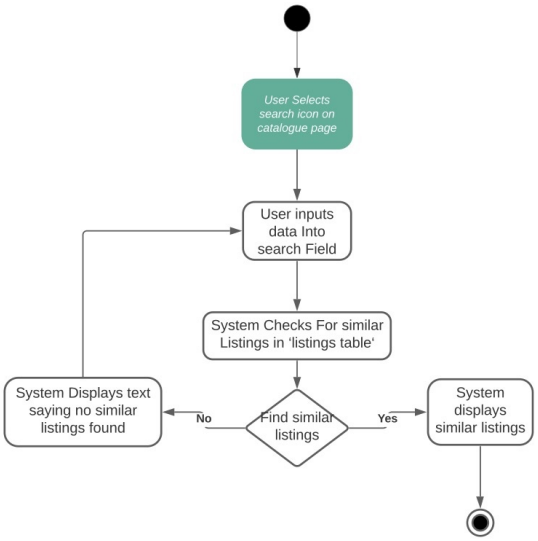
Normalized Data Model Diagram

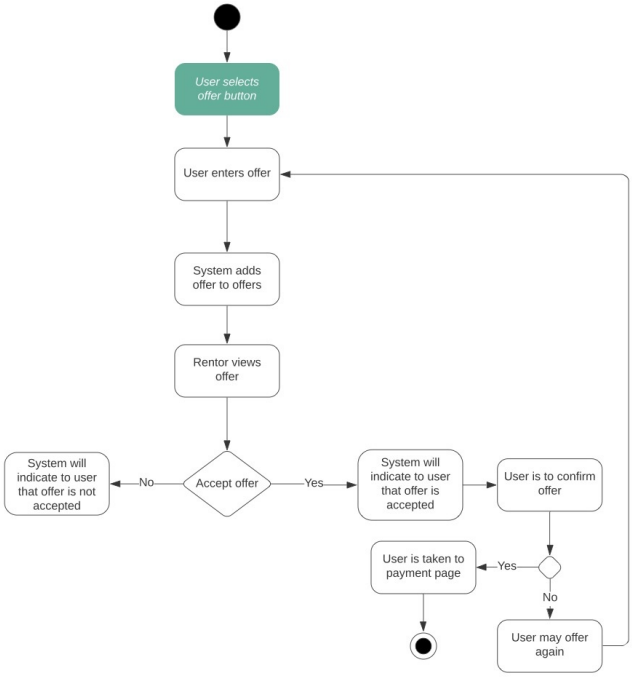
3rd Normal Form:



Activity Diagrams

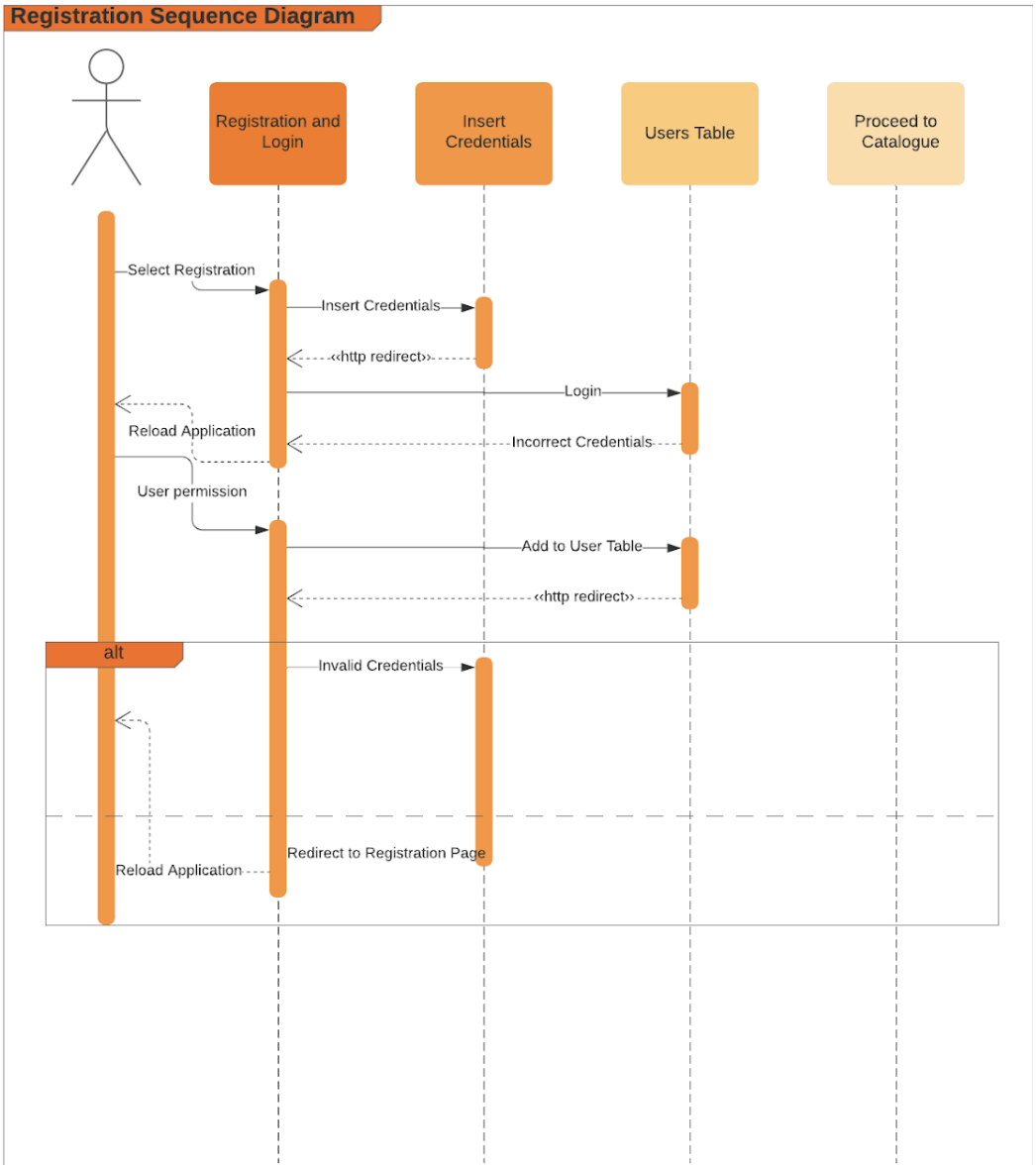



User Selects search icon on catalogue page
User inputs data Into search Field
System Checks For similar Listings in 'listings table'
Find similar listings
No
Yes
System Displays text saying no similar listings found
System displays similar listings

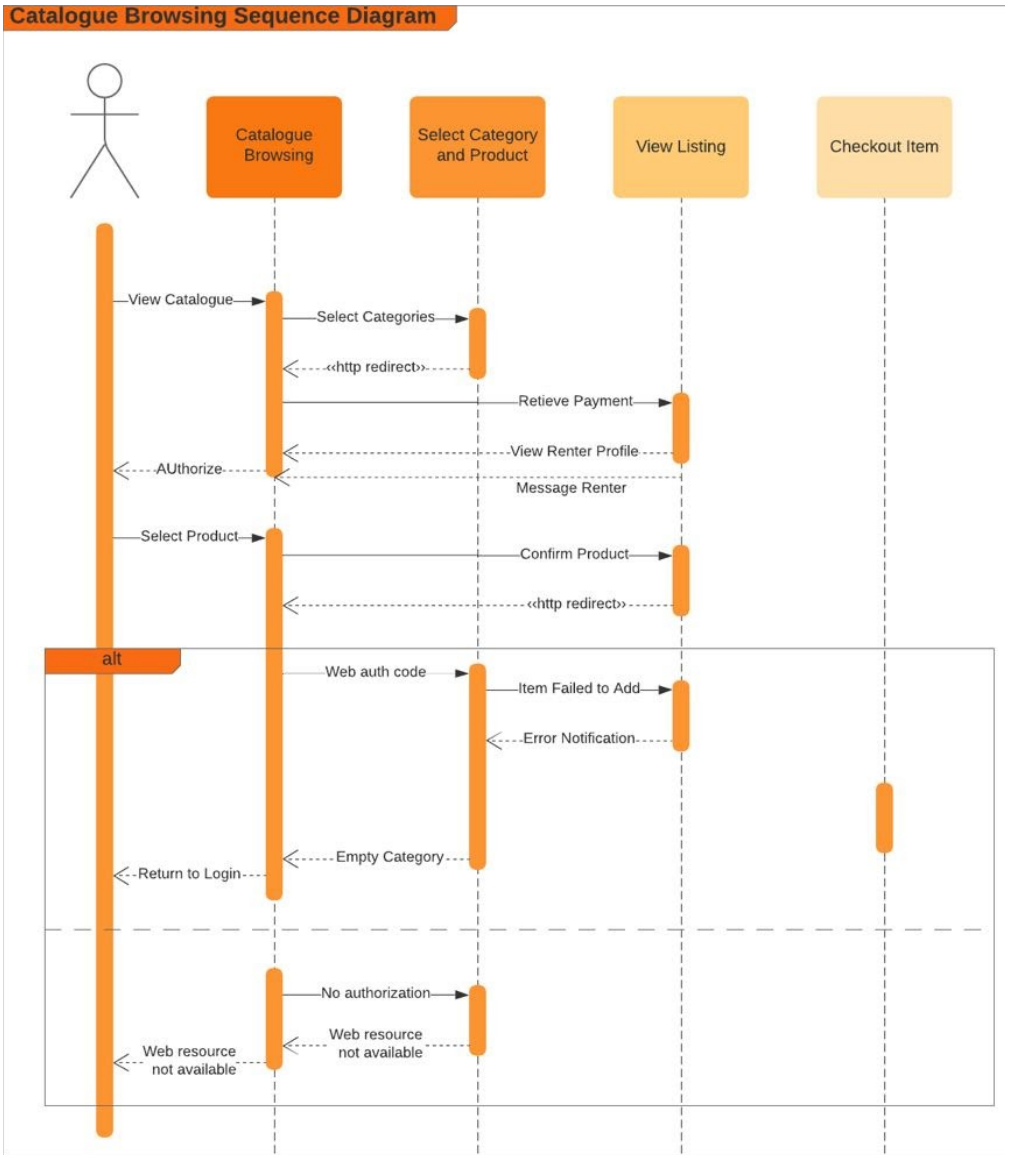

User selects offer button
User enters offer
System adds offer to offers
Rentor views offer
Accept offer
No
Yes
System will indicate to user that offer is not accepted
System will indicate to user that offer is accepted
User is to confirm offer
User is taken to payment page
Yes
No
User may offer again

Sequence Diagrams:

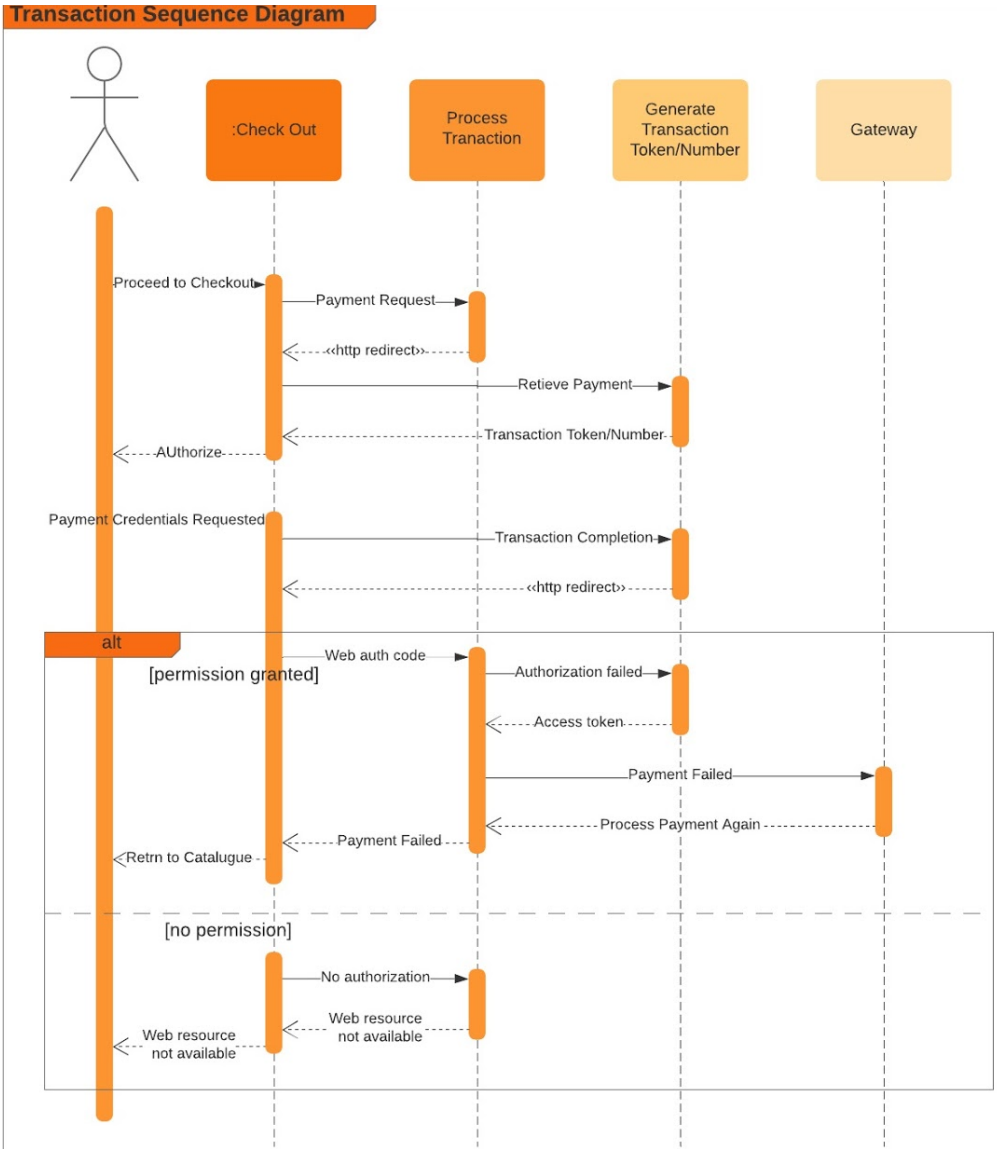
Registration:



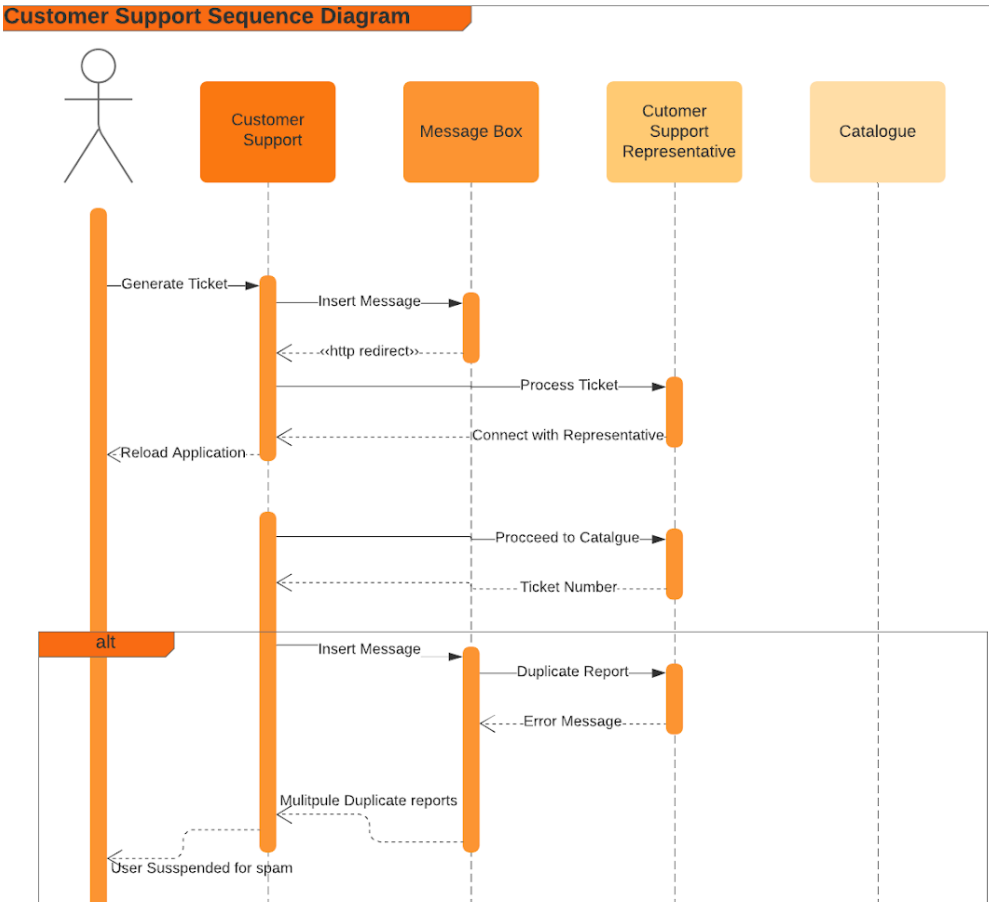
Catalogue Browsing:



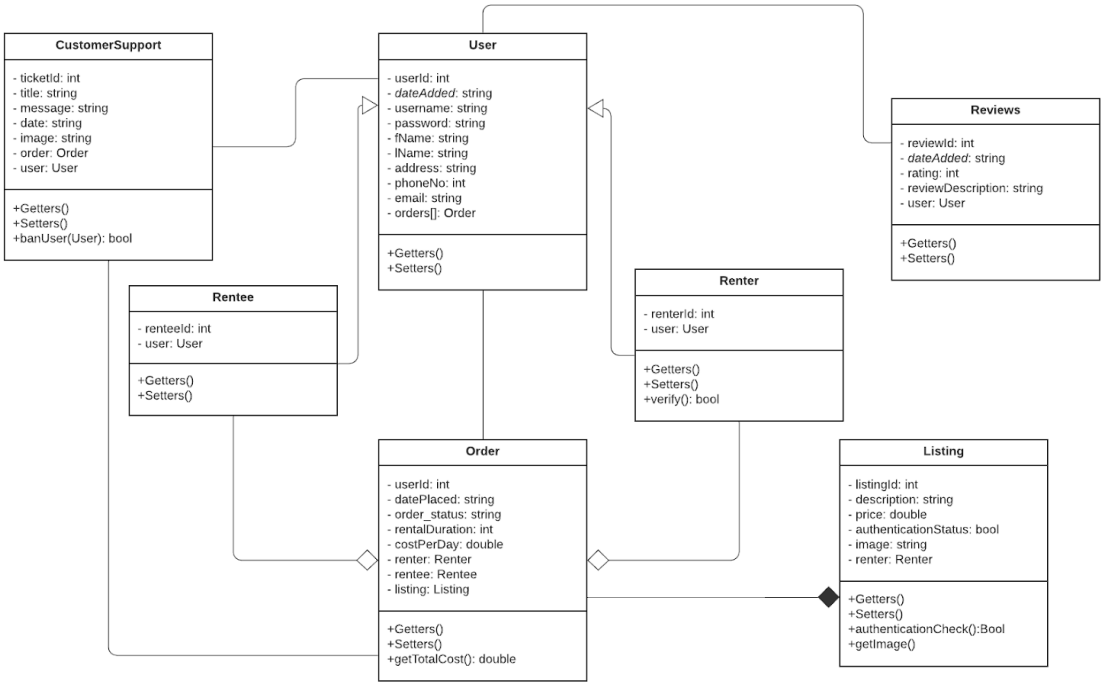
Transactions:



Customer Support:

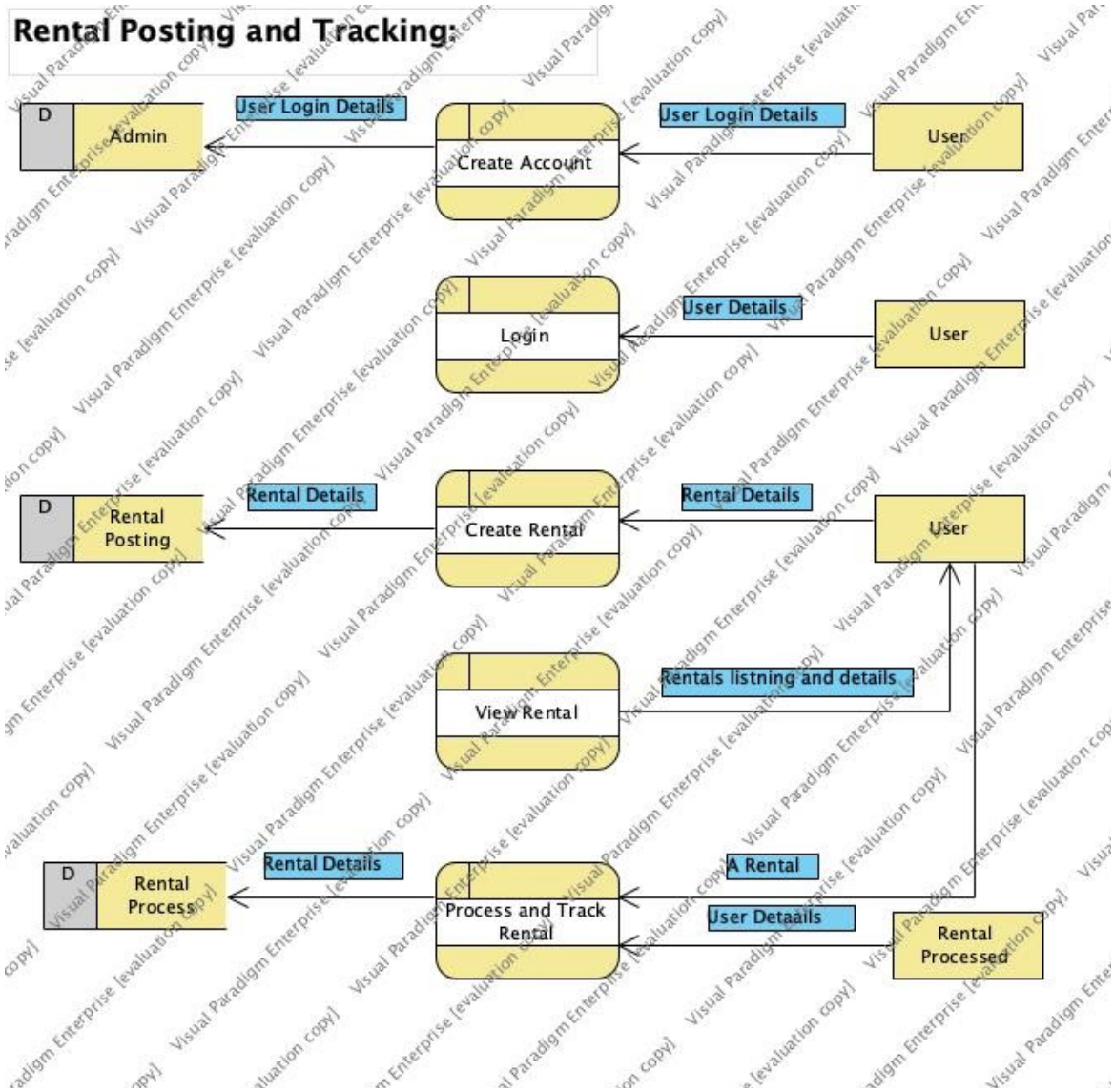


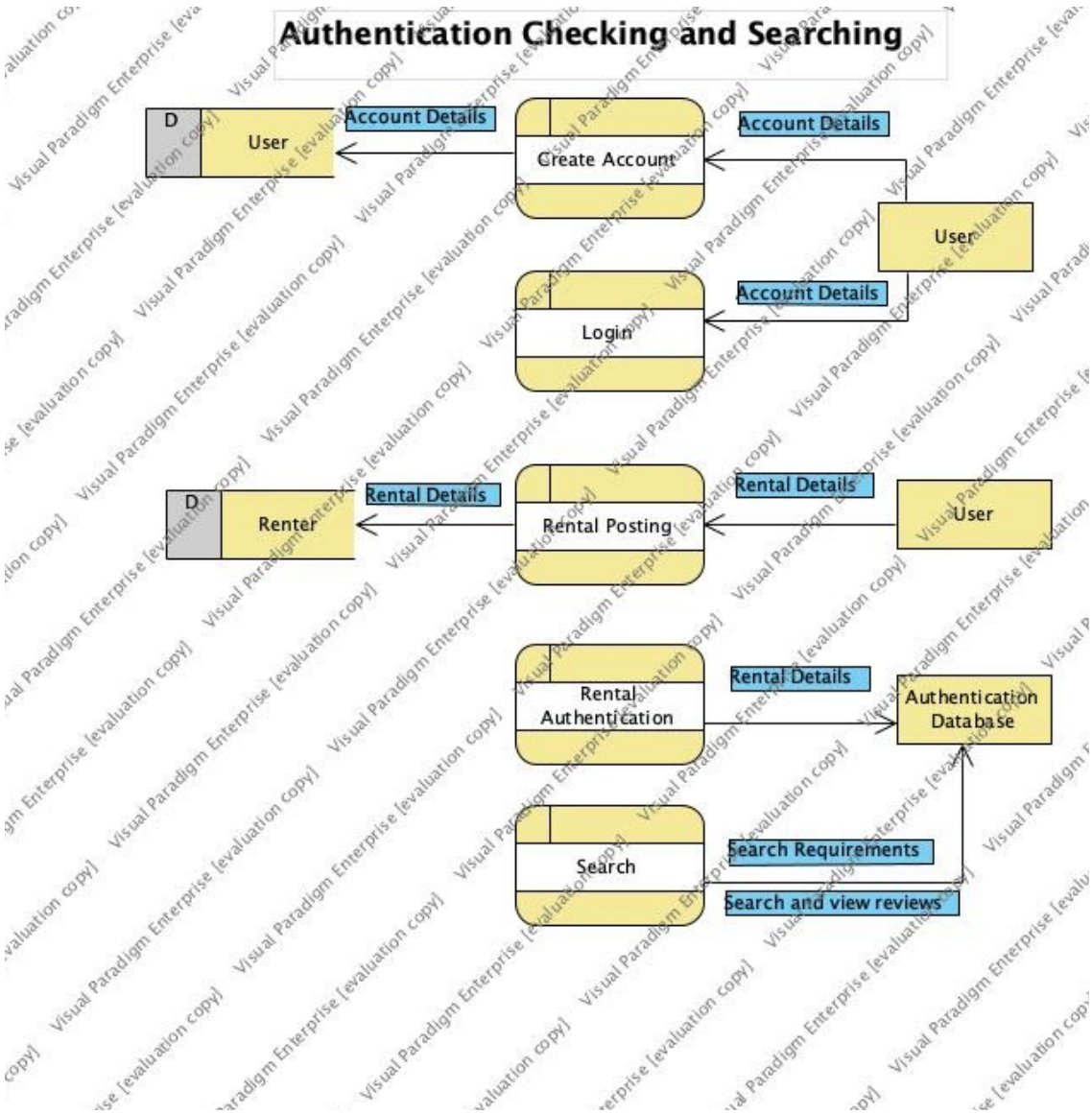
UML Class Diagram:

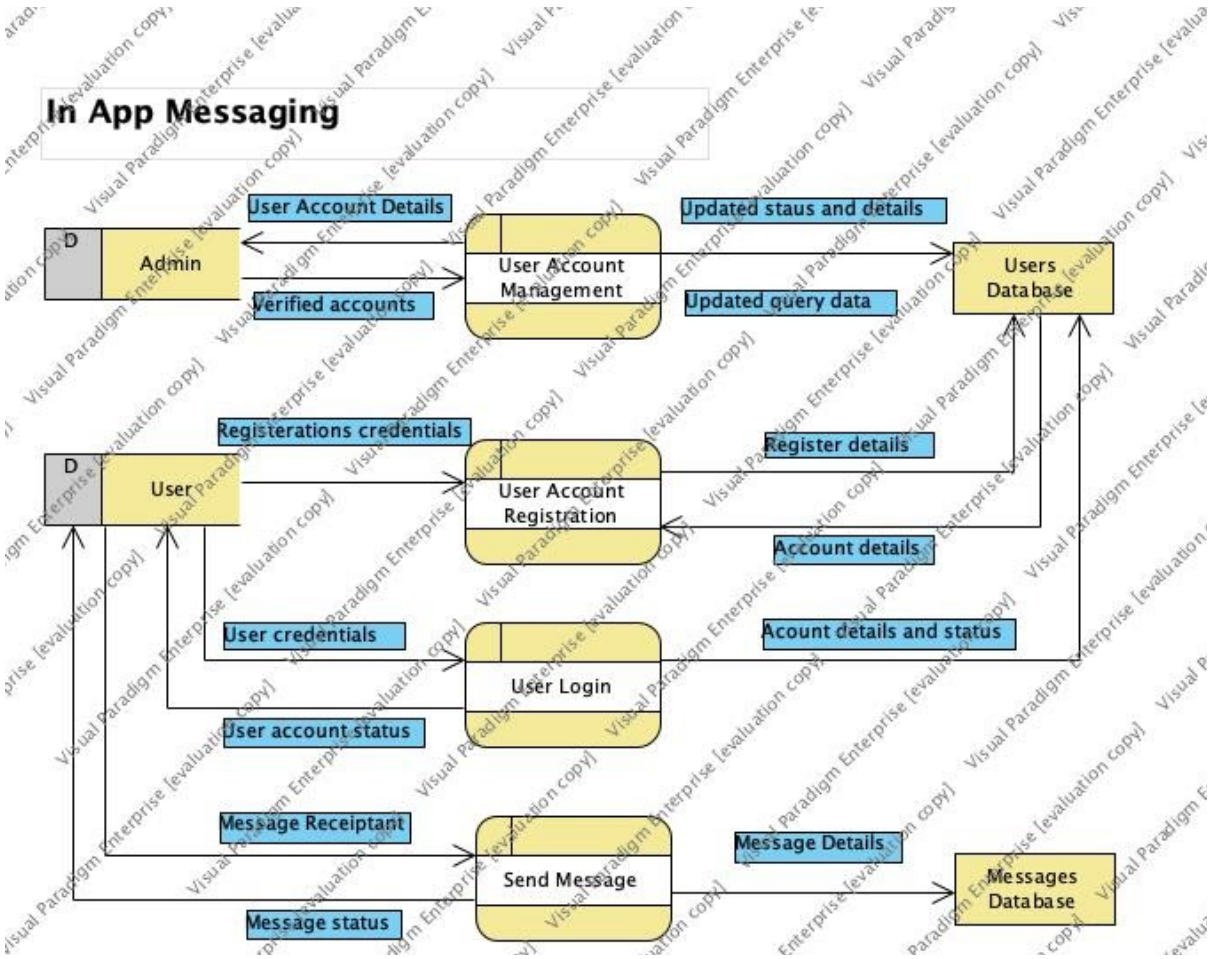


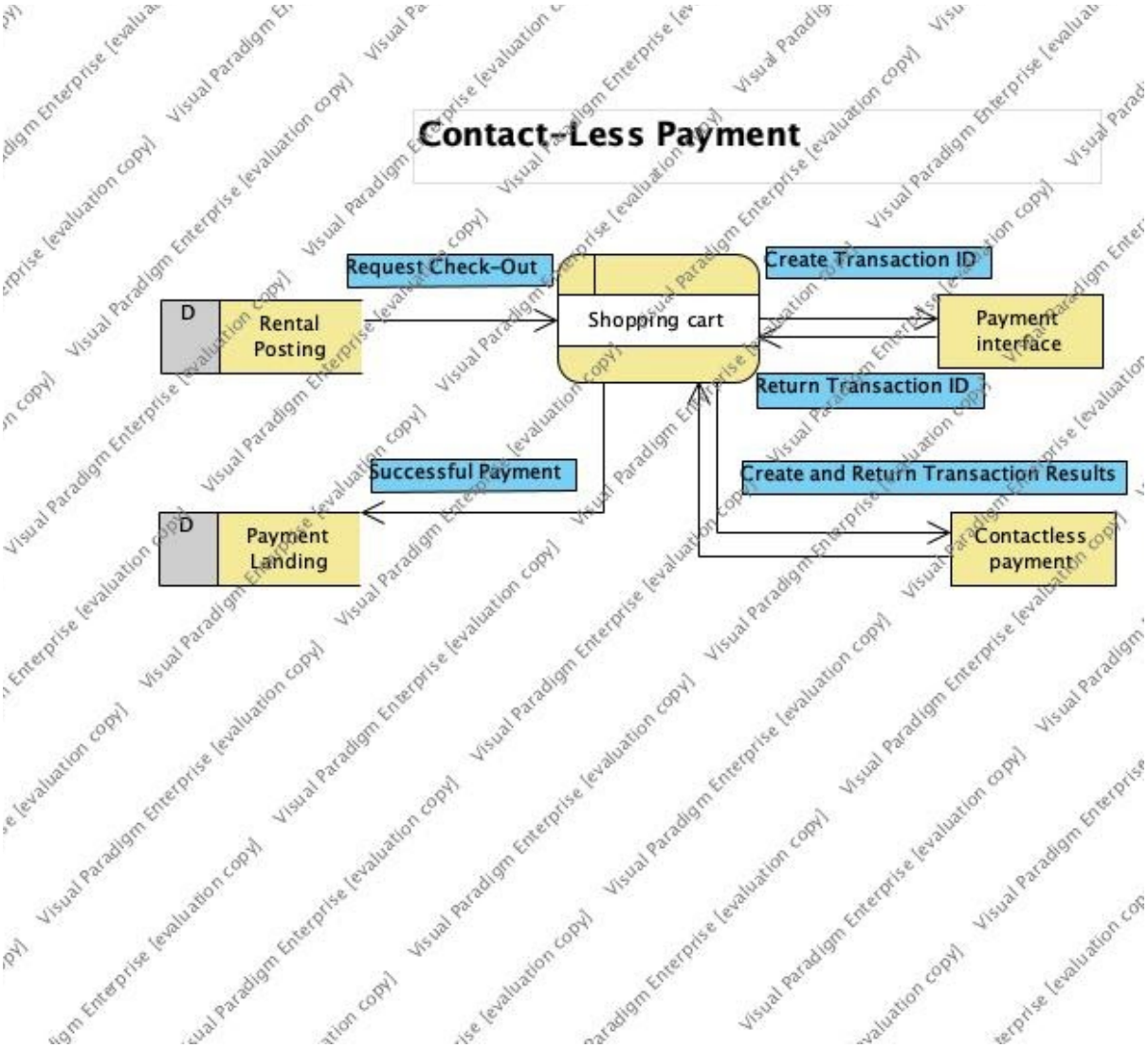
3.3 Process Modelling

Data Flow Diagram




Authentication Checking and Searching
D
User
Account Details
Create Account
Account Details
User
Login
Account Details
D
Renter
Rental Details
Rental Posting
Rental Details
User
Rental Authentication
Rental Details
Authentication Database
Search
Search Requirements
Search and view reviews


In App Messaging
D
Admin
User Account Details
Verified accounts
User Account Management
Updated staus and details
Updated query data
Users Database
D
User
Registerations credentials
User Account Registration
Register details
Account details
User credentials
User account status
User Login
Acount details and status
Message Receiptant
Message status
Send Message
Message Details
Messages Database


Contact-Less Payment
Request Check-Out
Create Transaction ID
D
Rental Posting
Shopping cart
Payment interface
Return Transaction ID
Successful Payment
Create and Return Transaction Results
D
Payment Landing
Contactless payment

4.0 Non-Functional Requirements

Performance: The Keda application will provide an average response time of 3 seconds when the number of users online are over 1000, and everything is being processed through the application.

Availability: The database will be responsible for storing all the information about users, and if it fails to respond users will be notified that the app is currently not responding and advised to try again with their rental in 24 hours as our team will send out emails and messages regarding the issue. When the issue is resolved the user will have to use login credentials to log in again and will only have three password attempts.

Scalability: The Keda application will be starting off in Toronto only first to see how well it takes off. As the need of the application grows we will expand gradually throughout Ontario first by expanding the application so that users from different parts of Ontario are able to register for HypeBeast rentals.

Portability: The Keda application will be compatible with IOS and Android devices through the native react platform, however it will only be compatible with those devices that remain updated as this will make it accessible to 99% of the android devices on the market along with the Apple devices.

Maintainability: The Keda application will be designed in a way where everything is sectioned off and has its own page and specific feature for the user to refer back to. These modules will assist the user with app direction as well as they will be able to flow through the app in a way that will lead them through until the payment process. Maintenance will also ensure all bugs and crashes are fixed up asap.

Security: The Keda application will ensure that all information is stored in the correct databases that correlate with the user profile whether they are a client or a renter. No information will be shared throughout the platform that is used to register besides the username the user decides upon.

5.0 Logical Database Requirements

For this application we will be using a database. For initial design and testing we will be storing the database on one of our local machines. We plan to use amazon web services or a similar cloud based system to host our database in the future as it will provide a method that is secure, can be accessed from anywhere and will make scaling the business much easier should we grow.

The users table is the table that we will use to keep track of our users. We have a renter and rentee table which is related to the user table and will be used to differentiate the two different users that will be involved in a transaction. In terms of the app anyone can be a renter or rentee but it helps to differentiate for the transaction. The orders table will keep track of both current orders, and past orders for a certain period of time. The table will use data indirectly from the users_table through the renter and rentee table to indicate the two users that will be involved in the transaction as well as the listing that contains the item that will be rented. Finally we will have a customer support table that will contain any user reports that may come up against the different users of this app.

Users Table:

Keys	Users_Table	Data Type	Notes
PK	user_id	int	
	date	timestamp	
	usernames	varchar(20)	
	passwords	varchar(20)	We would prefer not to store the passwords and let a third party

			handle it, however, if we do store passwords we will ensure they will be stored securely
	first_name	varchar(20)	
	last_name	varchar(20)	
	address	varchar(30)	
FK	order_id	int	For active orders
	phone_number	int	
	email	varchar(20)	
	suspendedDuration	int	If the number is 0 they will not be banned. the number will be related to the number of days suspended. if it is -1 then the user will be banned

Reviews Table:

Keys	Review_Table	Data Types	Notes
PK	review_id	int	

FK	user_id	int	
	date	date	
	star_rating	int	
	review_description	varchar(20)	

Orders Table:

Keys	Orders_Table	Data Types	Notes
PK	order_id	int	
	date	date	
	rental_duration	int	
FK	renter_id	int	User_id of the renter
FK	rentee_id	int	User_id of the rentee
	order_status	varchar(30)	Active order vs past order
	cost_per_day	double	
	total_cost	double	
FK	listing_id	int	

Renters Table:

Keys	Renter_Table	Data Types	Notes

PK	renter_id	int	
FK	user_id	int	
	verified	bool	

Rentee Table:

Keys	Rentee_Table	Data Types	Notes
PK	rentee_id	int	
FK	user_id	int	

Transactions Table:

Keys	Transactions	Data Types	Notes
PK	transaction_id	int	
FK	user_id	int	
FK	payment_id	int	
FK	user_id	int	
	payment_number	varchar(30)	

Listings Table:

Keys	Listings_Table	Data Types	Notes
PK	listing_id	int	
FK	renter_id	int	
	description	varchar(30)	
	price	double	
	authentication_status	bool	
	images	varchar(30)	

Customer Support Table:

Keys	CustomerSupport_Table	Data Types	Notes
PK	ticket_id	int	
FK	order_id	int	
FK	user_id	int	
	title	varchar(30)	
	message	varchar(30)	
	date	date	
	images	varchar(30)	

6.0 Other Requirements

- **3rd party messaging API:** We will be implementing a 3rd party API to bring messaging into our app between the users. This messaging service will be used to by our users to communicate with each other to asks questions, discuss about the items listed and negotiate prices

- **External courier services:** After the shipment is sent the renter will submit the tracking number and name of courier service to the app so that the shipment tracking can begin. To return the item the client will ship back and provide a tracking number to the

- **Shopify for payments:** Upon completing your shopping, the user will proceed to the checkout where they will complete their payment, that is completely contactless, through Shopify. Transactions will be processed and recorded by them into the database.

- **Renters:**The Keda application depends on rentersi in order to function properly and make it a profitable business market, as they will be providing the merchandise for the consumer to rent out and enjoy.

7.0 Approval

The signatures below indicate their approval of the contents of this document.

Project Role	Name	Signature	Date
Front End Developer	Manjot Sidhu	MS	November 12th,2020
Back End Developer	Benjamin Jenkyn	BJ	November 12th, 2020
Database Authenticator	Matthew Mukherjee	MM	November 12th,2020