KETCH

Beaton / Butler / Carder / Greeson

1 Definition

Ketch is a mobile app designed to facilitate transactions between two local parties for trading goods and services. It will utilize geolocation to create a local market derived from nearby users and allow them to buy and sell goods in a local marketplace. The program will emphasize security foremost, forcing the users to validate their identity, provide a secure transaction system, user reviews, and a safe meeting space for the actual transaction. The end result will be a more secure, safer way to buy and sell goods with far less fraudulent transactions occurring to the users.

Ketch's main goal will be to provide a safer way to buy and sell goods locally without having to carry large amounts of cash, provide a receipt of transactions and a rating system for accountability. It seeks to expand and improve on the basic formality of services like CraigsList and OfferUp and become the go-to place for local transactions.

2 Project Requirements

2.1 Primary

2.1.1 Functional

- Accept or deny current item offers.
- Message between user and buyer.
- Show items based upon location.
- Customer support functionality.
- Edit account information.
- Feedback/support system.
- Create and sell an item.
- Upload Pictures.

2.1.2 User Interface

- Item listing (where the seller points).
- Facebook login page.
- Account information.
- Individual item view.
- Loading screen.
- Favorites page.
- Chat system.
- Dashboard.

2.1.3 Usability

- Will follow general iOS design philosophy.
- No unnecessary functionality.
- Efficient and intuitive design.
- Simple to understand.

2.1.4 Performance

- Server must be able to handle consistent requests from many users.
- Database designed in an efficient manner.
- Mobile app must be lightweight.

2.1.5 System Interface

- Database talks to the mobile application and the website.

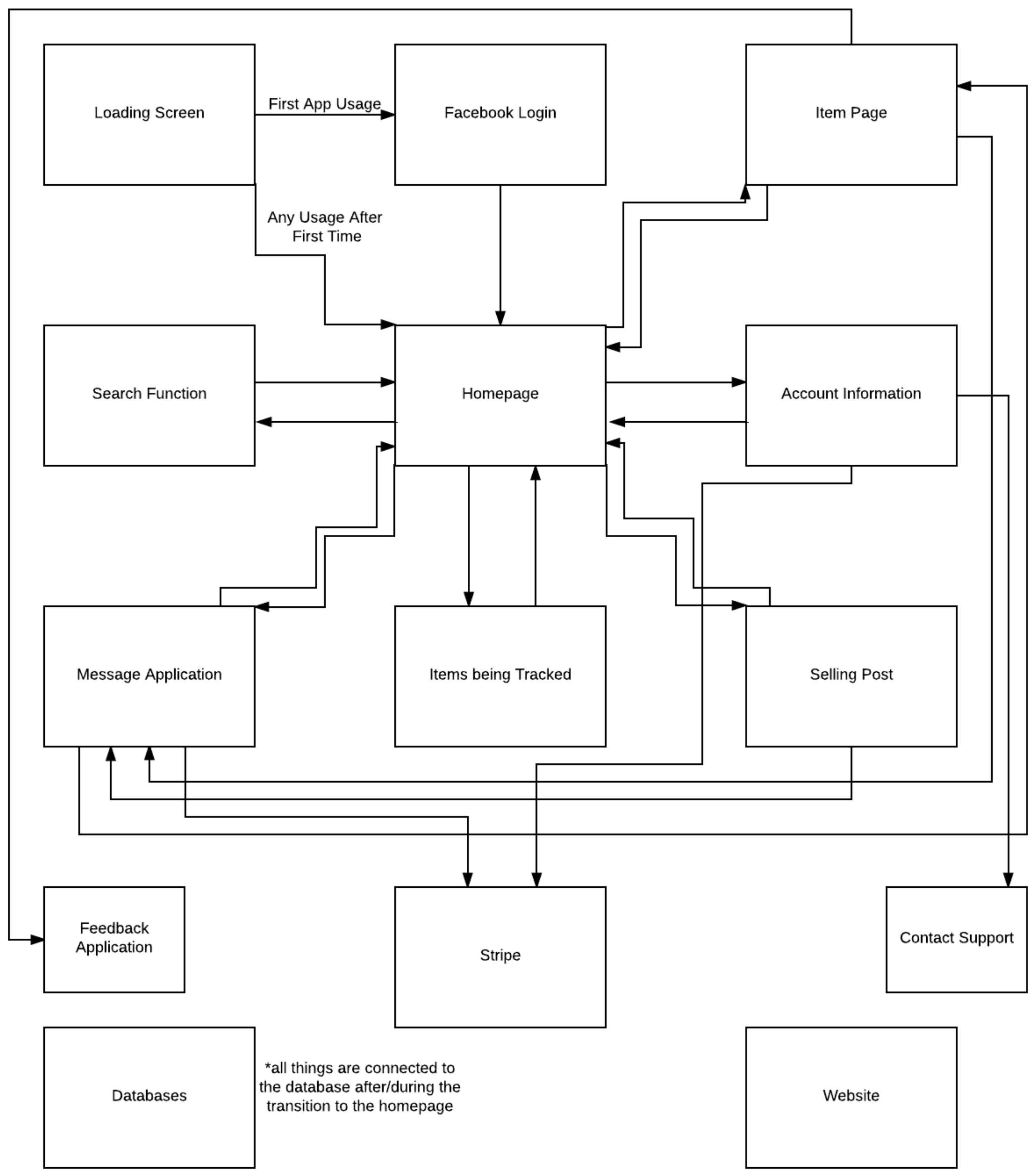
2.1.6 Security

- Payment system is done through Stripe.
- Facebook login to validate identities.
- Chat system is all done in-app.
- Only first names are given.
- User rating/report system.

2.2 Secondary

- Highly functional website.
- Geolocation for safe areas to meet.
- Be able to search specific categories.
- Be able to change view (items per page).
- Customize UI to user's liking.

3 System Model



4 Subsystems

Loading Screen: The loading screen will be a basic page that will be locally downloaded to the storage device. It will be immediately loaded upon the app opening and will remain on the screen until all the data can be retrieved from the server and be uploaded onto the homepage. Upon first usage, It will direct the user into the Facebook Login page. Anytime other than the first, it will load directly into the homepage.

Facebook Login: The facebook login is a function that will be added into the application for first time users. This applications use for this function is to create a secure environment for all the users, and to be able to link each user with a real person to avoid fake robotic accounts. This will help make users feel more comfortable with their transactions knowing that the person on the other side of the sell has to be a real person.

Homepage: The homepage is the basic center of our app that will connect to all other parts of the app. With this being the center, Every subpage of this center will have a link that goes back to the homepage. The homepage will consist of a search function, account information link, items being tracked link, a selling post link, item page connection, and message application link. It will at all times contain a 2 x 2 grid of items that you can choose from and click on. This will also have a exit out of app link.

Item Page: The item page can be reached by one of two ways. First it can be reached by clicking on an item from the homepage. The second will be by clicking the item description in the message application. It will have a picture or two of the item, as well as a description of the item and the first name of the seller. At the bottom of the page it will have a link to where you can message the seller about the product. It will also contain a link back to the homepage. This will also lead to a Feedback page, where you can give information on the seller anonymously.

Feedback Function: The feedback function allows buyers and sellers to tell others how the transaction went, if there was any kind of issue with the sale, or any other pertinent information that other users (or the Ketch team) may find relevant.

Search Function: The search function is just a basic search function that will not filter out items, but will try to find items based of off keywords in each post description.

Account Information: The account information will be a place that holds your email address, your zip code, a contact customer support feature and a items tracking list.

Contact Support: This is reached from the Account information page and it will open up a chat box on which you can contact customer support about any bug or bad transaction you have had that needs to be looked into.

Message Application: The message application will be a function that is very basic and will allow users to only send text messages and pictures to each other. The main meaning of this being in the application is that it allows for contact between the buyer and seller. Each party can talk to each other and send updated pictures on the item as well as a closer look at the item in question.

This will consist of two pages. The message homepage which will contain all of your messages with people and then the other page will be a page of the actual conversation between the buyer and seller. A link to this will be on the homepage. At the bottom of each individual message page, it will have a link to help find a safe place for the meeting to take place.

For the Buyer: When you enter an item page, it will say messager seller at the bottom of the page. When that link is clicked, it will link to your homepage of your messages which will list a picture along with the item description of each product. You click that product, then it takes you to a page where you can send a text to the seller.

For the Seller: When the seller gets a message, A notification will pop up the left hand corner of the selling post function. When they go to the selling post function, it they will click on the item with the notification and it will go to a message homepage that shows all the conversations he has about that item. He can then click on one of the conversations to message back or read the previous conversations he has had with a person.

Items Being Tracked: This function will display a page which details items that the user may want to keep an eye on. It works as a "favorites list" or a "watch list".

Selling Post: Every item posted will have an individual page with product images, a product description, a price, and any other relevant information that the seller wants to put in there.

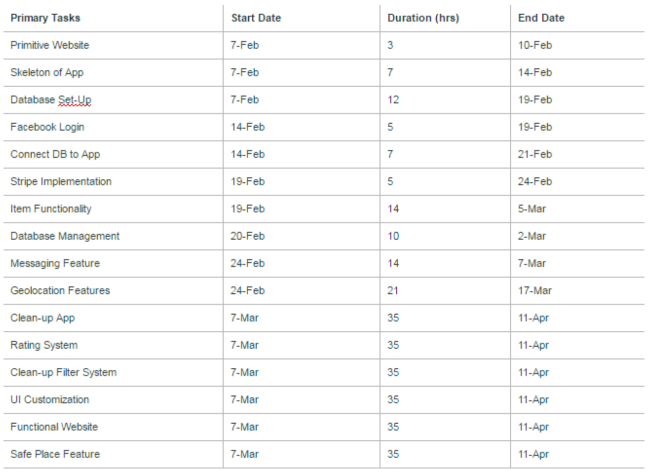
Stripe: Stripe will be used as a payment processor for our application to avoid meeting up with people while carrying a lot of cash. Once a price is agreed upon, the sale will go through Stripe. Both buyer and seller will agree on the price which will be finalized in our database. Before the payment gets sent to Stripe, both buyer and seller will have to select yes or no to the payment. We encourage this to be done in person before either person leaves the site of the exchange.

For the Seller: The seller will be able to request the payment through the messaging center. At the bottom of the screen there will be a button used to request the payment from the buyer. This will alert the buyer to confirm the sale.

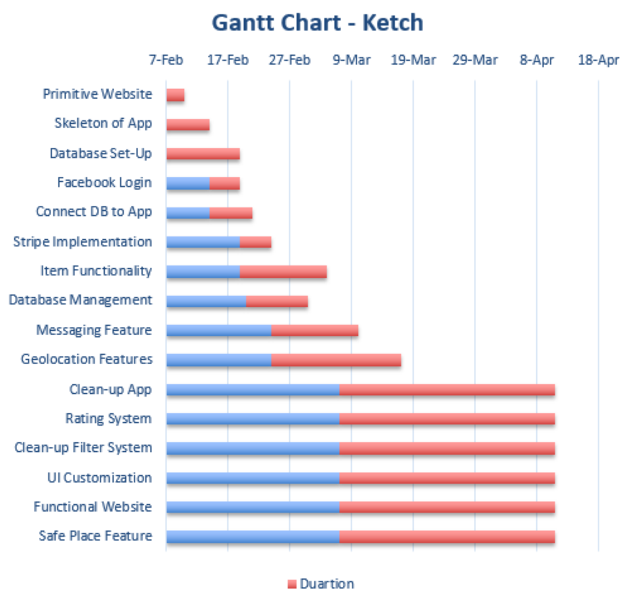
Database: The database will store all information such as individual user data, sales listings, user favorites list and chat histories. Every page in our application will be connected to our database in some way. Whether it's pulling a item listing with the picture, description, price, and other features, or displaying the user's account information, the database will be used.

Basic Website: Our basic website will serve to promote our application and show basic information that the user may want to know. Our primary focus will be in the mobile application, however we will embellish the website with more features if time allows.

5 Timeline



Primary Tasks	Start Date	Duration (hrs)	End Date
Primitive Website	7-Feb	3	10-Feb
Skeleton of App	7-Feb	7	14-Feb
Database Set-Up	7-Feb	12	19-Feb
Facebook Login	14-Feb	5	19-Feb
Connect DB to App	14-Feb	7	21-Feb
Stripe Implementation	19-Feb	5	24-Feb
Item Functionality	19-Feb	14	5-Mar
Database Management	20-Feb	10	2-Mar
Messaging Feature	24-Feb	14	7-Mar
Geolocation Features	24-Feb	21	17-Mar
Clean-up App	7-Mar	35	11-Apr
Rating System	7-Mar	35	11-Apr
Clean-up Filter System	7-Mar	35	11-Apr
UI Customization	7-Mar	35	11-Apr
Functional Website	7-Mar	35	11-Apr
Safe Place Feature	7-Mar	35	11-Apr



6 Alternative Models

6.1 iOS vs. Android

6.1.1 iOS

Pros:

- We have access to iOS devices which are connected to a mobile network.
- Xcode is a very intuitive and easy to use development environment.

Cons:

- iOS has to be developed on a mac environment. Only two of us have access to a personal mac computer.
- Requires a developer license to upload to the app store.

6.1.2 Android

Pros:

- Android can be developed on both Mac iOS and Windows PCs
- More versatile development environment.
- Uploading to the Google Play store does not require a developer license.

Cons:

- Android Studio is not as intuitive.
- We only have access to a few Android devices that are not activated.

6.2 Mobile Application vs. Web Application

6.2.1 Mobile

Pros:

- The application is best suited for on the go usage.
- Users can do card payments when they reach the meetup location.

Cons:

- Users must be connected to WiFi or a mobile network.

6.2.2 Web

Pros:

- A web-based environment may be easier for users who sell lots of items.

Cons:

- Harder to geolocate due to VPNs, proxies, etc. Also need users permission.
- Will not be developed natively for mobile devices, which may cause problems when users access the site using them.
- Harder to catch on in web form. Mobile is more popular.

6.3 Facebook login system vs. Local login

6.3.1 Facebook

Pros:

- Users are verified through Facebook.
- Increases accountability.
- Less information stored in our database.

Cons:

- Some users may not have a Facebook.
- Some users may not feel comfortable letting Facebook have access to their information.

6.3.2 Local

Pros:

- We don't have to rely on the Facebook API to retrieve information.

Cons:

- We cannot verify that the user is real.

6.4 Stripe vs. Venmo

6.4.1 Stripe

Pros:

- Students get no service charge for the first $1000 worth of transactions.
- Streamlined usability.

Cons:

- Security measurements do not seem as robust.

6.4.2 Venmo

Pros:

- Verifies the user's identity, checking against government watch-lists.

Cons:

- Requires users to have a Venmo account.

6.5 Using user's real names vs. usernames

6.5.1 Real Names

Pros:

- Users would be less likely to scam one another because there is more accountability.

Cons:

- Knowing someone's name may violate user privacy.

6.5.2 Usernames

Pros:

- It would introduce more anonymity in the application. Users would only know a pseudonym.

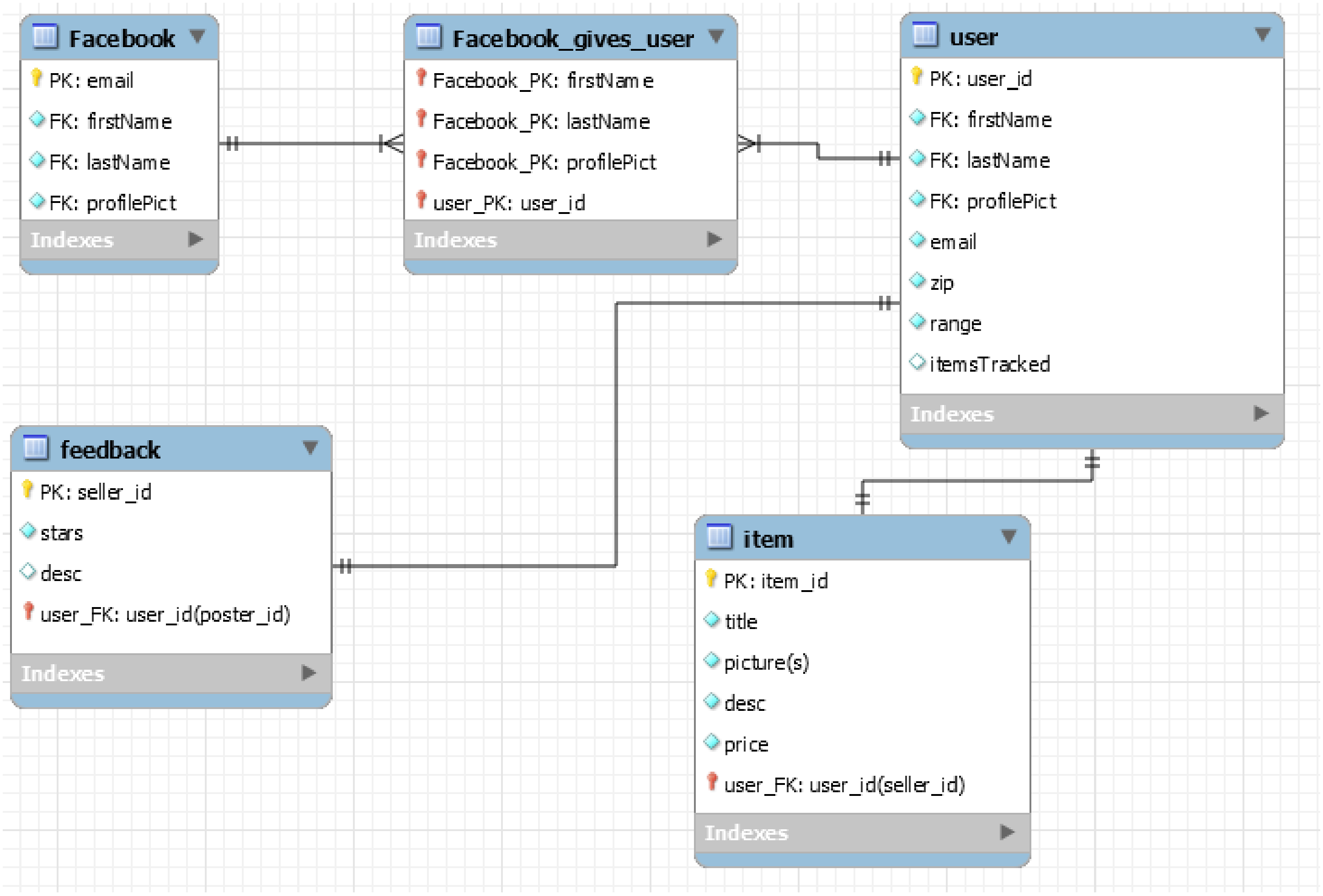
Cons:

- Security would be altered from one perspective to another. It would be more secure in the sense that individual users names are not being shared, however it is less secure because there is less accountability when not sharing real names between the users.

7 Group Roles

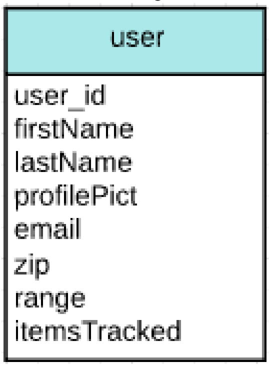
- **Connor** - UI/UX, Search Functionality
- **Sawyer** - Geo Tracking, Documentation
- **Keaton** - Database integration, UI
- **Patrick** - Facebook API, Stripe API, Lead programmer

8 Data Model



Facebook
PK: email
FK: firstName
FK: lastName
FK: profilePict
Indexes
Facebook_gives_user
Facebook_PK: firstName
Facebook_PK: lastName
Facebook_PK: profilePict
user_PK: user_id
Indexes
user
PK: user_id
FK: firstName
FK: lastName
FK: profilePict
email
zip
range
itemsTracked
Indexes
feedback
PK: seller_id
stars
desc
user_FK: user_id(poster_id)
Indexes
item
PK: item_id
title
picture(s)
desc
price
user_FK: user_id(seller_id)
Indexes

User Object



user_Id: A sequentially generated 8 digit number assigned to each user upon signing up for the app. Primary use is for the admins to navigate through users more efficiently. This field cannot be null, must be unique, and cannot be changed.

firstName: The first name is imported from Facebook. It is shown on a user's account information page as well as on a selling post, or in the messaging system. It cannot be null, does not have to be unique, and cannot be changed.

lastName: The last name is imported from facebook. It is shown on a user's account information page. It cannot be null, does not have to be unique, and cannot be changed.

profilePict: The profile picture is imported from Facebook. It is shown on a user's account information page as well as on a selling post, or in the messaging system. It cannot be null, does not have to be unique, and cannot be changed (unless the user logs out and re-logs back in after they have changed their profile picture on Facebook).

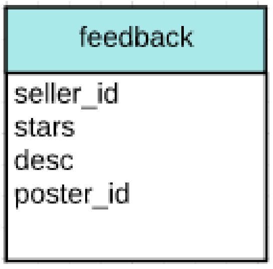
email: The user supplies this information. The email is meant to send notifications to the user about messages between the seller and buyer. There will be a check to make sure it is in the correct format when inputted. The email will only be seen on the account information page. The email value cannot be null, it must be unique, and it may be changed.

zip: The zip code is provided through the system using geo-location, but it may be changed if the user decides to look for items elsewhere. This is needed as a base to set the center of the radius when searching for items. The value cannot be null. Will be seen on the user account information page.

range: Range will already have values (multiples of 5 starting at 5) that the user may choose from. This sets the distance that a search will go out to, to look for items for sell. It cannot be null, and can be change.

itemsTracked: This will record all of the items that the user has favorited to be able to keep an eye on if the item sells or not. The user will be notified if the post is taken down. This will contain multiple items, it will constantly be updated, it can be null. And will be linked to the user account information page.

Feedback Object



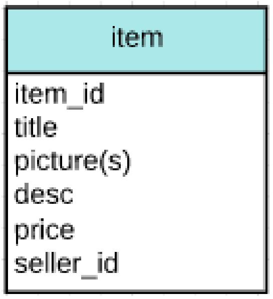
seller_id: This will be drawn from item page which drew from user. It will link the feedback post to the seller and will be posted on his profile page. Look to user object to understand this value.

stars: This will be a star rating system that each buy will give the seller. The stars will range from 1 star to 5 stars. This cannot be null and is base set to 1 star. This cannot be changed. It will also be seen with the description the buyer supplies.

desc: This will be the description area for the buyer to tell others about the transaction process went. The feedback system is only available to people who have actually purchased something from someone, and a person only has two weeks to write a review after the date of purchase. This field can be null, and cannot be changed. This will be seen on a seller's profile page is a buyer wants to see their reviews.

poster_id: This will be drawn from the user who makes the post. It will link the feedback to the user who posted. Look to user object to understand this value.

Item Object



item_id: The item identification will be a sequentially generated number that is a nine digit number. This value cannot be null, will be unique, and cannot be changed.

title: The title comes from the user's input when setting up an item for sale. This is just a short description of what it is. Tis value cannot be null, but it may be changed.

picture: The picture of an item is uploaded into the the database by the user. There may be up to 3 images per Item. Picture cannot be null, and the pictures may be changed.

desc: The description is provided by the user when setting up the item page. It cannot be empty and must contain at least 16 characters. It can be changed.

price: The price is a value provided by the user. It is created when the user sets up an item being sold. This value must be 0 or greater. The format will be Integer based, no doubles or floating numbers. The value is defaulted to zero, and can be changed by the user(seller only) at anytime.

seller_id: This value is the seller's first name, which is pulled from user. It cannot be null and cannot be changed.

9 System / Algorithm Analysis

9.1 Queries to the database

Our database is stored in a JSON file as a B+Tree. Database queries have a time complexity of O(logn). The space complexity of the database is O(n).

Insertion and deletion of entries are in O(logn).

9.2 Indexing attributes

We will index our user database based on the user_id. The user_id is an 8 digit number from 00000000 to 99999999. We will index our item database in the same way with item_id. The item_id is a 9 digit number from 000000000 to 999999999.

10 Glossary

Android: A Linux based operating system known mainly for its use on mobile devices.

API: An Application Program Interface(API) is a set of routines, protocols, and tools for building software applications. Used for programming UI's and tells software components how to interact with one another.

Apple: A company that creates software and hardware such as computers, phones, OS software or MP3 players.

Dashboard: The homepage that connects all of the other pages in the application.

Database: A structured set of data that is stored in a computer and can be referenced from outside that computer.

Function: The role or task of an object.

Geolocation: The identification or tracking of a real-world geographic location of an object.

iOS: Apple's operating system for all of it's mobile devices.

Linux: An open-source operating system with many types of distributions that all serve various roles.

Mobile device: A computer or phone that is able to move freely.

Mobile App: A mobile application that is designed to run on a mobile device.

Open-Source: The source code of a program is available to the general public to use or modify.

Operating System: The software that provides a computer's basic functions.

Party: A group of participants that are using the application.

Peer-to-peer: To connect from one person to another directly.

Rating: To give a numerical value to something or someone.

Search: To look for an object by means of a search function.

Server: A computer or computer program that manages access to a centralized resource or a service in a network.

Stripe: An application that allows for payment transactions to be made via the app without the use of cash or credit cards.

Subsystem: A self-contained system of a larger system.

System: A set of connected subsystems that all work together to perform a task.

UI: The User Interface is the medium in which the user interacts with the system, usually made up of graphics and buttons that perform tasks executed by the program.

User: A person using the application.

Username: A nickname that the user will be referenced by.

UX: The User Experience (UX) the user has with the application.

Venmo: A mobile application that allows a user to make peer-to-peer payment transactions.

Website: A series of pages hosted on the Internet accessed by users.

Web: The Internet otherwise known as the World Wide Web which is a giant connection of networks.

11 Amended Models

11.1 SQL to Firebase

Our original database plan was to utilize a SQL-based database. The basic structure was implemented but we were having trouble with many things, such as linking it to the application.

We discovered Firebase shortly thereafter which utilizes a real-time NoSQL database and switched over. Firebase was designed with mobile development in mind, and provided countless other features such as free storage, real-time analysis, easy API functions for Facebook authentication, a cloud-messaging structure and testing lab.

The key to Firebase's immediate value is its real-time database that stores data in JSON. Any change in the database is immediately synced across all devices that use the same database. Firebase is extremely fast and everything is updated instantly. Switching to it was the correct choice.

11.2 Minor Changes

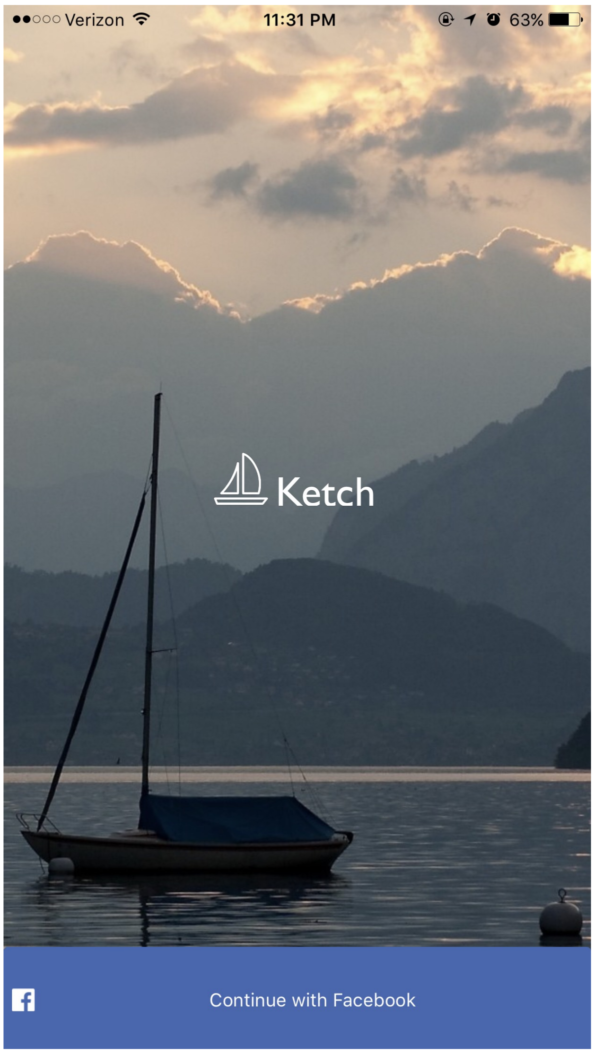
The following are a list of minor system changes that do not warrant their own subsection:

- Added in a check-in system to to test if an item has been sold.
- Linked feedback to items as well as users.
- Linked Stripe transactions to users.
- Reworked time complexity.
- Expanded on administrator-side queries.
- Added the date of post and zip-code to items.

12 Design and Controls

12.1 UI

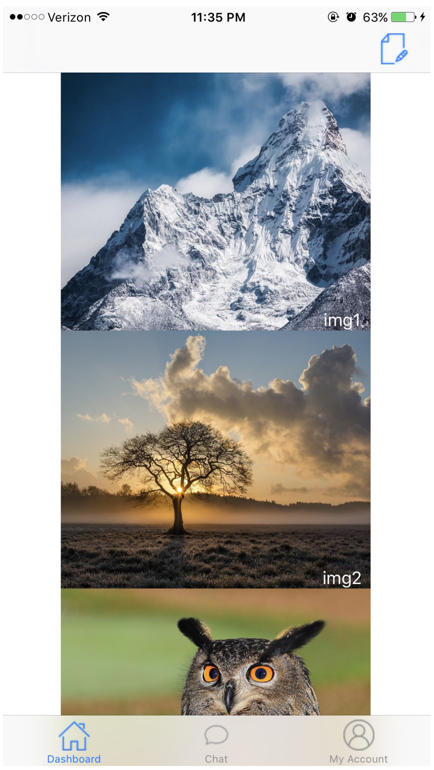
12.1.1 Log-in Screen



Performs Facebook authentication with user. Checks to see if there is a pre-existing account and if not creates one with a unique user ID. If the user does not have Facebook, they cannot use the application.

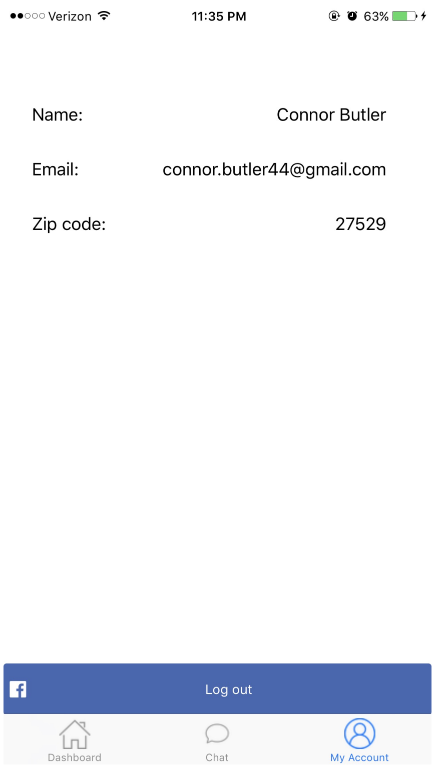
The UI is designed to be minimalistic, with as little clutter as possible. This ensures a streamlined experience for the user to buy and sell goods quickly and efficiently. The sailboat or nautical theme pertains to the title of the application, as a ketch is a type of sailboat, but is also a homonym for the word "catch" which can allude to the user catching a great local deal.

12.1.2 Home Screen



The home screen will show a list of local items (using geolocation) hopefully using an infinite scroll where the list will be populated with new items as the user scrolls down. There are links at the bottom for access to the chat feature and the user account. At the top right corner is an icon for posting a new item. Again, the focus is on a minimalistic design. The user should have no difficulty navigating the UI.

12.1.3 My Account Screen



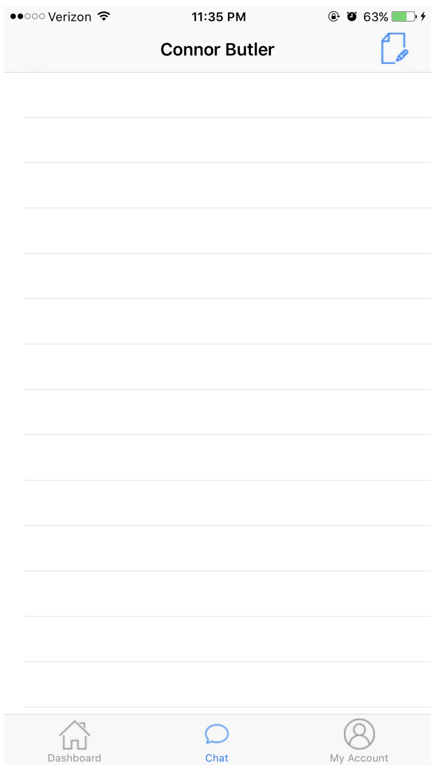
Name: Connor Butler

Email: connor.butler44@gmail.com

Zip code: 27529

The current account screen pulls and displays user information from the database. At the moment it only displays name, e-mail and zip code but in the future it will include a list of items the user is currently selling, feedback, and more.

12.1.4 Chat Screen



This will be a list of chats the user has with other users. At the moment it is not populated as the cloud chat engine has not been implemented, but it will be in the future. The chat engine is the Firebase Cloud Messaging (FCM) service which is based on Google Cloud Messaging. It is a cross-platform cloud chat service which can send and sync messages in real-time across many different platforms (iOS, Android, etc).

12.1.5 Future UI Plans

- Payment page with Stripe authentication
- Watch List page for users to watch items
- Feedback page to allow users to submit reviews.

12.2 Framework

Ketch is using the Firebase framework which itself implements many different public SDK frameworks and APIs.

13 Coding Models

13.1 Approach

- **Object Oriented -** Using Swift which is an OO based language.

- **Research -** Before coding, the next module is researched to make sure it is being done in the most correct and efficient fashion.

- **Coding -** Develop a code skeleton, keep code clean, make new methods and functions when needed and make sure variable names are easy to follow.

13.2 Team Contributions

- **Connor -** UI/UX, Login Page, Admin Panel, Messaging Functionality

- **Keaton -** Database Integration, UI, Stripe Implementation

- **Patrick -** Item Posting, Dashboard, Search Functionality

- **Sawyer -** Geo-Tracking, Documentation, Rating System

13.3 Frameworks

- **Firebase -** Firebase Real-time Database, Firebase Authentication

- **CocoaPods -** Dependency Manager for Swift/Objective-C. Quickly and effectively update and install frameworks.

- **Facebook SDK -** Framework for Facebook that allows us to easily connect/use user info.

13.4 Languages

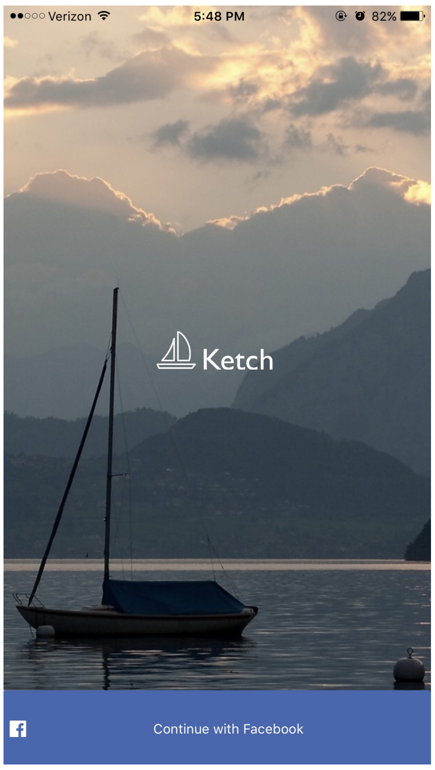
- **Web -** HTML, CSS, JavaScript

- **Mobile -** Swift, Objective-C

13.5 Subsystems & Progress

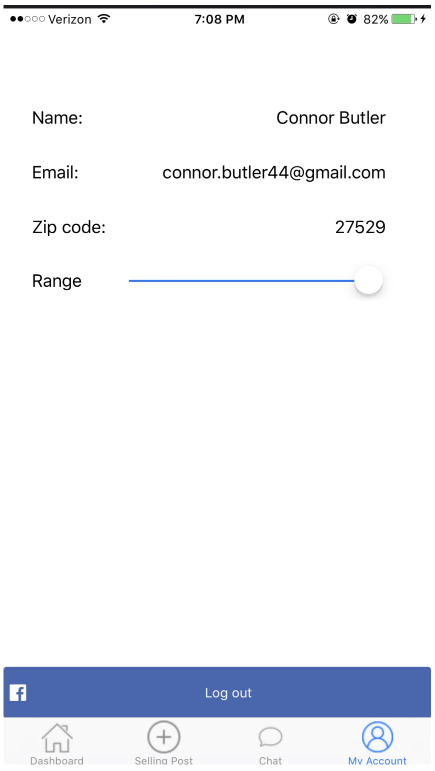
13.5.1 Login (completed)

Pulls user data from facebook and Authenticates/stores the user with Firebase successfully.



13.5.2 Account Information (80%)

Pulls user data from Firebase. Allows user to set Zipcode. The user can logout of his/her account. Still need to implement range setting for the user to find items.



Name: Connor Butler

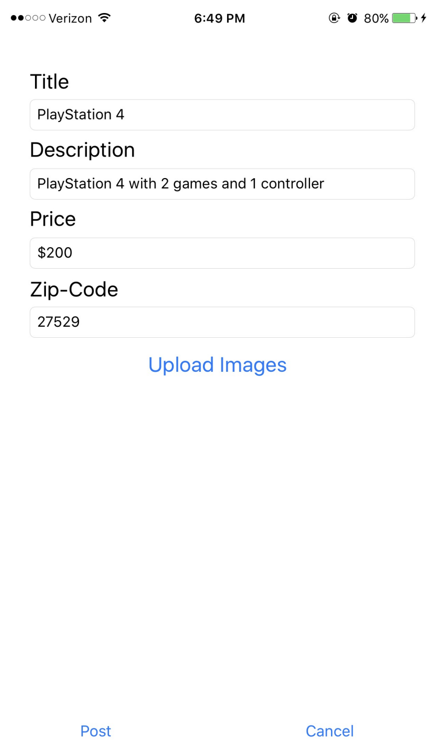
Email: connor.butler44@gmail.com

Zip code: 27529

Range

13.5.3 Posting an Item (70%)

User can set a Title, Description, Price, and Zipcode of the item. Need to implement a limit on the number of Title and Description characters. Picture uploading is the next task to be completed. It can use an image stored on the user's phone or the user can select to use the camera to take a new photo. A flag for whether or not the price is negotiable will be added as well.


●●○○○ Verizon 6:49 PM 80%

Title
PlayStation 4

Description
PlayStation 4 with 2 games and 1 controller

Price
$200

Zip-Code
27529

Upload Images

Post Cancel

13.5.4 Dashboard (15%)

Need to display items in a collection view. Need many updates to improve UI style and flow.

13.5.5 Individual Item Page (15%)

Need to be able to pull the item ID from the dashboard. Need to group items to a user. Need to link the chat function to the seller ID so chats can be initialized.

13.5.6 Stripe (0%)

Stripe Implementation was waiting on both the account page and messaging system to be set up. Now that those are mostly completed, Stripe functionality will be the next component added to the application.

13.5.7 Geolocation (0%)

Geotracking will be added during the next two weeks. It will take some time to make sure items get posted into the correct zip-code . Users will be searching for items within specific zip-code areas.

13.5.8 Search System (40%)

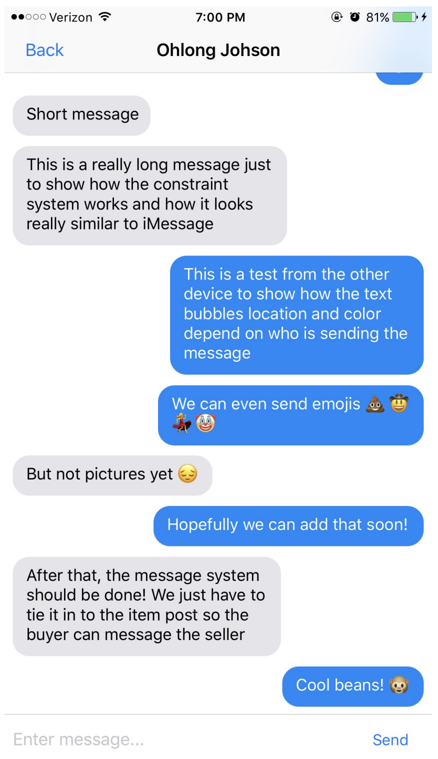
A search algorithm needs to be added as well as an actual search bar.

13.5.9 Rating System (0%)

The Rating System implementation cannot take place until Stripe is implemented. The rating process takes place after the purchase so there will be no access to this until after the final transaction has taken place.

13.5.10 Messaging System (90%)

Will be complete once the implementation of uploading a picture through the message feature is added.


●●○○○ Verizon 7:00 PM 81%

Back Ohlong Johson

Short message

This is a really long message just to show how the constraint system works and how it looks really similar to iMessage

This is a test from the other device to show how the text bubbles location and color depend on who is sending the message

We can even send emojis 💩 🤠 💃 🤡

But not pictures yet 😔

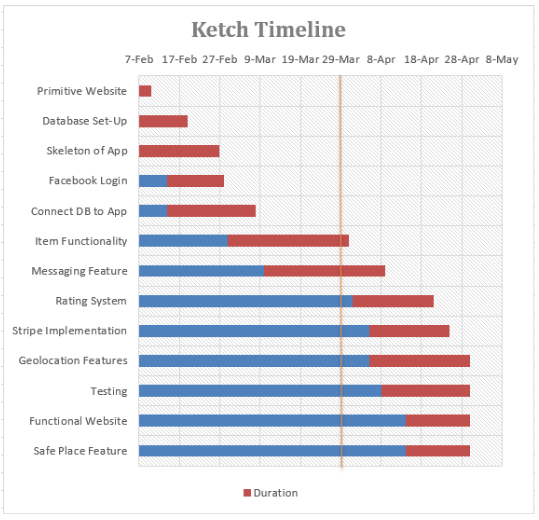
Hopefully we can add that soon!

After that, the message system should be done! We just have to tie it in to the item post so the buyer can message the seller

Cool beans! 🐵

Enter message... Send

13.6 Updated Timeline



14 Project Testing

14.0.1 Login System

- Register lots of users concurrently with no errors.
- Ensure correct user is being identified when logging in with various different accounts (Facebook Authentication).

14.1 Dashboard

- Pull item posts with various search constraints while being updated.
- Ensure continuous scrolling does not introduce memory leaks.
- Ensure continuous scrolling does not introduce memory leaks.

14.2 Messaging

- Messages appear correctly while receiving and sending messages.
- Ensure continuous scrolling does not introduce memory leaks.

14.3 Validation

- Ensure user is never changed, logged out, or not authenticated with constant usage of the application.

14.4 Geolocation

- Ensure location is being reported correctly and if location can not be determined, the last known location will be saved and used.

14.5 Security

- Test extensively for security holes in the login, messaging and posting systems.
- Make sure certain user information such as location, etc is not accessible to other users.
- Make sure the messaging system is encrypted.

15 User Training (NEW)

Ketch's goal is to use a simple, intuitive UI to facilitate ease of use. As such, it should not require any kind of in-app tutorial or tooltips.

A user guide for the application will be hosted on the Ketch website and will include instructions with screnshots on how to buy and sell items along with any necessary subsystems. The app can launch the help section of the website in a browser via a help button on the application homepage.

15.1 Team Contributions II

- **Connor -** Messaging, UI, Debugging
- **Keaton -** Stripe Functionality, Website Update (Help page and database link)
- **Patrick -** Photo Uploading, Item Functionality, Dashboard
- **Sawyer -** Documentation, Rating System, Unit Testing

References

[1]"Features - Firebase", Firebase, 2017. [Online]. Available: https://firebase.google.com/features/. [Accessed: 08- Mar- 2017].

[2]B. Lim, "Introducing Cloud Functions for Firebase", The Firebase Blog, 2017. [Online]. Available: https://firebase.googleblog.com/2017/03/introducing-cloud-functions-for-firebase.html. [Accessed: 10- Mar- 2017].

[3]M. Maher, "Introduction to Firebase: Building a Simple Social App in Swift", Appcoda.com, 2016. [Online]. Available: https://www.appcoda.com/firebase/. [Accessed: 09- Mar- 2017].