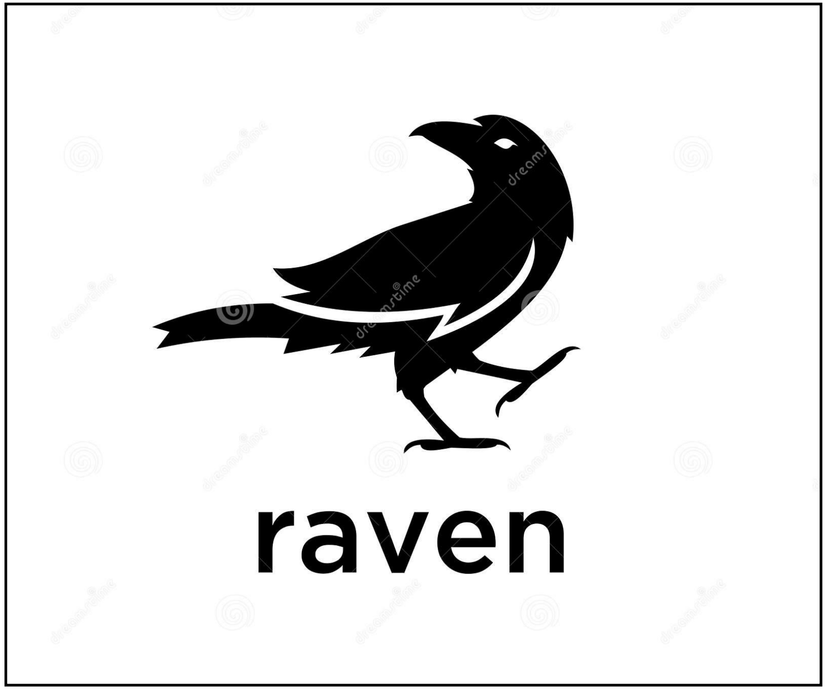
Raven



- Group 1: MASST Inc.
- Meetings: Friday 11am in ECSW

Team Profile

Alfredo Mejia, BS SE
3rd Year
C++, Java
Full-Stack Developer

Carson Sharp, BS CS
2nd Year
Java, C++
Back-end Developer

Natasha Trayers, BS SE
2nd Year
Java, C++
Back-end Developer

Team Profile Pt. 2

Mohammed Alotaibi, BS SE
3rd Year
Java, C++, Python
Backend developer

Mustafa Sadriwala, BS CS
3rd Year
Java, C/C++, JS
Full-Stack Developer

High Level Overview

- This mobile application will assist students of The University of Texas at Dallas (UTD) in receiving alerts that keeps them informed of important events happening nearby and make it easy to spread the word quickly when something significant happens as well as allow students of UTD to have their own local online marketplace, where they get to buy and sell items.
- This application has a high business value because it will be used the majority of the year and it is flexible to change. It is a part of UTD’s mission to create a more inclusive and connected community on campus and will allow them to market their university as one of the first to implement such virtual neighborhood applications.

Inception of Raven

- We selected this project idea because we think this will be an effective way to connect students in universities and create a sense of community
- This application will make it easy for students to find local resources, ask other students for recommendations, and alert one another of events such as crime
- It will also allow university officials to connect with students so they are aware of the issues that matter to students

Functional Requirements

- User shall be able to register an account with UTD SSO
- User shall be able to login and logout of the app
- User shall have an interactive walkthrough of steps and functionalities when they login for the first time
- User shall be able to add tags to the bottom of posts using ‘#’ character
- User shall be able to select visibility for the news, alerts, and messages that they post
  - User shall be able to choose to post to friends, campus, or current location
- User shall be able to add and accept friend requests
- User shall be able to create friends circles/cliques/“conspiracy”

Functional Requirements

- User shall be able to privately message individuals
- User shall be able to send and accept private message requests to and from users who have not accepted friend requests
- User shall be able to search through posts using tags.
- User shall be able to view most recent posts or top posts in social media style feed on the home page
- User shall be able to sort and filter through posts using Username, Location, Type Of Post, Date, and Popularity
- User shall be able to like or dislike posts
- User shall be able to report inappropriate content
- User shall be able to like or dislike comments

Functional Requirements

- User shall be able to view and update their profile information
- User shall be able to see their previous posts on their profile
- User shall be able to make posts and categorize them into general posts, items for sale, or alerts
- User shall be able to make comments on posts
- User shall be able to save/bookmark items for sale
- User shall be able to browse items for sale by categories
- User shall be able to switch views between their profile, private messages, public posts, and marketplace
- User shall be able to upload photos from camera roll
- User shall be asked for permission for notifications, location services, and camera usage

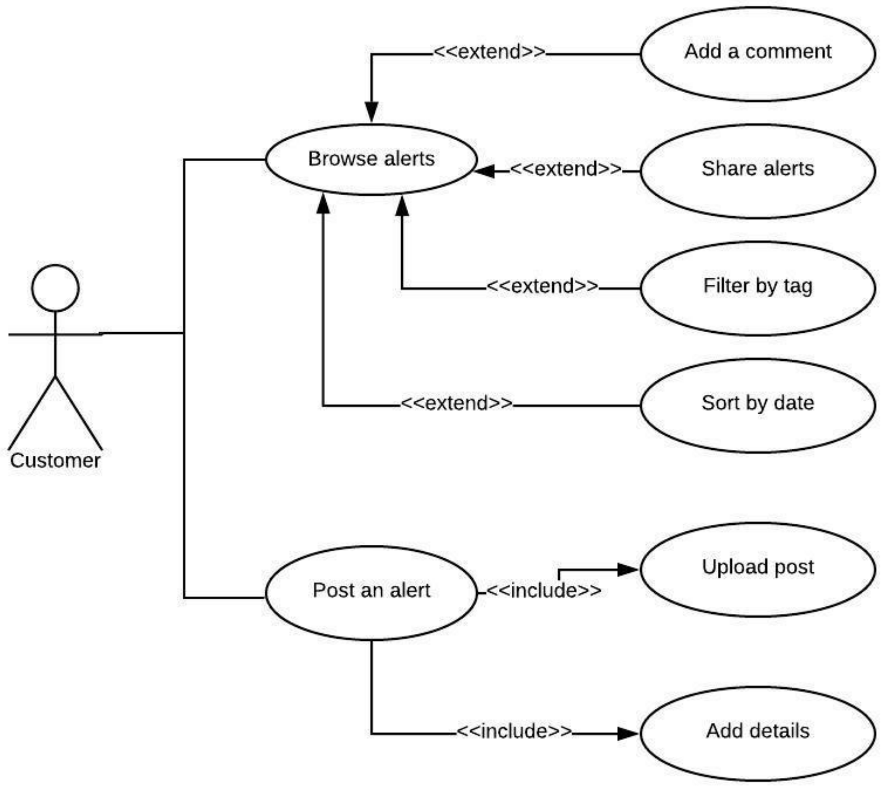
Non-functional Requirements

- App should load within 3 seconds
- Users should be able to login with UTD SSO within 5 seconds
- The account should lock after 5 failed attempts
- Our mean time between failures (MTBF) should be over 24 hours
- Data should be handled in accordance with Fair Information Principles
- Maintenance - during the project scope - should respond to issues within 12 hours

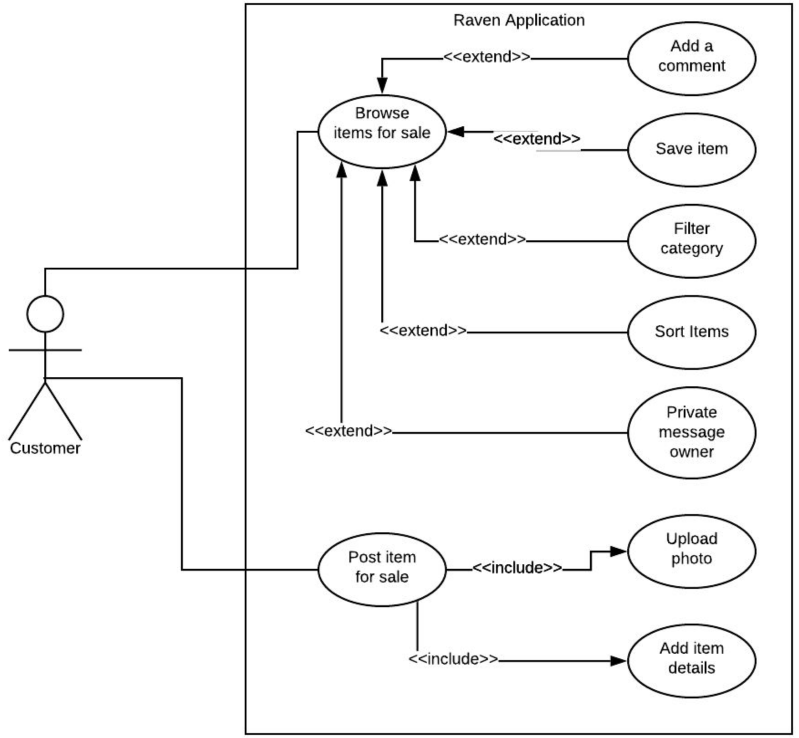
Non-functional Requirements

- User should be able to post to the app within 5 seconds
- Data should be backed up every 6 hours during 8am - 8pm and once between 8pm - 8am
- The app should not be ‘down’ for more than 3 hours at a time
- The app should always be available for usage unless brought ‘down’ for maintenance or critical bug
- Server should be able to support up to 50KB of data per post
- Server should be able to support up to 500KB of data per item for sale

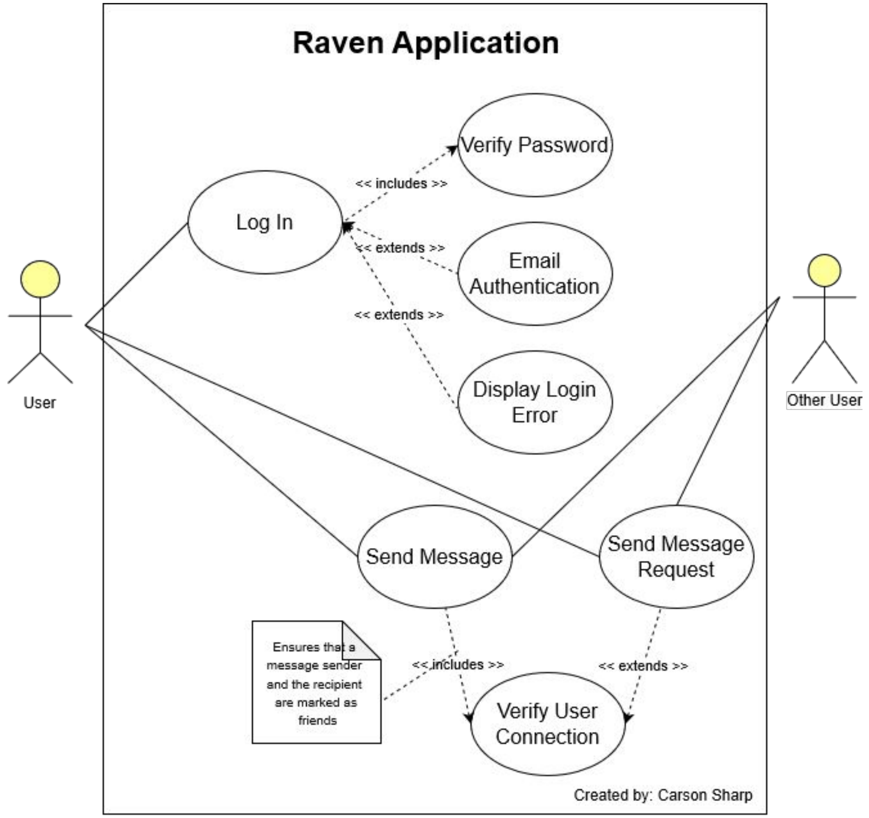
Alert Section Use Case



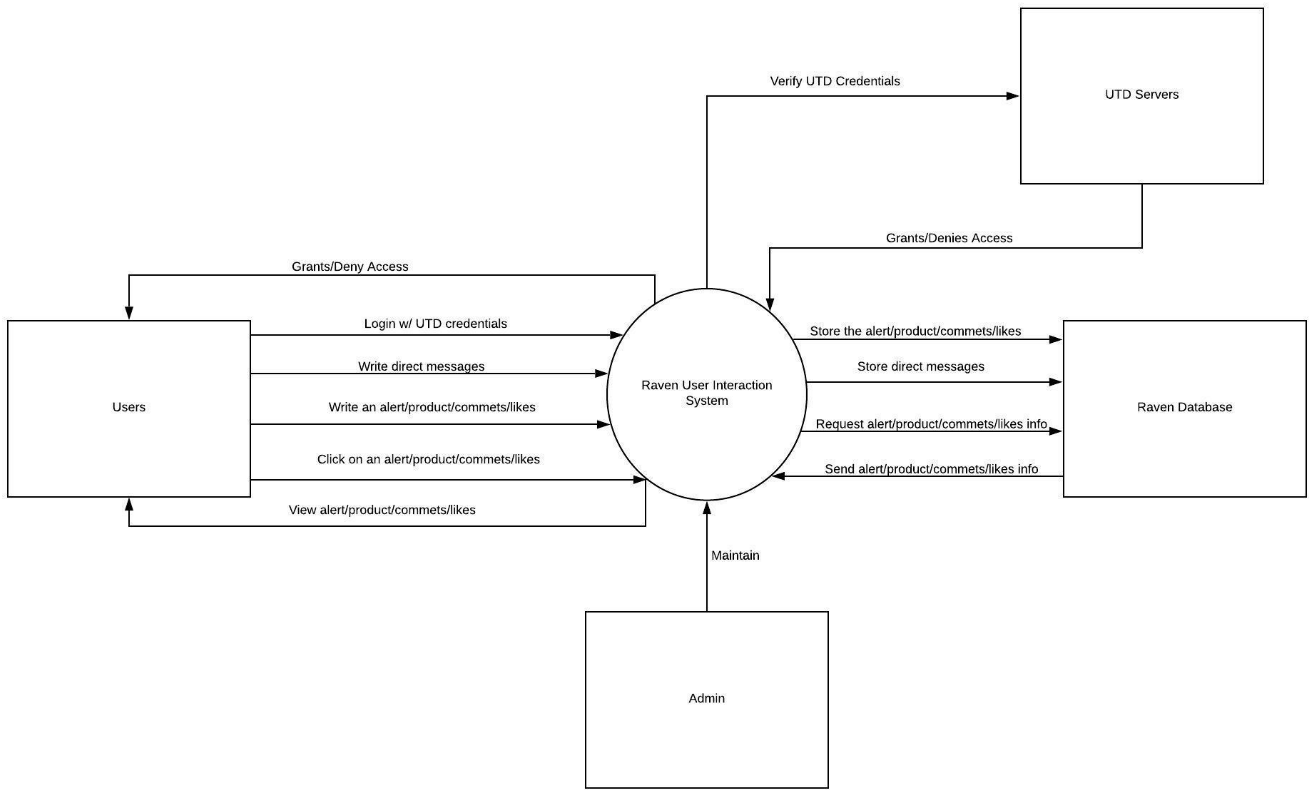
Market Place Use Case



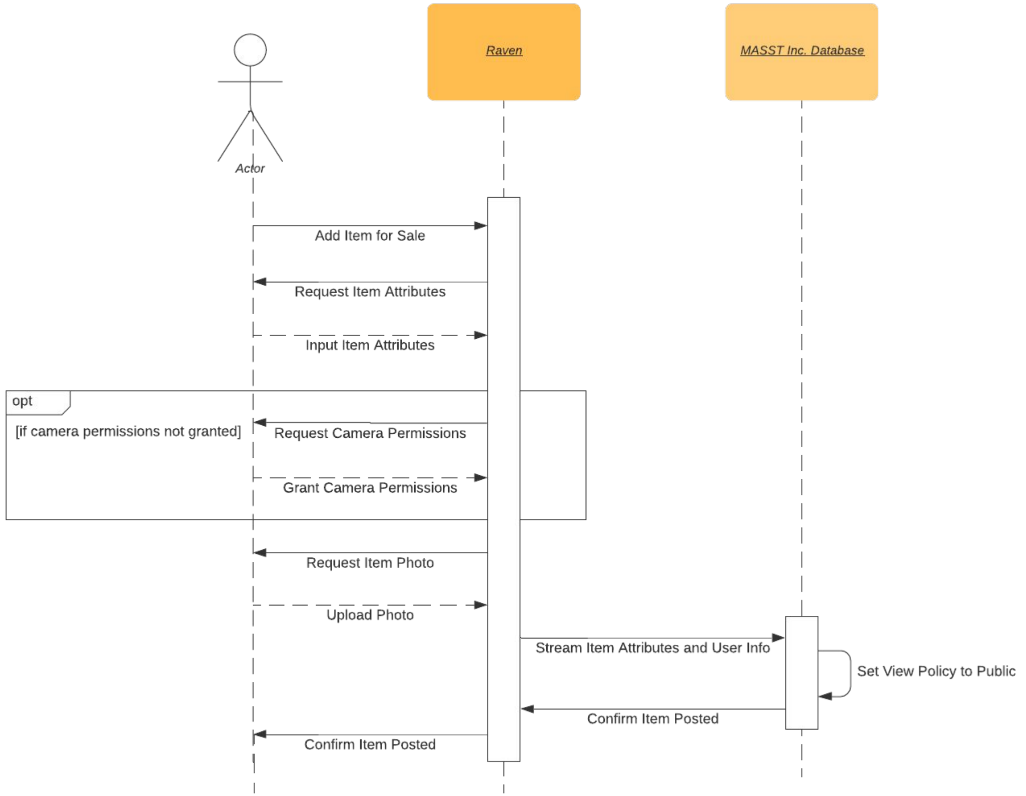
Login and Send Message Use Case



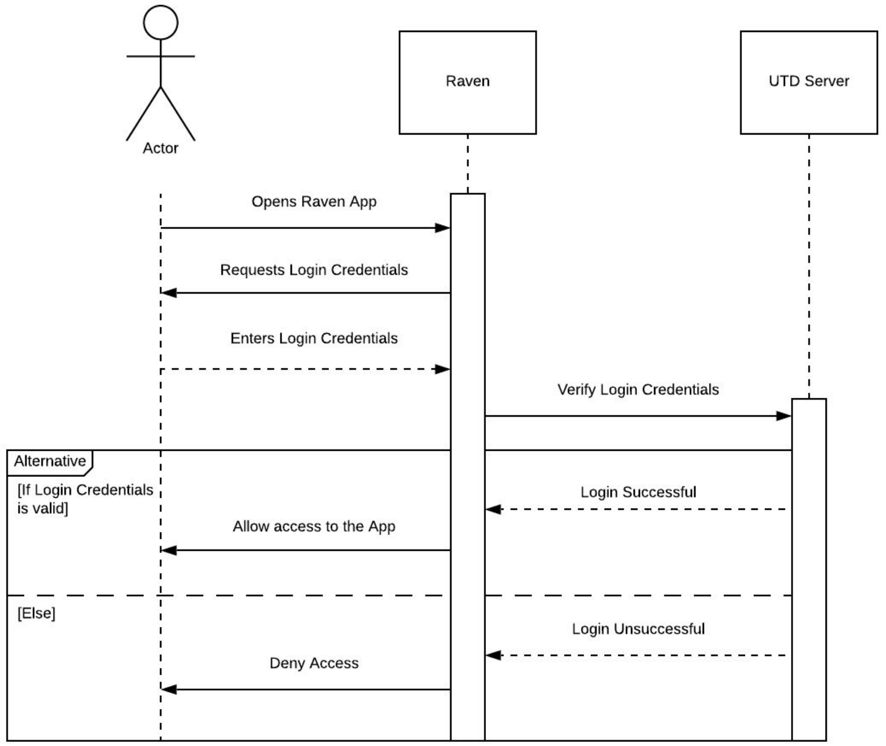
Context Diagram


Verify UTD Credentials
UTD Servers
Grants/Denies Access
Grants/Deny Access
Login w/ UTD credentials
Write direct messages
Write an alert/product/commets/likes
Click on an alert/product/commets/likes
View alert/product/commets/likes
Users
Raven User Interaction System
Store the alert/product/commets/likes
Store direct messages
Request alert/product/commets/likes info
Send alert/product/commets/likes info
Raven Database
Maintain
Admin

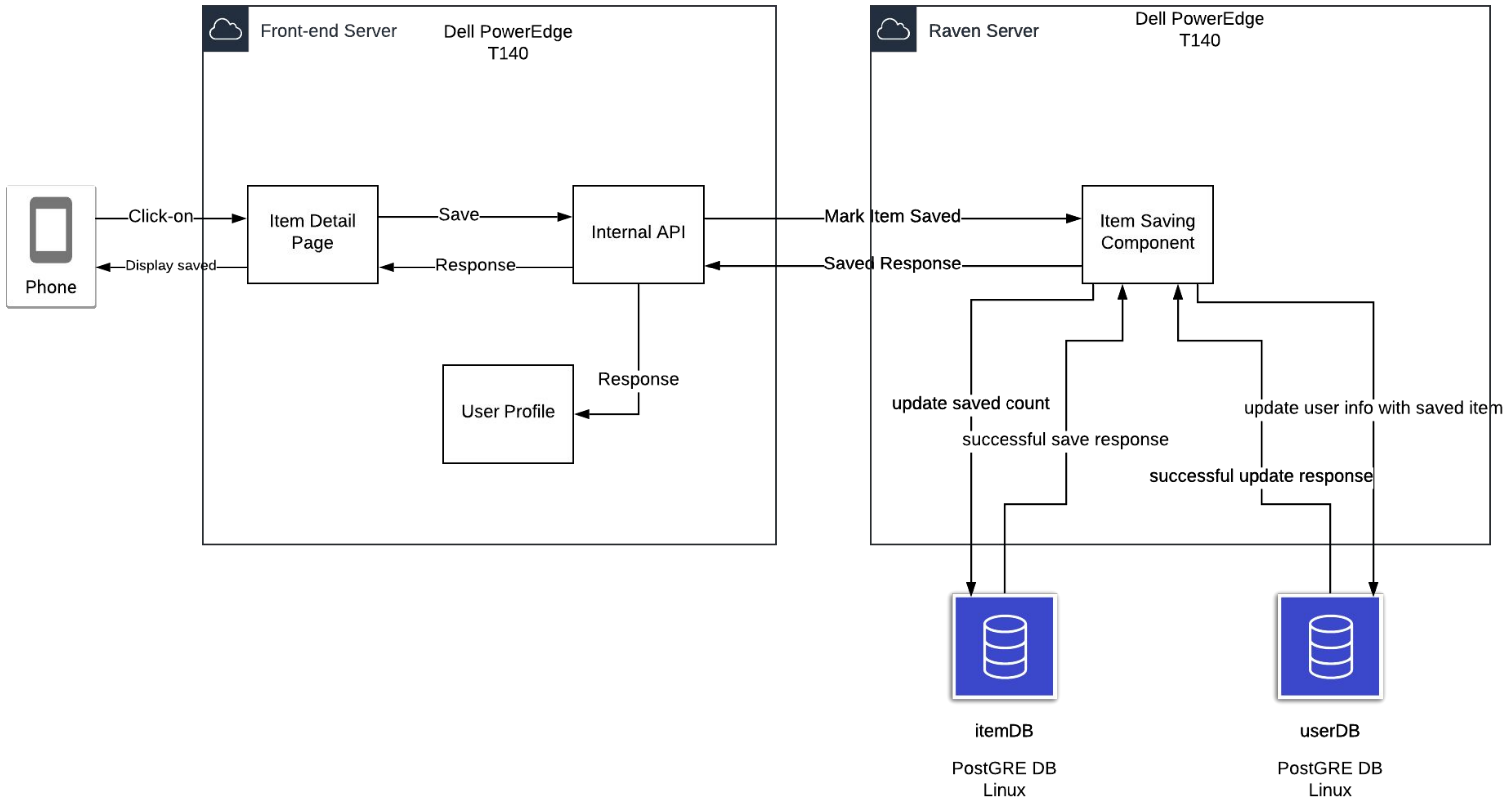
Add Item for Sale Sequence Diagram



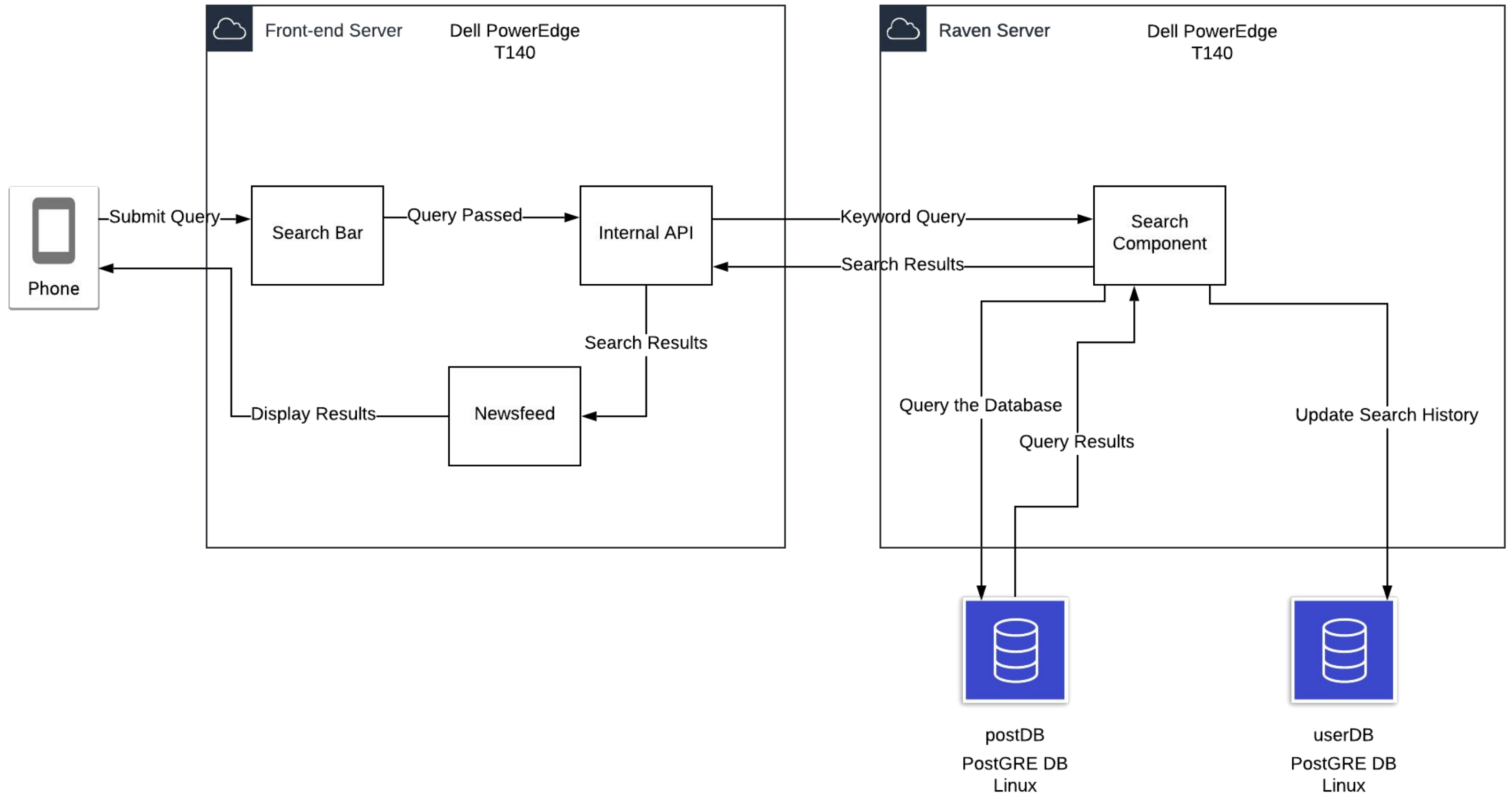
Login Sequence Diagram



Save an Item (Mid-Level Diagram)



Search posts (Mid-Level Diagram)



Coding Technologies

- Front End: React Native
  - Modular Architecture
  - Code Reusability
  - Handy Solutions & Libraries
  - Cost Effective
- Middle End: Express.JS
  - Stable, secure, quick
  - Constrained to the languages we are familiar with
- Back End: MySQL & SQLite
  - MySQL: Fast, secure, scalable
  - MySQL: Compatibility with React Native
  - SQLite: Compatible with MySQL & Reactive Native
  - SQLite: Meets security and scalability requirements

Testing Process

Testing will be broken into phases corresponding to different phases of development:

1. Regression Testing
    a. Unit Testing
    b. Component Testing
    c. System Testing
2. Scenario Testing & Performance Testing
3. Alpha Testing & Beta Testing
4. Acceptance Testing

Requirements Traceability

Testers will be required to sign off on individual tests linking them to the requirements they originate from. If tests can no longer be linked to a requirement, tester should inform team lead to fix any discrepancies such as a changed or removed requirements that can cause a testing failure.

Testing Technologies

Jest

- React Native testing
- fast, reliable, compatible
- for JavaScript codebase

Sinon

- For Express.JS testing
- clean integration, minimal setup

Docker

- MySQL Database testing
- ability to package units into standardized applications called containers
- simple to test individual components and functionality
- continuous integration tools, minimal setup

Sample Functional Test Cases				
#	**Name**	**Corresponding Requirement #**	**Execution**	**Acceptance Criteria**
F1	Comments on Post Test	19	Input a string as a comment to a post	1. The UI of the post will change to indicate it has a comment 2. All users should be able to see the comment on the post
F14	Privately message individuals	8, 9	1. Privately message an individual that is already on the friends list 2. Privately message an individual who is not on the friends list.	1. The other user should receive the message in the chats section and get a notification 2. The other user should receive a message request from the individual in the notification section
F7	Like and Dislike Posts	13	A user is able to press on the thumbs up and thumbs down button on a post.	1. The UI of the post will change to indicate it has a like or a dislike 2. All users should be able to see the like or dislike on the post
F10	Categorize items for sale	21	1. Click on search bar in items-for-sale feed 2. Click a category to filter by 3. View items-for-sale feed	1. The items-for-sale feed should only show items tagged under the chosen category 2. All previous sort features should still apply to the items-for-sale
F16	Log In and Log Out Test	2, 3	1. New account is registered using a UTD SSO. 2. Walkthrough is completed by input. 3. The user is logged in and out repeatedly.	1. Walkthrough is given on the first successful login attempt 2. All proceeding logins are successful and the walkthroughis never shown.

Sample Non-Functional Test Cases

#	Name	Corresponding Requirement #	Execution	Acceptance Criteria
NF3	Downtime Test	9	1. Run the application, server, and database continuously for at least 100 days and track failures and the time needed to restore the system	1. The whole system should be fully restored within 3 hours of the crash of the system.
NF5	Scalability Test	13	Simulate 1000 virtual users logging into the app concurrently	The application should be accessible with minimal delay in response time. Maintain an average transaction time of five seconds at 1000 simultaneous users.
NF6	App Load in 3 Seconds	1	When activating the app, the home screen must be shown and ready for usage in around 3 seconds or less.	The user will be able to access the app in 3 seconds from opening it.
NF1	User Post Speed	7	1. Make a post with an empty string and no picture 2. Make a post with a picture of max dimensions and resolution 3. Make a post with a picture of max string length 4. Make a post with both max string and max picture size	1. All posts should be received by database. 2. User should receive confirmation of post being uploaded within 5 seconds of pressing Submit 3. Post should be viewable to other within 5 seconds of original user pressing Submit
NF8	Lock Out Test	3	1. Attempt to log in to a sample account using an incorrect password repeatedly. (Can test case sensitivity with password choice)	1. Login process is halted after the fifth automated attempt.

Deployment

Sprint Deployment: Since we will be using Agile methodologies in development we will be deploying at the end of each spring

Project Deployment: At the end of the project, we will build the application and confirm with the customer that it is acceptable for final release. The application will be compiled for native Android and iOS builds and then deployed onto the respective app stores. The back-end and front-end codebases will be deployed onto DELL PowerEdge T140s

Deployment

Potential Issues and Resolutions: The installation program may not be built correctly which could lead to a missing or corrupted script being installed. The installation process should be tested and inspected to ensure deployment is accurate and complete.

Inconsistencies could arise between the iOS and Android versions of the application. The deployment process to both platforms should be carefully done in parallel to each other.

Effective communication between the deployment team and the client over all documentation and maintenance team will be necessary to ensure a lack of communication does not lead to confusion.

Maintenance

Regular Maintenance:

- Servers will be monitored by performance tests and analyzing for any security threats. Maintenance team will regularly update server to enhance performance and security.
- Databases will be monitored by performance tests, storage level, security threats, and back-ups. When appropriate the maintenance team will update database software, write back-ups, and flush out cache memory.
- Mobile operating systems will be monitored to ensure compatibility of our solution with any mobile OS update.
- Mobile application will be monitored through different tests and customer reports. The maintenance team will ensure any bugs, errors, security threats, and performance leaks are resolved.

Maintenance

Human Resources:

- Technical expertise in the following fields/technologies:
  - React Native
  - Database Analyst (preferably for Mobile technologies)
  - Mobile Application Architectures (Express.JS, React Native)
- Maintenance team should have general proficiency with JavaScript and MySQL

Technical Resources:

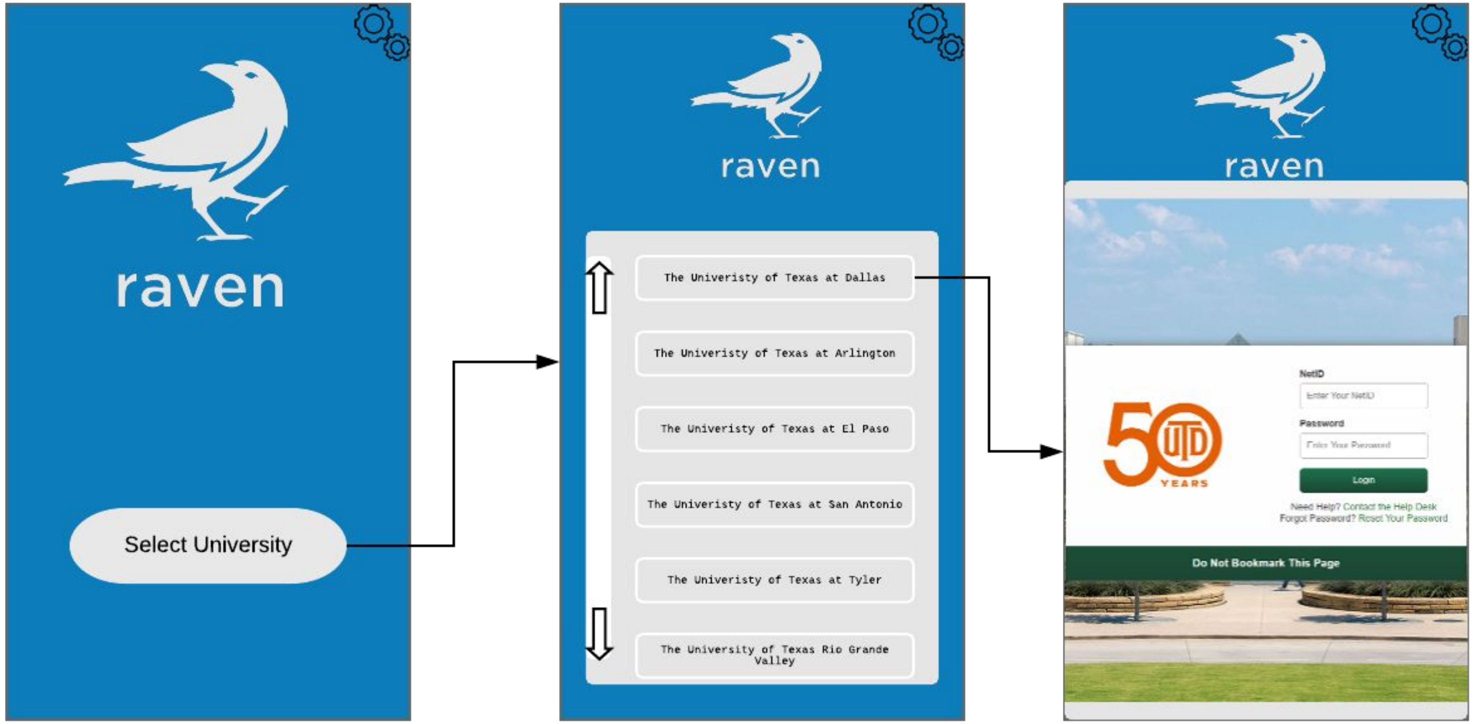
- Suite of tools for identifying and locating critical pieces of code quickly
- Various applications for analyzing effects to changes in specific slices of the codebase

Maintenance

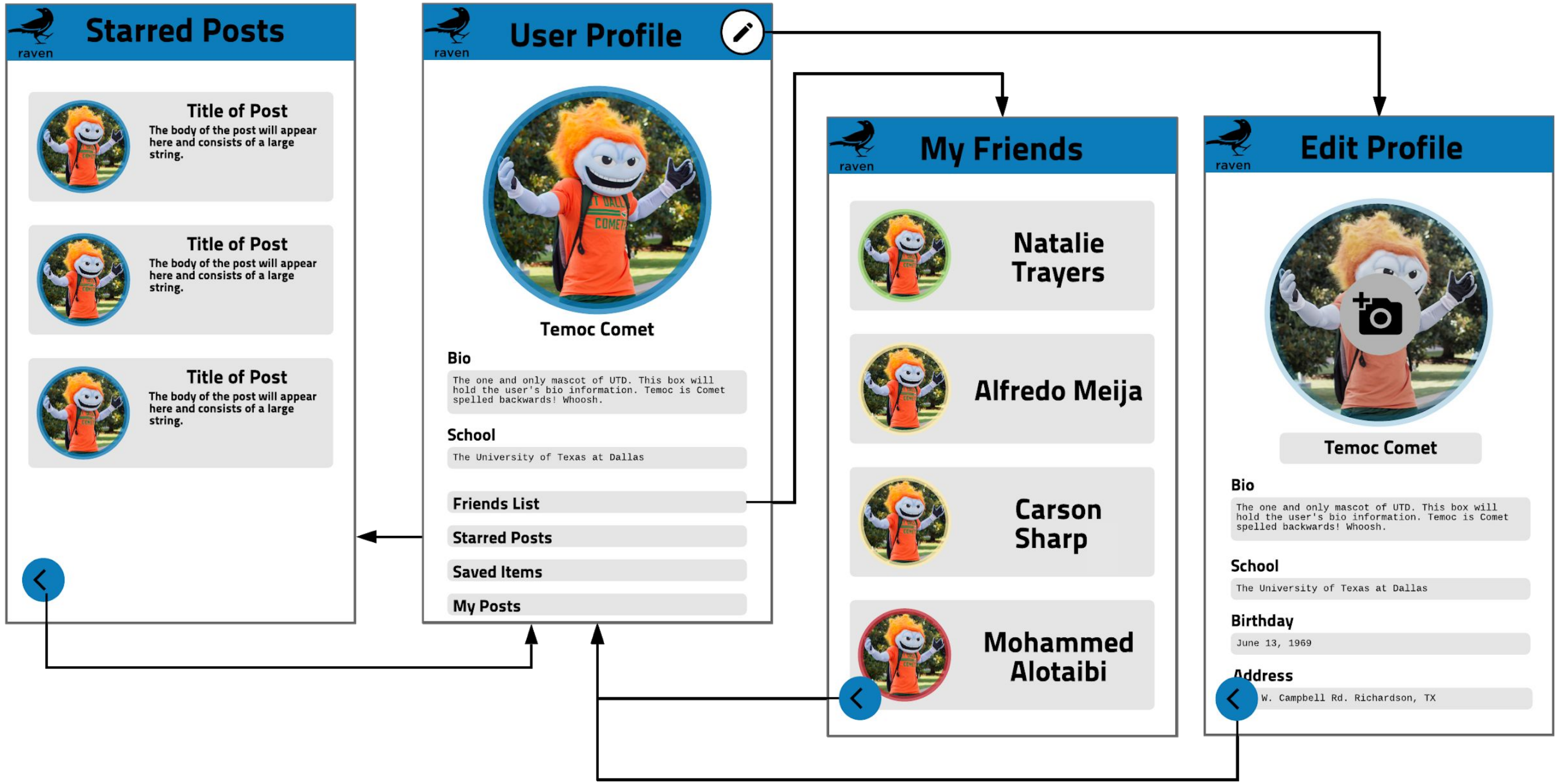
Transition Plan:

- The development team will provide the files to install the system and tools to be able to maintain the system. They will also provide documentation on how to check for errors to keep the system maintained and how to respond to specific alerts. They will have instruction on how to patch a system if there is a bug and how to rebuild and redeploy it.
- Maintenance will first review requirements, designs and understand the code, how it was written and the classes. They will also need to understand the configuration tools and how they work. The development team will show the maintenance team automated testing, what testing tools are used, the test data and how it was deployed.

Login Wireframe


raven
Select University
raven
The Univeristy of Texas at Dallas
The Univeristy of Texas at Arlington
The Univeristy of Texas at El Paso
The Univeristy of Texas at San Antonio
The Univeristy of Texas at Tyler
The University of Texas Rio Grande Valley
raven
50 UTD YEARS
NetID
Enter Your NetID
Password
Enter Your Password
Login
Need Help? Contact the Help Desk
Forgot Password? Reset Your Password
Do Not Bookmark This Page

User Profile WireFrame



Starred Posts

Title of Post
The body of the post will appear here and consists of a large string.

Title of Post
The body of the post will appear here and consists of a large string.

Title of Post
The body of the post will appear here and consists of a large string.

User Profile

Temoc Comet

Bio
The one and only mascot of UTD. This box will hold the user's bio information. Temoc is Comet spelled backwards! Whoosh.

School
The University of Texas at Dallas

Friends List
Starred Posts
Saved Items
My Posts

My Friends

Natalie Trayers
Alfredo Meija
Carson Sharp
Mohammed Alotaibi

Edit Profile

Temoc Comet

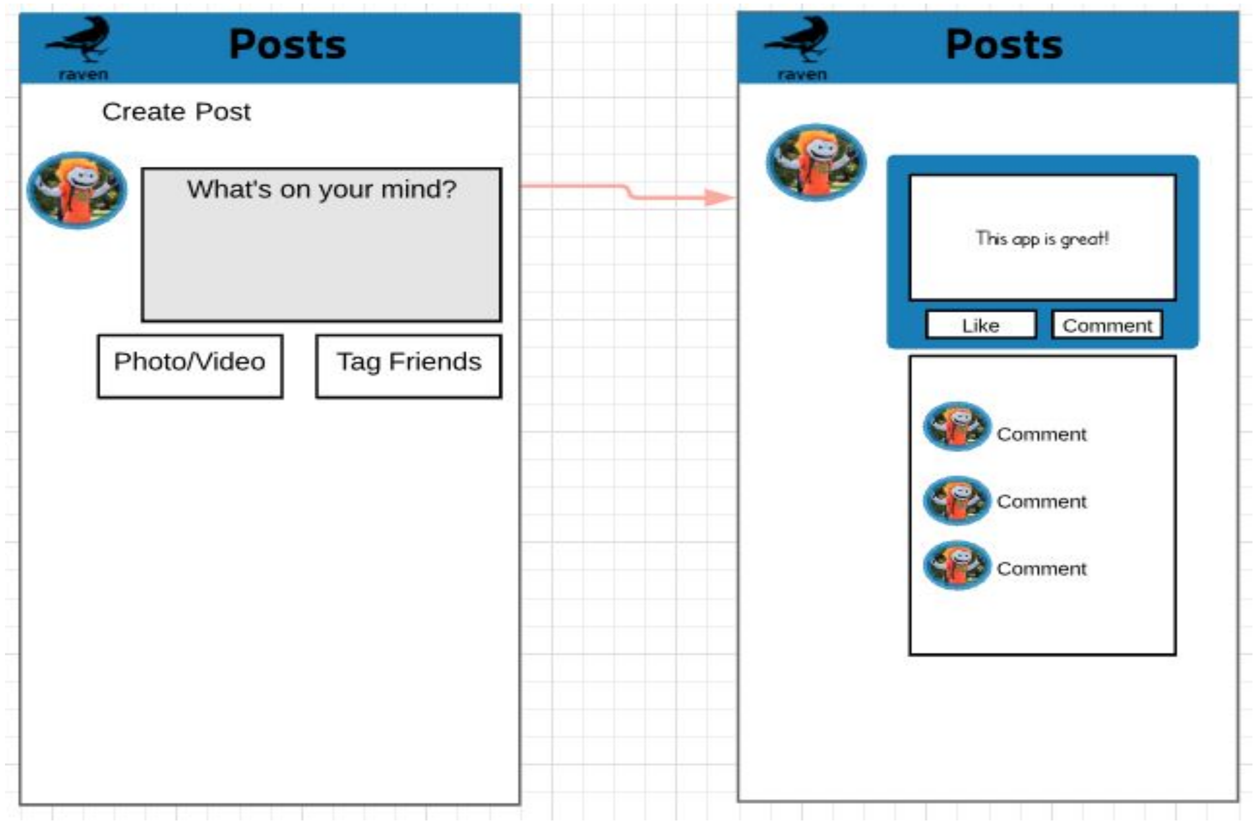
Bio
The one and only mascot of UTD. This box will hold the user's bio information. Temoc is Comet spelled backwards! Whoosh.

School
The University of Texas at Dallas

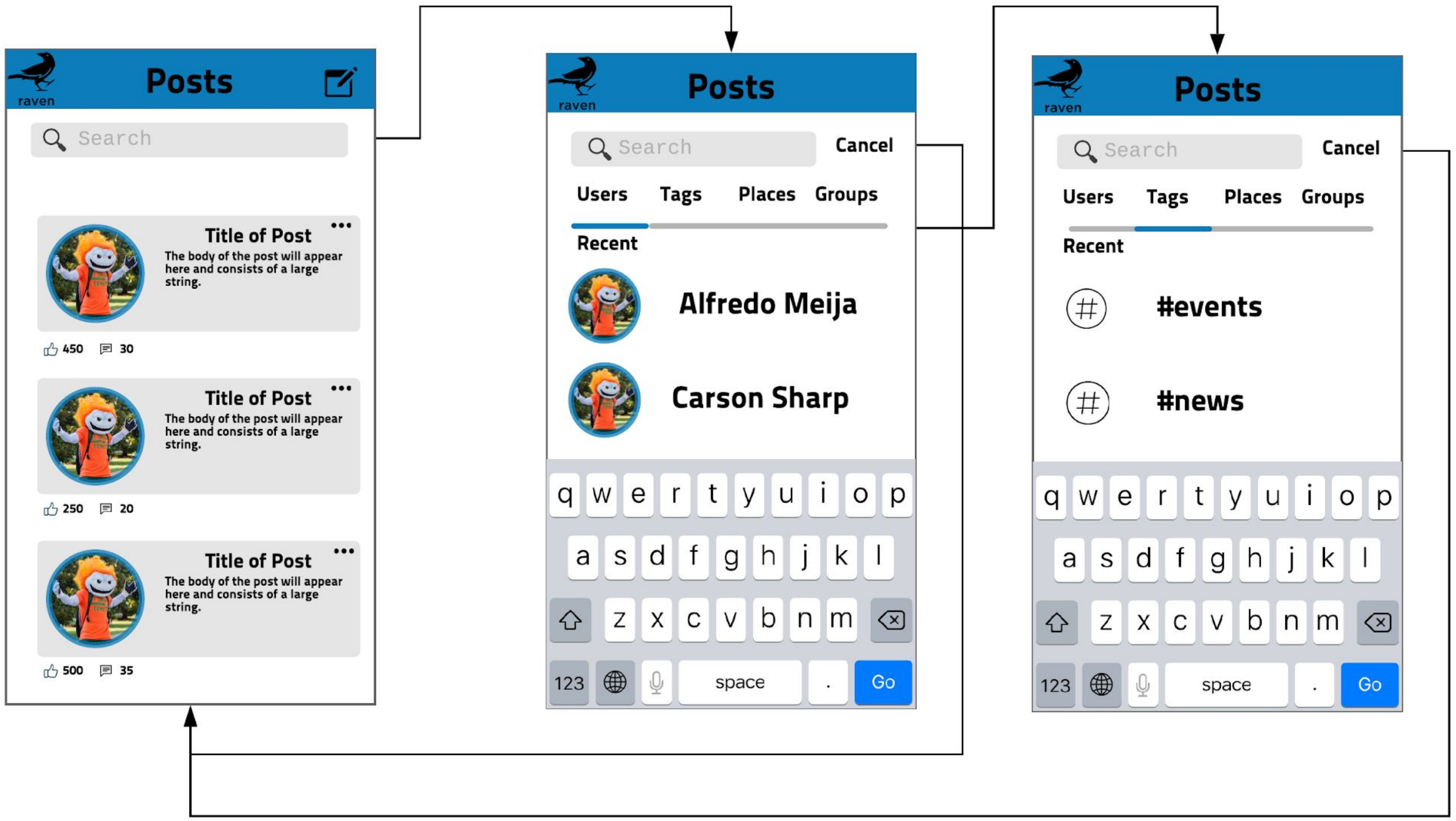
Birthday
June 13, 1969

Address
W. Campbell Rd. Richardson, TX

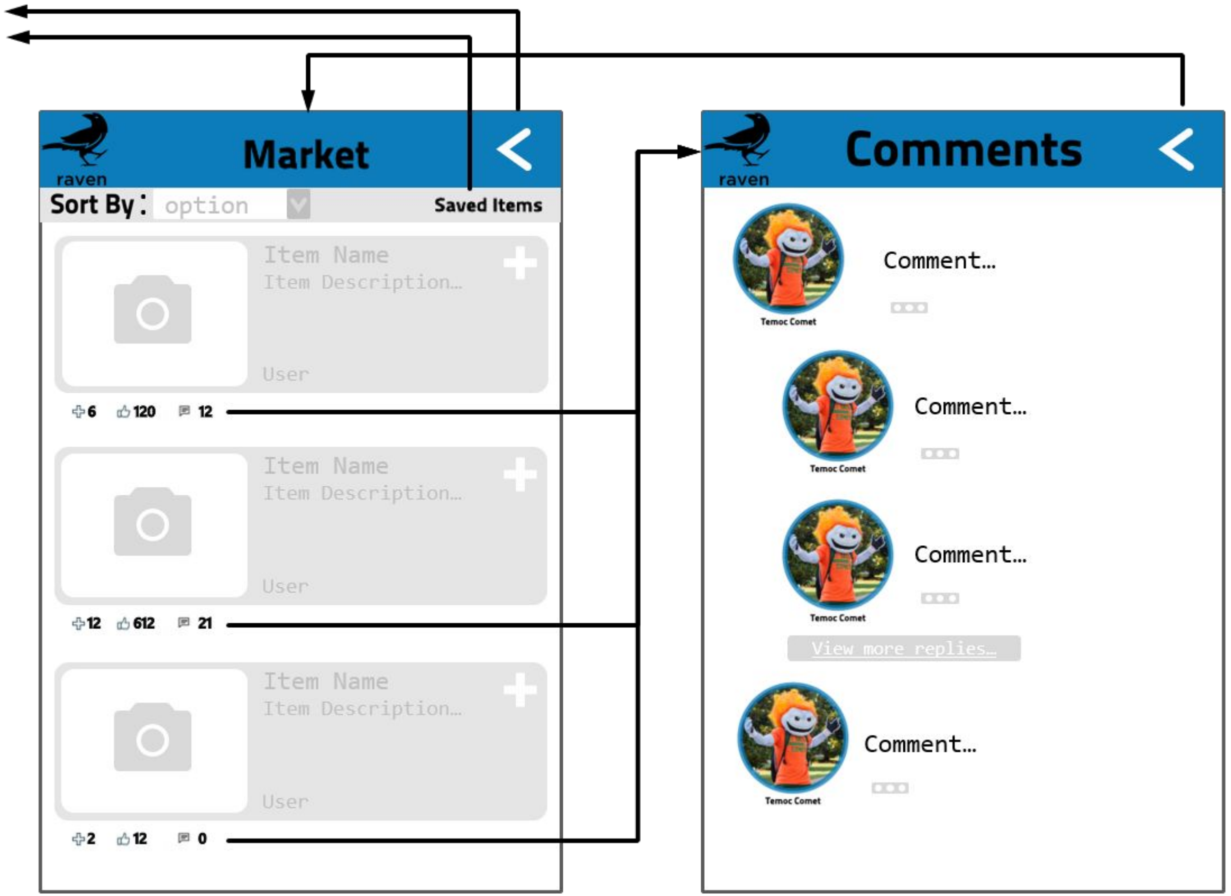
Post Creation Wireframe



Post Feed Wireframe



Items-For-Sale Feed



raven
Market
Sort By : option
Saved Items
Item Name
Item Description...
User
6 120 12
Item Name
Item Description...
User
12 612 21
Item Name
Item Description...
User
2 12 0
raven
Comments
Comment...
Temoc Comet
Comment...
Temoc Comet
Comment...
Temoc Comet
View more replies...
Comment...
Temoc Comet