Legends App

Varsha Roopreddy and Anusha Praturu
California Polytechnic State University
Senior Project, Winter and Spring 2016
June 3, 2016
Advisor: Dr. Franz Kurfess

Abstract

The purpose of this project is to create a location-based story archiving mobile application for the iOS platform. The application is aimed to be a social network with an emphasis on storytelling and location. It will have functionality similar to most popular social networks, such as following, liking, and commenting capabilities, among others. Stories themselves can be accompanied by photos, and can be discovered by users in a number of different ways. The application has a huge range for potential audiences, as it has no geographical or age-based bound for interest.

Table of Contents

Project Overview

Topic and Domain

There is currently no way to preserve the stories tied to a specific location. This application aims to store and tell the entertaining/funny/sad/inspiring/crazy stories of residents and visitors to a particular location. Locations could include things like residence halls on college campuses, monuments, parks, intersections, bars, etc. Anyone can post a story to a location and others will be able to see it, 'like' it, and/or comment on it. Users will be able to follow certain locations, see stories at particular locations that they search, and sort stories by various categories (i.e. Recently Posted, Best This Month, Best This Year, All-Time Best, etc.), among other features.

Intended Users

Our intended users will initially be young adults (18-25 age range) but we hope to expand outward from there. Once the application gains popularity, it will be appealing to almost all age groups. It will most likely be more popular with users in heavily populated areas, such as cities or college campuses, since there are more stories to tell in places like these and more interest in popular locations.

Why is this Interesting?

This application is very unique because it ties a location to a story. Users have the ability to select the story location, unlike similar applications, such as Yik Yak, that group any posts to the nearest "community" using the user's current location. Users also have the ability to write detailed stories with pictures and share them with the world.

Overall this application though similar to things already on the market, has distinguishing features that really will stand out to the users. It also includes features that are present in multiple different applications and puts them all together.

Difficulty

This project is estimated as a moderate difficulty level. Since we both consider ourselves beginners in mobile development, the whole process will take considerably longer than if we were experienced mobile app developers. The bulk of the work to be done is in the back-end. However, we are approaching the front-end first as we anticipate that it will take longer, given our aforementioned limited experience. The back-end system is more familiar to us, and we hope that once the front-end is completed, things will move more quickly. Even though the development of an iOS app is considered a large feat, we hope that splitting the work between two programmers and spending 5-10 hours per week developing in the coming weeks will allow us to finish a working prototype of the application in accordance with our original timeline.

Relevance

In order to capture what we have learned in 4 years studying computer science, we felt that developing an iOS app would be the perfect culmination of all the skills we have accumulated thus far. Our application will utilize data structures, advanced programming, databases, user interface and design principles, artificial intelligence, and many other topics that we have studied at Cal Poly. Mobile platforms account for an enormous sector of the technology industry that we will soon be a part of, and we feel that mobile development is an important skill to learn to be an asset in the tech world. Since we did not directly study Android, iOS, or any other kind of mobile development directly at Cal Poly, we decided that choosing this area for our Senior Project would be the ideal way to learn the fundamentals.

Features, Requirements, and Evaluation Criteria

This section is a user-oriented overview of the different ways the application can be used and interacted with. It is broken up into three subsections: Posting, Viewing, and Interacting. Posting refers to the ability to post content to the app for other viewers to see. This primarily refers to posting a story. Viewing refers to the different ways a user can view content on the app and discover new content. Interacting refers to the different ways a user can interact with existing content on the app.

Posting

- Ability to post a story to current location.
    - Users will need to allow their current location to be used in order to use this feature.
- Ability to post a story to other location
    - Users can input a location that is validated as a real location, so that it can show up on the map.

Viewing

- Map that displays all archived stories

    - Selecting a location displays all stories at that location, sortable by the following categories: Most Popular (All-Time), Most Popular (This Month), Most Popular (This Week), and Most Recent
    - “Most Popular” can refer to views, likes, or comments.
- “Explore” page that displays trending stories
- Ability to search on map, use zoom feature for more specific locations
- Ability to search stories by text using either the location, keywords, or tags
- Ability to search by user (?)

Interacting

- Ability to like/comment stories
- Ability to follow users

This section is a developer-oriented high-level overview of some of the requirements that will need to be met in order to implement the features listed above.

We must save the following information in our backend server/ database:

- A representation of each story, with meta-data including exact location, tags, photos, user, views, likes, and comments.
- A representation of each user, with meta-data including basic user profile information, stories posted, stories saved, and likes.

We will need to create a library that will allow us to perform queries and retrieve the following:

- A set of stories, given a location and the proximity.
- A set of stories, given a keyword or multiple keywords.
- A set of stories, given a user.

We will need a library that will perform the following:

- Return a set of stories to populate the “Explore” page (based on current trends).
- Order a set of stories by popularity or or time.

We will iteratively test the application so that in case there is a need to add a new requirement based on the feedback received and data collected, we will have ample time to do so.

System Design and Architecture

Detailed below are the backend system design for our story archiving mobile app.

User

- List<Story> postedStories

- List<User> followers
- List<User> following
- List<User> blocked
- List<Story> savedStories
- List<Locations> locationsFollowed
- UserInfo info

- User(String fn, String ln, String email, String username, String password, String bio, String phone)
- void addStory(Story s)
- List<Story> getStories()
- Story getStory(int index)
- void deleteStory(int index)
- void addFollower(User add)
- void removeFollower(User remove)
- void addFollowing(User add)
- void removeFollowing(User remove)
- void addBlocked(User add)
- void remove Blocked(User remove)
- void saveStory(Story s)
- void unsaveStory(Story s)
- void followLocation(Location l)
- void unfollowLocation(Location l)
- UserInfo getUserInfo()

Story

- User author
- String storyText
- Location location
- int views
- Date date
- List<User> likes
- List<Comment> comments
- List<Image> photos
- List<String> tags

- User getUser()
- String getStoryText()
- void editStoryText(String newContent)

- Location getLocation()
- Location setLocation(Location l)
- int getViews()
- Date getDate()
- Date setDate()
- List<User> getLikes
- void addLikes(User like)
- void removeLike(User like)
- List<Comment> getComments
- void addComment(Comment c)
- void removeComment(Comment c)
- List<Image> getPhotos
- void addPhoto(Image i)
- void removePhoto(Image i)
- List<String> getTags
- void addTag(String tag)
- void removeTag(String tag)

Comment

- User author
- String content

- Comment(User author, String content)
- User getUser()
- String getContent()
- void edit(String newContent)

Location

- Long longitude
- Long latitude

- Location(Long longitude, Long latitude)

UserInfo

- String firstName
- String lastName
- String email
- String username

- String password
- String bio
- String phoneNumber

- UserInfo(String fn, String ln, String email, String username, String password, String bio, String phoneNumber)
- String getFirstName()
- String getLastName()
- String getEmail()
- String getUsername()
- String getPassword()
- String getBio()
- String getPhoneNumber()
- void setFirstName(String fn)
- void setLastName(String ln)
- void setEmail(String newEmail)
- void setUsername(String newUsername)
- void setPassword(String newPassword)
- void setBio(String newBio)
- void setPhoneNumber(String newPhoneNumber)

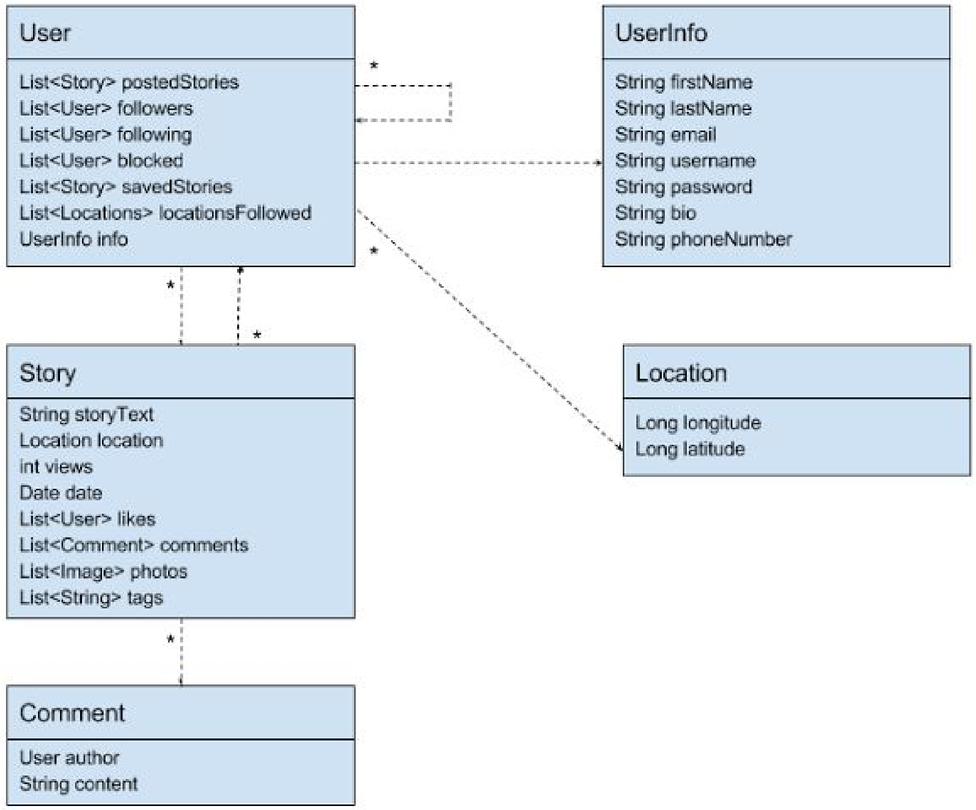
Search Library:

- List<Story> nearUser(Location l, int proximity)
- List<Story> searchKeywords(List<String> tags)
- List<Story> getUserStories(User u)

Explore Library

- List<Story> Trending()
- List<Story> Trending(Location l)
- List<Story> AllTime()
- List<Story> AllTime(Location l)
- List<Story> Recent()
- List<Story> Recent(Location l)
- List<Story> Order(List<Story> toOrder, int orderBy)

Below is a high-level UML diagram based on the backend system detailed above.



Implementation

The implementation of our project was done in Swift using Apple's Xcode as the primary IDE. All the source code for the application is contained within a single Xcode project. To store our data, we are using a third-party data storing service called Firebase. We are able to use their servers in the cloud to store all our user and app data, and we even use their services to create new users and authenticate existing ones when they sign in. Relying on their tried and tested methods for data storage and authentication allowed us to focus on the functionality of the application and the new and exciting features rather than worry about security and other non-innovative aspects of our app.

Our front-end and back-end were implemented more or less concurrently. The front end was started first, but once the database was running and connected to the application on Xcode, the front-end and back-end were worked on simultaneously. Connecting the database also allowed us to work with real data in real time, which improved our development because the results we were seeing were no longer hypothetical.

Our application was implemented from the inside out, so to speak. We started with the "Home" tab, which is essentially the core of our application and the landing page for all users when they open the app. Next, we implemented the "Explore" tab, which is very similar to the "Home" tab, but contains some additional features. We then moved to the "Profile" tab, which displays information about the user and their posted stories. For the current user, the "Profile" tab is where they can make changes to their profile or account and where they can log out. Then we created the page that allowed users to post a story. This was a major breakthrough because we consider it the very first minimum viable iteration of our application that contained all the basic functionality.

After this we moved on to non-core yet still important features. We implemented the "Log In" and "Create Account" pages that are displayed when a user is not currently signed in. This is where a user can log into their account, and if they don't have one, they can sign up. The back-end of these pages utilize Firebase's authentication and other user services. These pages, while essential, were not part of the core functionality of our app.

Once this bare bones setup was complete, we moved on to finer grained aspects of implementation. For example, one of the next things we implemented was the "Like" feature on individual stories and the "Follow" feature for individual users. We also implemented the Search feature on the "Explore" tab. We allowed users to search stories by Author, Location, and Story Title. We also made a major modification to the "Post Story" feature. By connecting to Google's Map API, we use validated locations that are recognized by Google as locations for the stories posted to our app. Any location that appears on Google Maps is now an option for our users to choose for the location of their story. The purpose of this is twofold. First, the geographical locations of stories (as they appear on the "Map" tab) are guaranteed to be accurate, so long as Google maintains coordinates. They also come with addresses that can appear on the "Map" tab when a pin is selected. The second purpose is the legitimize locations that the users select. Even if users are tempted to make up their own locations or select something that does not agree with the community standards of our application, they will not have the option to do so.

User Feedback

Feedback on User Interaction

The feedback we received regarding user interaction was primarily positive. The application was "intuitive" and "easy to use" according to most users. The sign-up and login processes were very simple. The only aspect that users seemed to have trouble with was entering information when they were creating/posting a story. It was not intuitive to have the search bar for the location in the navigation bar when all the other elements were in the main part of the screen. One of the next iterations of our application will move the location search bar down to where the rest of the story information is located. This makes more sense than to have it in the navigation bar. Other than that minor detail, users were able to interact with the application fairly easily with no major roadblocks.

Feedback on Functionality

The feedback on functionality was also primarily positive. Some users were a little confused on how the app worked and was meant to be used, but once they saw examples of stories it became more clear to them. I don't think that there is really much of a solution to this issue. If this app takes off, people will be signing up and downloading it because they already know its purpose. The other feedback we got on functionality was that there were additional features that people wanted to see implemented. We took note of these features and intend to implement most of them in follow-up iterations of the app.

Feedback on Implementation Aspects

In general, people gave positive feedback about the way we chose to implement our app. We set it up as a tabbed application that closely resembles the setup of many other social media platforms that people are already used to using, such as Twitter or Instagram. As previously mentioned, there were other features that they would have liked to see implemented, such as hash-tagging, reporting stories, searching story text, etc. and these are all features that we intend on implementing in the future.

Validation and Evaluation

Features

1. Posting
2. Viewing
3. Interacting

Evaluation Criteria

Criteria	**Result**
Can the users easily post stories?	Yes
When posting a story, can the user only post a	Yes

valid location?	
Can the users easily view the stories of the users they follow?	Yes
Can the users search for other users, stories, and locations?	Yes
Can the users follow other users?	Yes
Can the users like other stories?	Yes

Overall Assessment

Overall we have met our evaluation criteria very well. We were able to complete all the features that were essential to the core functionality of the application. We have a fully-functional iOS application that customers can use. There is definitely more that we could do to improve the application and enhance the user’s experience, but we are confident with the end results.

Conclusion

The past two quarters of senior project we accomplished and learned a lot regarding iOS application development. We were able to write an iOS application from end-to-end by setting up the database on Firebase to designing the user interface of Legends. Throughout this process we learned that it is not the easiest thing to write an iOS app, and that it takes a lot of research and searching to figure out how to do things. For example, most of the times we looked up tutorials they were outdated because Swift is a fairly new language that is constantly being improved. What could have worked a year ago will not work with the new Swift version.

Overall we were able to pick up Swift pretty quickly and had a really good experience developing Legends. The scope of the projects allowed us to learn a lot about the different aspects of iOS application development. With our knowledge, now we can create iOS applications more confidently and maybe even try another type of mobile development.