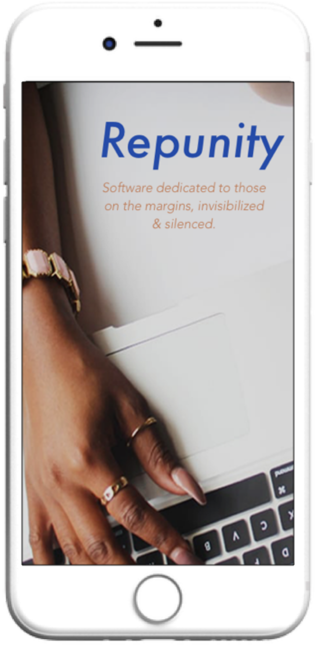
CPSC490

Repunity:

Community of underrepresented role models.

Presented to the Computer Science Department At Yale University Spring 2018


Repunity
Software dedicated to those on the margins, invisibilized & silenced.

By: Erika Alexandra Hairston
Advised by: Professor Ruzica Piskac

To my Hidden Figures -- Software dedicated to those on the margins, invisibilized and silenced. Bringing minority voices to the forefront, paving a more accessible path for future generations.

Table of Contents

Table of Contents 2

Abstract 3

Introduction 3

Overview 4

User Stories 4

College Goal 4

General Inspiration 5

Background - Why this matters 5

Goals of Project 5

Learning Trajectory 6

Resources 7

The App 7

Implementation | User Interface 8

SignIn/SignUp Flow 8

Role Model Suggestions Scoring Algorithm- Home 9

Discovery , 9

Messaging Feature 10

Async 11

Backend - Data Models 12

App Structure 12

Discussion 14

Future Work 15

Testing 15

Considering Privacy 15

Path Suggestions 15

Conclusion 15

Acknowledgments 16

Sources 16

Abstract

For my culmination of study in the Computer Science Department at Yale, I chose to tackle an issue that has colored my entire experience as an undergraduate in a highly homogenous major. I came into college with no clue of what code even was, let alone how to create and innovate with it. However, the one thing I knew for sure was that I had never seen anyone that looked like me in the field. This did not change at Yale, nor in Silicon Valley today. This app, what I call Repunity, aims to tackle that severe issue of underrepresentation. In this paper, I discuss technical aspects of my final project along with my learning trajectory in teaching myself iOS.

Repunity serves as a database of underrepresented individuals that can be filtered by identifiers and suggest Role Models to you based on your user Profile. I aggregate user information to suggest helpful resources that may be relevant to your identity and professional path. I also implemented messaging functionality within the app for users to reach out to Role Models. My goals of this project were: 1) use technical skills that I've gained to tackle a critical issue in my community, 2) submerge myself in the app development cycle from ideation to deployment, learning difficulties around choices and mobile specific technologies, 3) best prepare myself for a role in Product Management at industry level where I will be focused on design, analytics, and user-first decisions.

Repunity has been one of the most valuable projects to work because of how much it solidified my confidence as a Computer Scientist that strives for optimal time complexity, data structures, and ease of User Experience -- all topics I learned from internships, electives, and core classes in my time at Yale.

Introduction

What happens when you never see people who look like you in roles you aspire to? You're implicitly told that role is not for you. (Mis)Representation powerfully can reinforce or shift mainstream norms and stereotypes. For my senior Computer Science Project, I developed an iOS app that aims to tackle under representation across fields and career phases, making diverse professionals and their paths visible and accessible. Additionally, I challenged myself with the task of learning a completely new coding language, environment, and backend in order to better prepare myself for industry level working with mobile development. *While the technical aspects of my project were highly prioritized, the imperative value in this work would be missed if I didn't discuss the need for a platform such as Repunity and my process of developing the skills to do so.*

Overview

User Stories

College Goal

Jeffrey, a 15 year old boy from Chicago, is completing his freshman year of high school. He has strong grades and intense ambition. He loves science and is an incredibly strong writer. Both of his parents immigrated to the States right after he was born and neither had any experience with higher education. His high-school barely had funding for textbooks, let alone sent kids off to college. However, Jeffrey had bigger plans -- he knew he wanted to effect the law and still have a hand in the sciences. To do that, he found out he needed a college degree. Everyone around him told him college was only for the rich and that no one from his school got in. He believed them and stopped trying.

Ideally, Jeffrey would be able to search a vast database of diverse individuals to find someone who came from an immigrant, low-income home and still went to college. He should be able to find role models who have overcome similar struggles and see scholarships they applied to or extracurriculars he could start. He should know he's not alone and see positive images of Black men as patent judges and attorneys. Jeffrey should have someone or something to aspire to that is accessible from his world.

General Inspiration

Tory is a college sophomore with no clue what they want to do in the world. They're interested in LGBTQ rights, social justice, and sports. Tory is used to be being told that their passions don't align and that they can't pursue careers in the sports world post graduation.

Ideally, Tory should have access to a pool of real people who identify as LGBTQ and work in the sport industry. One day, Tory could find someone who has a career that extremely excites them. Despite the norm, Tory should be able to be affirmed that just because their interests are different, they are not impossible to pursue. They should be inspired by other people in the world who also have uncommon passions and want to inspire others with their own story. Tory now has a career to aspire to and doesn't have to abandon their passion nor identity.

Background - Why this matters

Currently, few platforms aim to dispel the negative effects of misrepresentation and severe underrepresentation of minorities in the professional world. Seeing white men as the coder in media for decades tells us a story; Seeing Black and Brown people as the help tells another story. In order to change the mainstream narrative, the app saturates society with the stories of underrepresented individuals pursuing their passions and careers. Everyone's story could inspire someone somewhere.

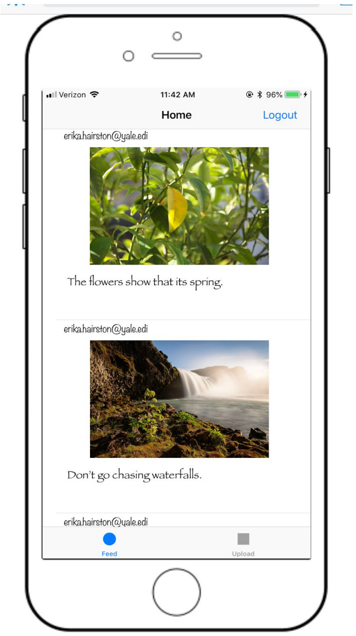
Repunity aims to solve this problem by growing a database of diverse individuals, "Role Models" along with their interests, career paths, and identifiers. Using given background information from each "Role Model," the app provides users with a curated list of people that have achieved a similar goal they are looking for and can be filtered by their mutual identifiers or interests. They then are able to message their Role Model within the app without having to track down emails, phone numbers, or addresses.

Goals of Project

Throughout this project, my goal was to reach a place with iOS where I felt confident bringing ideas to fruition using Swift along with other backend features and APIs. For the first half of the project, I focused on building smaller test iOS apps to better acquaint myself with xCode, Swift, FireBase, CocoaPods, etc. Prior to beginning Repunity, I developed over ten smaller apps iterating on design, connecting to databases, working with images, cloud storage, and optimizing run times. I then hoped to use those skills to create at the base level, an app that utilizes Firebase's backend, user Authentication, and developing my own algorithms to suggest Role Models to users. Therefore my goal was twofold -- becoming fluent in the app development cycle, whilst building a technically sound application.

Learning Trajectory

While working on Repunity, I quickly learned that there was no clear path towards understanding every important aspect of iOS development. I began the semester following along with Stanford's Course, "Engineering CS193p: Developing iOS 10 Apps with Swift,"[1] that is offered on Youtube. However, for the sake of having time to fully develop my project, I followed along with the first seven classes in order to develop a solid foundation in not only coding for iOS, but also learning optimal methods and best practices. In addition to the Stanford Online course, I took an online Udemy iOS 11 Course to supplement my applicable learning with up-to-date concepts to create Repunity.[2] Throughout this process, I developed several sample apps that helped prepare me for my larger scale project. For example, I created an Instagram Clone app that allows login, image sharing, and posting.



erika.hairston@yale.edi

The flowers show that its spring.

erika.hairston@yale.edi

Don't go chasing waterfalls.

erika.hairston@yale.edi

Figure 1: Sample Instagram Clone App

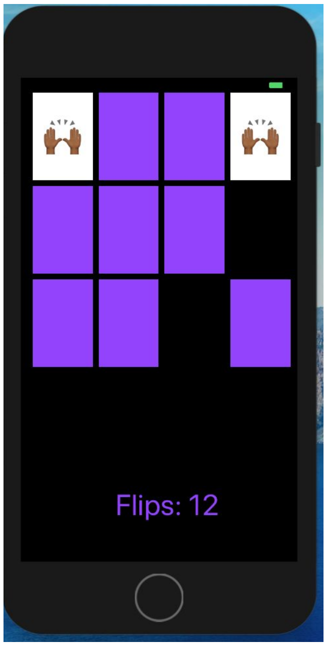


Figure 2: Sample Concentration App

[1] "Developing IOS 10 Apps with Swift - Free Course by Stanford on ITunes U." Apple Music, Apple Inc, itunes.apple.com/us/course/developing-ios-10-apps-with-swift/id1198467120.

[2] Samancioglu, Atil and Joe Parys, directors. The Complete IOS 11 Developer Course - Beginner To Advanced! . Udemy, Apr. 2018, www.udemy.com/the-complete-ios-10-developer-course-beginner-to-advanced/learn/v4/content.

Resources

Though the more formal courses played a formative role in progressing my learning, short Youtube videos, StackOverFlow, and Apple's Developer pages proved to be the most helpful and effective, especially once I began coding Repunity. From a different light, I also took the time to learn more about best design and UX practices. For instance, I read up on several blogs and websites about the best way to format a mobile signup form,[3] while also finding UIKit exercises and tools. These skills proved imperative to developing a fully immersive app experience that puts the user first.

The App

Repunity is an iOS app built using Swift, CoCoaPods, xCode Core Data and Google Firebase. The user interface is heavily inspired by Spotify, Hipmunk, and Apple's App Store, featuring several scrollviews, bright color, and navigation/tab bars.

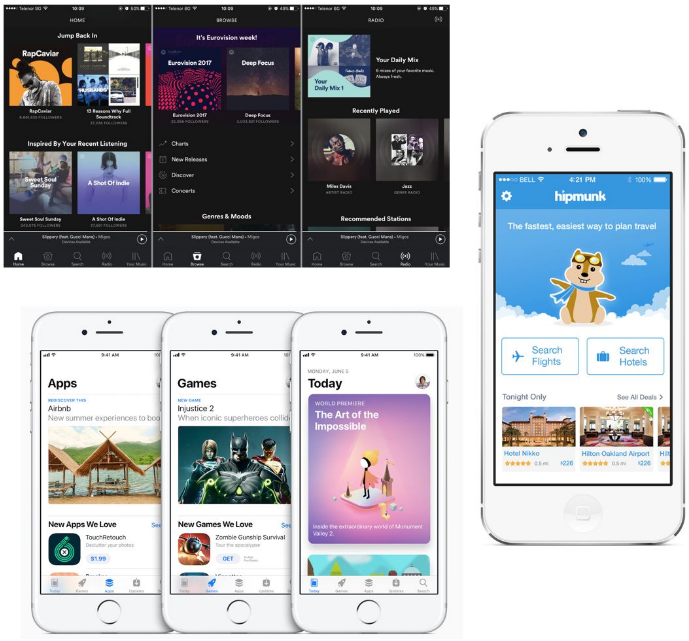


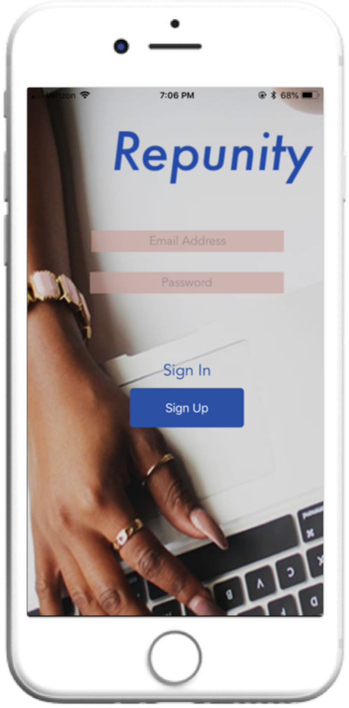
Figure 3: Design Inspirations for Repunity [Pictured: Spotify, top; Apple bottom; Hipmunk right]

[3] Ibragimova, Eleonora. "10 Best Practices for Designing User-Friendly Forms."*Theuxblog.com*, Medium, 3 Jan. 2017, medium.theuxblog.com/10-best-practices-for-designing-user-friendly-forms-fa0ba7c3e01f.

Implementation | User Interface

The User Interface of Repunity contains four main flows: Login/SignUp, Role Model Smart Suggestions, Discovery Page and Messaging Platform. A user must be logged in to use the app and access Role Model information.

SignIn/SignUp Flow[4]



The signup page asks for simple userName and Password with confirmation, in order to create an initial Role Model account. I refer to all signed-up users as ‘Role Models’. The signup flow then guides the user through a series of prompts about their personal identity, employment and education experiences. They are also able to upload a profile picture that will be associated with their account.

I upload all of this data to three Databases: RoleModels, Auth users, and Storage. Each is for their profile information, login authentication, and profile photos respectively. I also implemented Swift’s Core Data model to remember when a user has already logged in; therefore, the next time the app is opened on their own phone, they will already be brought inside the app to the Home Screen, skipping “Sign In.”

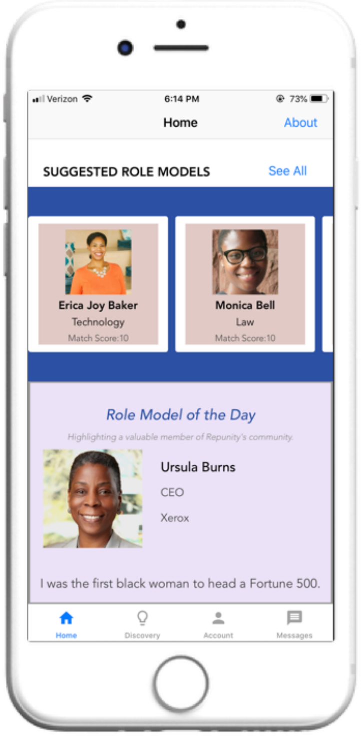
Role Model Suggestions Scoring Algorithm- Home[5]

The Home Page of the app features the most valuable aspect of Repunity -- suggested Role Models based on your given user profile information. As the goal of Repunity was to surface underrepresented groups across fields, I decided to center a list of “Suggested Role Models” at the front of the app for users to engage with. I used Swift’s CollectionViewController framework across the top half of the screen to feature previews of ten Role Models that share industries, majors, race, gender etc. with the logged in user. The

[4] *Figure 4: Sign In Screen*

[5] *Figure 5: Home Screen of Suggestions & Spotlight*

cells are presented in order of ranking, an algorithm I used to determine Role Models they might like. Additionally, in order to view more information about each Role Model, I included a button to a Results page that I use to list several Role Models and their attributes that are clickable to their specific profile.



Verizon 6:14 PM 73%

Home

About

SUGGESTED ROLE MODELS

See All

Erica Joy Baker
Technology
Match Score:10

Monica Bell
Law
Match Score:10

Role Model of the Day

Highlighting a valuable member of Repunity's community.

Ursula Burns
CEO
Xerox

I was the first black woman to head a Fortune 500.

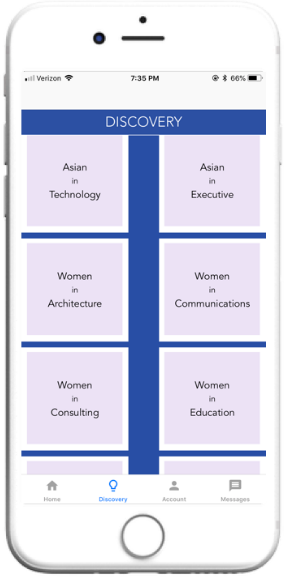
In addition to suggested Role Models, the home page also features a spotlight user in order to surface individuals we might not come across ourselves. While the goal of the app focuses on identity representation, it is also powerful to come in contact with people, ideas, and experiences far different from our own. The "Role Model of the Day" spotlight aims to capture that value by presenting a randomized user, that in the future would change every different day the user logs in.

Finally, the home screen also contains a tab button "About" that uses the Modal ViewController presentation to highlight Repunity's intentions and assure users what it'll be doing with their information .

Discovery [6],[7]

The second Tab of the app is the Discovery Feature. Like the CollectionView that displays Top Role Models, in this page I dynamically present a ScrollView of several categories that work as buttons. Each category is generated from user identifiers and industries in order to capture every user's experience. I dynamically created the buttons to then segue to the Results model with Role Models that fit within that specified category.

For example, if on sign-up you entered that you're a "Non-binary women in technology," the category "Non-binary in Technology" would dynamically



Verizon 7:35 PM 66%

DISCOVERY

Asian
in
Technology

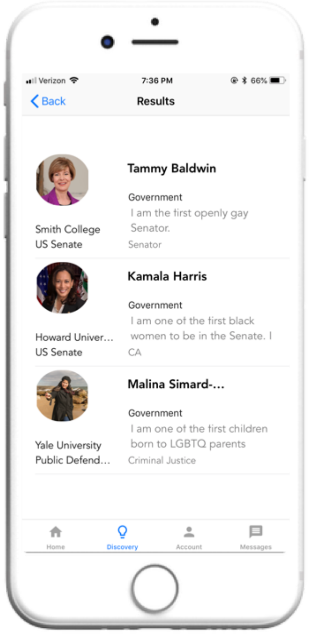
Asian
in
Executive

Women
in
Architecture

Women
in
Communications

Women
in
Consulting

Women
in
Education



Verizon 7:36 PM 66%

Back

Results

Tammy Baldwin
Government
I am the first openly gay Senator.
Senator
Smith College
US Senate

Kamala Harris
Government
I am one of the first black women to be in the Senate. I
CA
Howard Univer...
US Senate

Malina Simard-...
Government
I am one of the first children born to LGBTQ parents
Criminal Justice
Yale University
Public Defend...

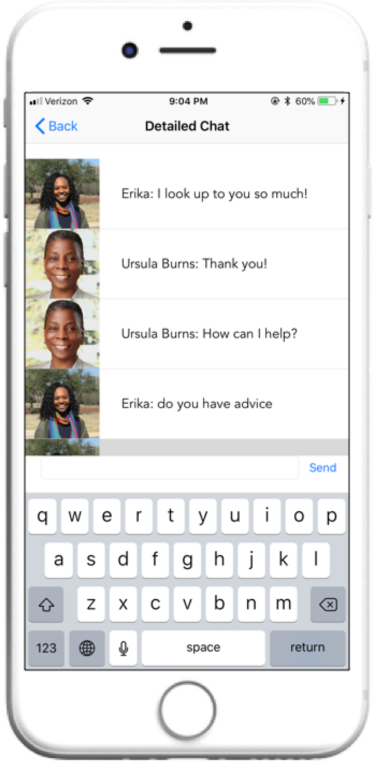
[6] *Figure 6: Discovery Screen*

[7] *Figure 7: Results of clicking "Women in Government" Screen*

appear, if you are the first, or you'd join the list of results with other users that are non-binary in tech. Each time you click on a category, you are brought to a results page that accepts a query of Role Models and displays them in a TableViewController Framework that makes each result clickable. The quickable RoleModel cell then brings you that selected Role Model's profile page. Like the Results ViewController, the profile view controller also takes in a profile and displays that specific user's information along with a button that opens a message to that Role Model.

Messaging Feature

The in-app messaging was by far the most complex technical aspect of the project to create. I began by following along with Firebase's tutorial; however, after completing it I realized that their sample app created more of a public "messaging board", rather than a unique 1-on-1 messaging experience between users. My development goal was more complex than the sample app. The messages flow has two main screens: a TableView of all RoleModels the current user has a shared message with[8], and a TableView of all messages between two users.[9]



< Back

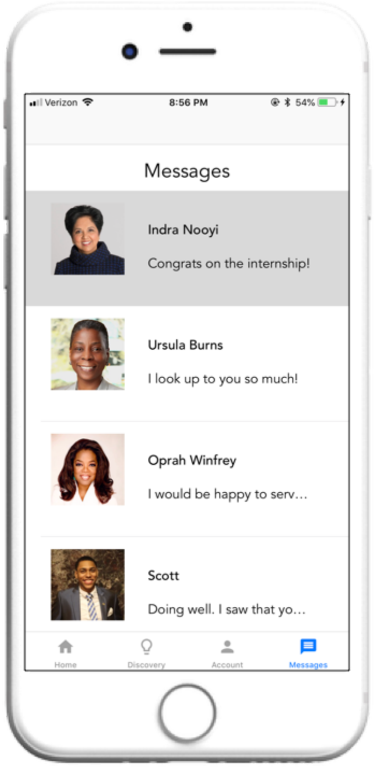
Detailed Chat

Erika: I look up to you so much!

Ursula Burns: Thank you!

Ursula Burns: How can I help?

Erika: do you have advice



Messages

Indra Nooyi
Congrats on the internship!

Ursula Burns
I look up to you so much!

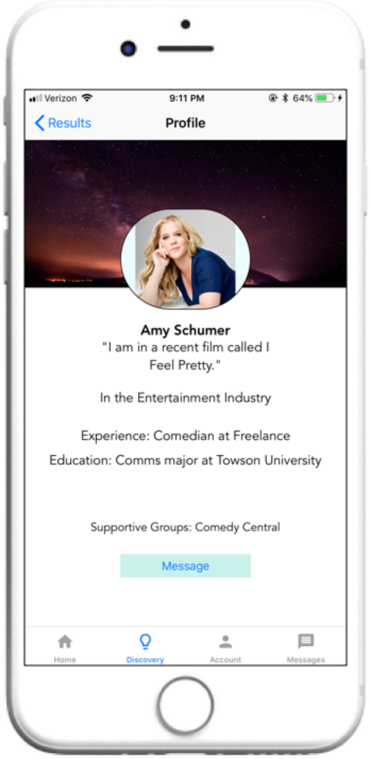
Oprah Winfrey
I would be happy to serv...

Scott
Doing well. I saw that yo...

The detailed chat screen came almost directly from the tutorial; however, instead of allowing every user to chat in one place, it only displays to the relevant Receiver and Sender. Additionally, the experience is in Real-Time. Therefore, if you send a user a message while they are in the chat, it will display immediately for them -- a delightful user experience that avoids action to refresh.

[8] *Figure 8: Detailed Chat Screen (messages between two users)*
[9] *Figure 9: All Messages Screen*

Despite the complexity issue, the tutorial provided a solid framework for which I iterated on several times in order to transpose to a true messaging experience. The messaging feature uses a Firebase Collection that stores individual "messages" by an auto-generated message-id. In order to make messaging possible on an individual level, I had to add attributes to the messages model, such as "SentToID" and "SentByID" in order to display relevant messages to appropriate users. I'll discuss the specifics of the messages data model in the backend chapter of this paper. [10]



Amy Schumer
"I am in a recent film called I Feel Pretty."

In the Entertainment Industry

Experience: Comedian at Freelance
Education: Comms major at Towson University

Supportive Groups: Comedy Central

Async

Along the lines of storing the messaging information, retrieval was the most complex part because I kept running into issues around Asynchronous calls to the database. Since the 'messages' data was linked to the account data and I needed to know whose profile sent or received which messages, I ran into several issues where the "Sender" and "Receiver" attribute would return null because I stored the message before my query to the 'user' database finished. I tried several solutions that led to assuming the sender based on log-in, completing the task on a new thread, but finally ended with optimizing my data models to include the ids of users and receivers. That way, from the start, a message assigns the correct sender and retriever profiles.

Backend - Data Models

Last semester, in CPSC 437 Database Systems, was my first time learning about data modeling and backend operations used to power technology. From that class, I built a website using SQL operations in order to apply my new knowledge of creating optimal entity-relational models, normal form, and integrity constraints. In this project, I wanted to continue practicing those same good relational database design skills; however, I quickly realized that my initial choice to use FireBase because of its built-in security, speed, and authentication meant sacrificing a traditional relational data model since Firebase is a

[10] *Figure 10: Profile Screen that brings you directly to Detailed Message to that Role Model*

NoSQL document-oriented database. Therefore, within my data there were no tables nor rows, but instead organized into collections.

Though I had just spent a semester learning how to query and store data using entity-attribute data models, for Repunity there was a steep learning curve on how to store and retrieve data with the Collections JSON framework. Once I was able to get a hang of data retrieval, I realized that my skills about functional dependencies, concurrency control, and general database design still were necessary and came in handy for this project. Moreover, in addition to Firebase, I used iOS's Core Data for remembering login information, which is an object graph and persistence framework used to manage the model layer objects in my application.[11] It allowed me to directly apply my skills from CPSC437, while also serving as a small way to create a much nicer user experience -- not having to re-login every time you open your own app.

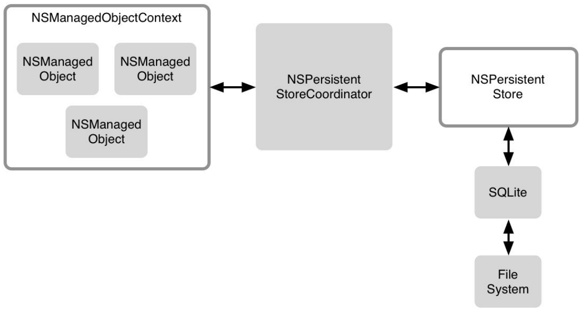
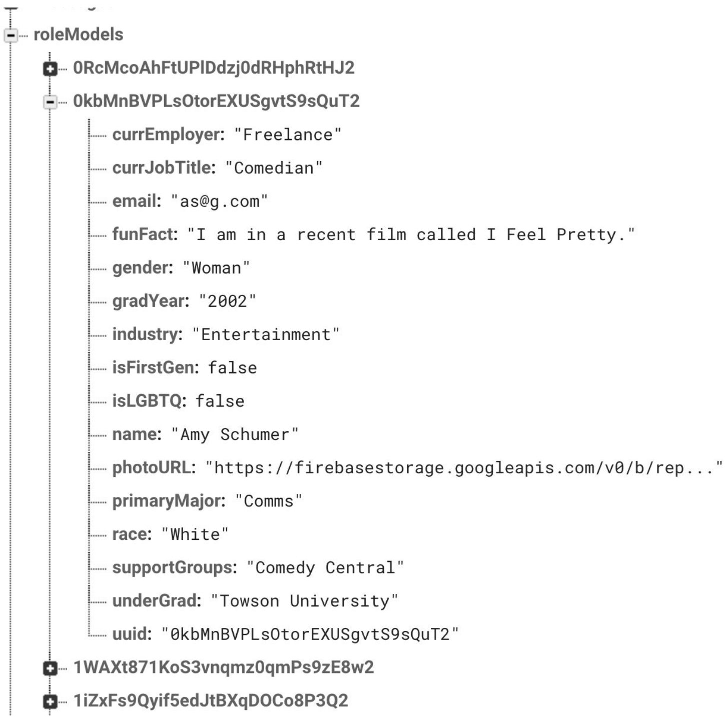


Figure 11: Core Data model

App Structure

Repunity has four main back-end attributes: user logins and profiles (Role Models), messages, and photo storage. Upon creating an account (submitting an email and password confirmation), an authentication profile is created for that user, giving them a unique id and hashing their password. Once a user continues through the signup flow and supply their RoleModel information, I use their identifies and experience tied to their unique user id, to create a Role Model class and database.

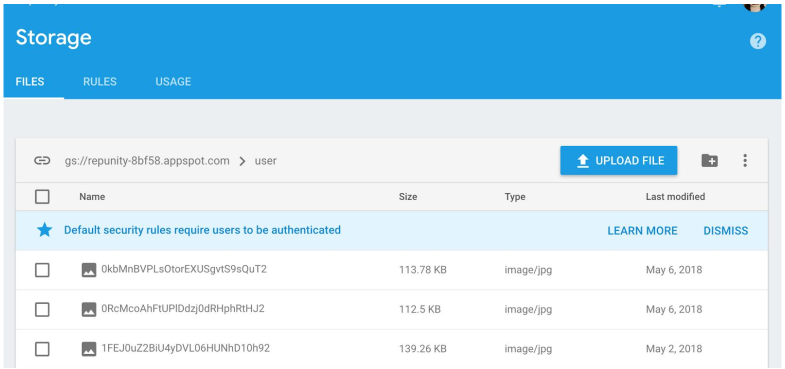
[11] Apple Inc. "Core Data." *Apple Developer* , Apple, developer.apple.com/documentation/coredata.



```
roleModels
    0RcMcoAhFtUPlDdzj0dRHphRtHJ2
    0kbMnBVPLsOtorEXUSgvtS9sQuT2
        currEmployer: "Freelance"
        currJobTitle: "Comedian"
        email: "as@g.com"
        funFact: "I am in a recent film called I Feel Pretty."
        gender: "Woman"
        gradYear: "2002"
        industry: "Entertainment"
        isFirstGen: false
        isLGBTQ: false
        name: "Amy Schumer"
        photoURL: "https://firebasestorage.googleapis.com/v0/b/rep..."
        primaryMajor: "Comms"
        race: "White"
        supportGroups: "Comedy Central"
        underGrad: "Towson University"
        uuid: "0kbMnBVPLsOtorEXUSgvtS9sQuT2"
    1WAXt871KoS3vnqmz0qmPs9zE8w2
    1iZxFs9Qyif5edJtBXqDOCo8P3Q2
```

Figure 12: Role Model entry with Sample user

Additionally, a user is able to upload a profile picture, which gets stored in the Storage database that generates a downloadable URL for that specific photo. I associate that URL with the RoleModel as well.

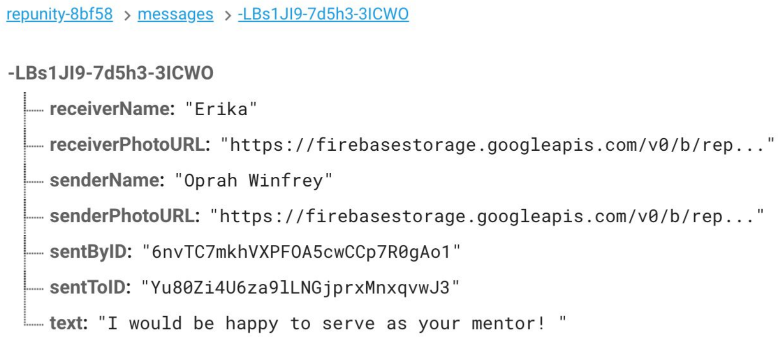


gs://repunity-8bf58.appspot.com > user

Default security rules require users to be authenticated

	Name	Size	Type	Last modified
□	0kbMnBVPLsOtorEXUSgvtS9sQuT2	113.78 KB	image/jpg	May 6, 2018
□	0RcMcoAhFtUPlDdzj0dRHphRtHJ2	112.5 KB	image/jpg	May 6, 2018
□	1FEJ0uZ2BiU4yDVL06HUNhD10h92	139.26 KB	image/jpg	May 2, 2018

Finally, the messages flow requires a new database that stores all individual messages in order of time. When I use the push method to add data to Firebase Realtime Database for messages, an automatic ID is added as the “title” for that specific message. “These auto generated IDs are sequential, which ensures that new messages will be added in the correct order."[12]



```
repunity-8bf58 > messages > -LBs1JI9-7d5h3-3ICWO

-LBs1JI9-7d5h3-3ICWO
├── receiverName: "Erika"
├── receiverPhotoURL: "https://firebasestorage.googleapis.com/v0/b/rep..."
├── senderName: "Oprah Winfrey"
├── senderPhotoURL: "https://firebasestorage.googleapis.com/v0/b/rep..."
├── sentByID: "6nvTC7mkhVXPFOA5cwCCp7R0gAo1"
├── sentToID: "Yu80Zi4U6za9lLNGjprxMnxqvwJ3"
└── text: "I would be happy to serve as your mentor! "
```

Each message contains the photoURL, ID, and name of both the Sender and the Receiver in order to avoid the issues I had earlier with asynchronous calls returning null when trying to assign RoleModels as the sender and receiver.

Discussion

Throughout this process I have developed an app that I am truly proud of and will continue working on after graduation. I achieved my goals of feeling confident that given my Computer Science foundation, I am capable of not only learning a completely new language, backend, and environment, but also applying those skills to produce relevant and complex technical content. Additionally, I added messaging functionality that was not a part of my original deliverables because I underestimated my own technical ability. This feature posed the most unique and teachable aspect of the project.

Future Work

Lastly, going forward, there are many features to consider for the optimal growth of Repunity.

Testing

For one, testing was not an aspect of Yale Computer Science that I had been introduced to. However, my advisor pushed me towards the end to read about it and familiarize myself

[12] https://codelabs.developers.google.com/codelabs/firebase-ios-swift/#6

with the process, as it is pertinent to the app development cycle. Though I used the time I had hope to devote to Testing, to messages, in the future I would implement Testing throughout the entire development process. I have also linked resources to information about Test Driven Development, that I found helpful to understand.

Considering Privacy

In addition to testing, one of the key aspects of Repunity that I must be sensitive about going forward are the privacy concerns regarding self-identifying race, gender, secuality etc. I recognize that some people may not want this information public; therefore, these fields are not required. However, I built Repunity on the intention of having users who are proud to represent marginalized communities and opt-in to sharing their experiences and identity with the community. In the future however, I would develop more safe-guards optional fields around identity to ensure user's confidence.

Path Suggestions

Lastly, I begun the process of creating an additional functionality that suggests personalized helpful resources to each user based on their Top Role Models' experiences; however, I wasn't able to finish it for the final submitting. In the future, I would continue building this feature out, as I see it as one of the most actionably valuable aspects of Repunity.

Conclusion

Building Repunity was one of the most challenging and rewarding experiences during my time at Yale. It truly felt like a culmination of all that I have learned throughout the last four years as a Computer Science and African-American Studies Double Major. Though not a joint project, I truly believe the work of my African-American Studies thesis, entitled "Racial Realism and the Failed Promise of Diversity in Tech: Mapping Counter/Stories of Diversity to Produce a Roadmap for Inclusion," inspired the mission and content of Repunity.

Repunity is an attempt to put the theory of racial realism into practice in the technology sector. It offers counterstories from minorities which contradict majoritarian narratives around how one becomes a technologist and who can be a technologist. In this way, it works to solve issues of underrepresentation across various fields and career phases. Repunity is an outgrowth of my positionality, investments, and concerns as a Black female engineer. When Black technologists are excluded from the industry, ideas like Repunity are lost.

Acknowledgments

This idea was sparked from hearing all of my peers' life stories and their paths to Yale from places that ordinarily are forgotten. Thank for sharing your experiences and paving a road for those ahead. Additionally, I would like to thank Professor Ruzica for pushing me to dive deeper into functionality and develop an app that is technically complex and up to par with my skills.

Sources

- "Why Representation Matters": https://www.buzzfeed.com/alannabennett/reminders-that-representation-really-is-important?utm_term=.vsEJNY82W#.fiq1K7XE0
- Design Inspirations: https://www.justinmind.com/blog/best-mobile-app-ui-designs-of-2017/
    - Spotify, Hipmunk
- Apple Inc. "Core Data." *Apple Developer* , Apple, developer.apple.com/documentation/coredata
- Ibragimova, Eleonora. "10 Best Practices for Designing User-Friendly Forms."*Theuxblog.com*, Medium, 3 Jan. 2017, medium.theuxblog.com/10-best-practices-for-designing-user-friendly-forms-fa0ba7c3e01f.
- https://www.appcoda.com/tdd-quick-nimble/

Existing Similar Projects

People are currently tackling this problem mainly with blogs that spotlight underrepresented individuals by certain fields or sectors.

- Spotlights Black Grad student experiences: https://justtrynagraduate.com/office-hours/
- Spotlights women in tech: https://medium.com/@WomenOfSiliconValley
- Spotlights underrepresented people in tech: http://www.techiesproject.com/