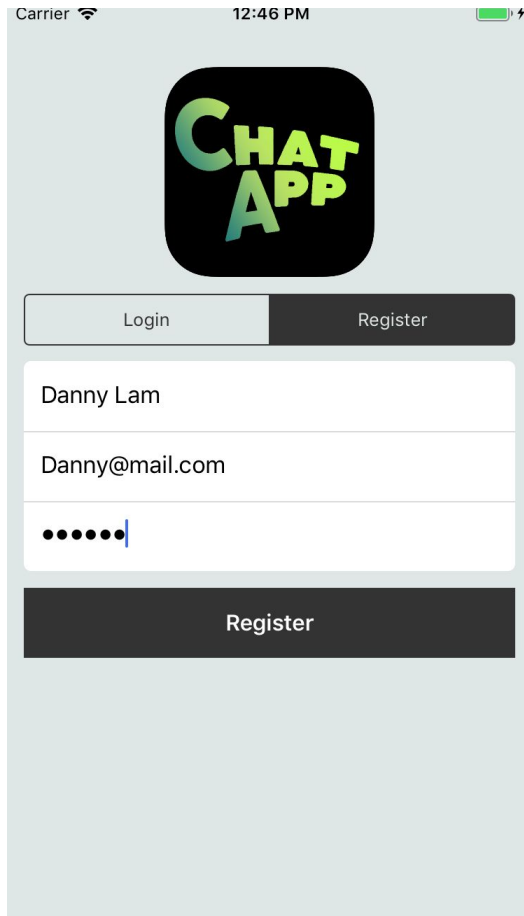


Chat App

Danny Lam | John Kang | Peter Sison

CISP53 | June 10, 2019 | Team Project



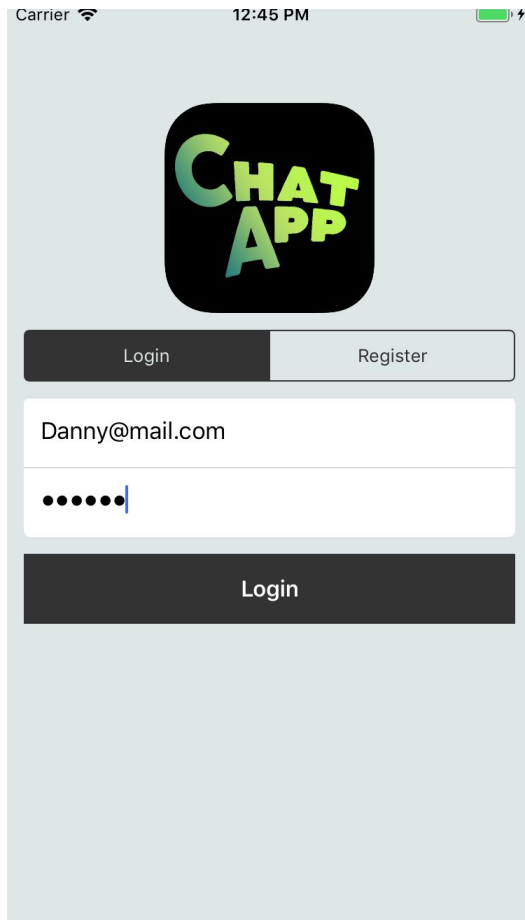
Registration

This is the first screen users view once the app loads. Registration is required for new users. The required fields are Email and Password.

Users can upload a profile pic also, which is done by clicking on the avatar icon just above the Login/Register segment.

Details

We utilize *Firebase* for Authentication (Login and Registration). Profile images use a Storage service which *Firebase* provides as well.

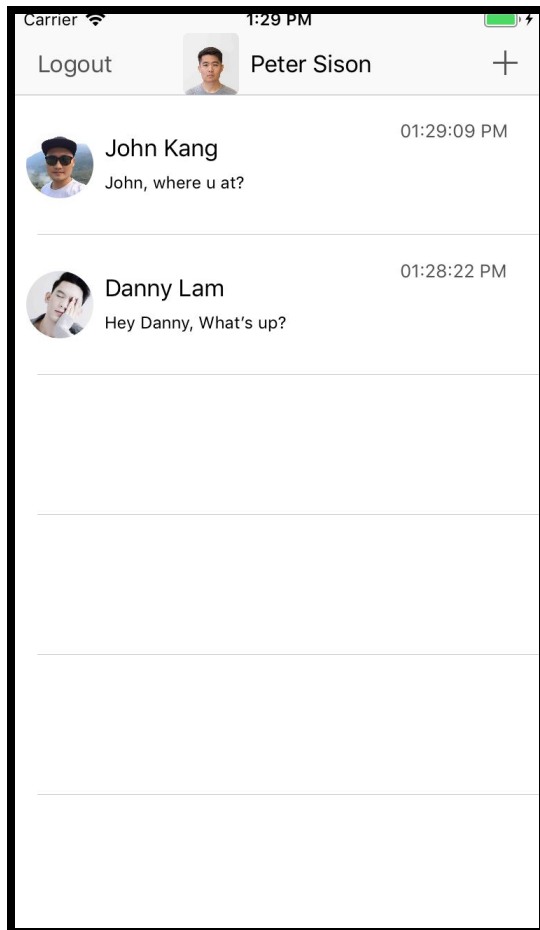


Login

By selecting the Login button from the segmented control, users are presented the Login form. A successful authentication directs users to their Active Messages screen.

Details

Email and Password fields are Authenticated against 'users' stored in the *Firebase* database. Analytical data, specifically creation date and login dates are recorded in *Firebase Authentication*.



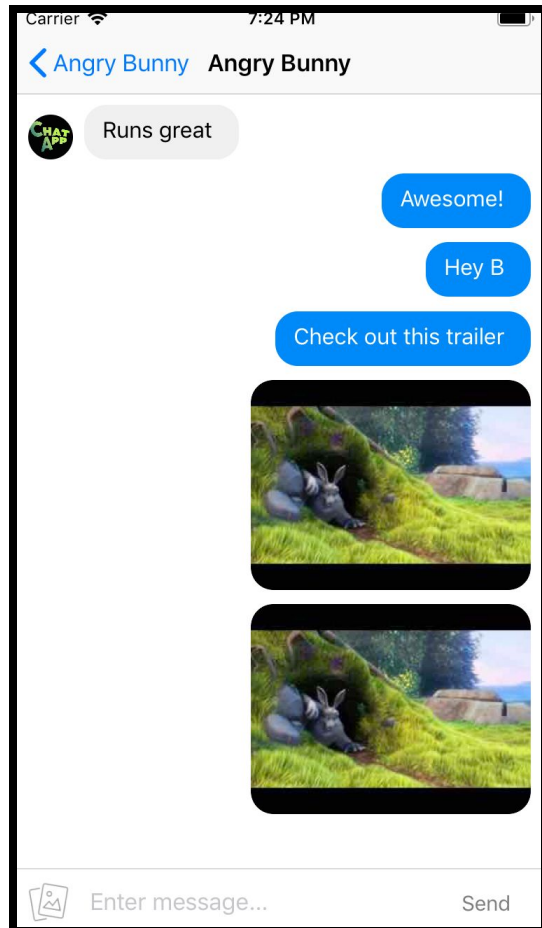
Messages

User's active messages are displayed here. In the navigation bar, the user's name and profile image are displayed. A Create New Message button is displayed on the right side, (plus sign). Logout button is displayed on the left.

Details

Messages are displayed through a *UITableViewController*.

We utilize a function to retrieve active and new messages within the "user-messages" child node by capturing the logged in user's UID and then retrieving all messages associated with that UID as well as the "chatPartnerId" and the message content.



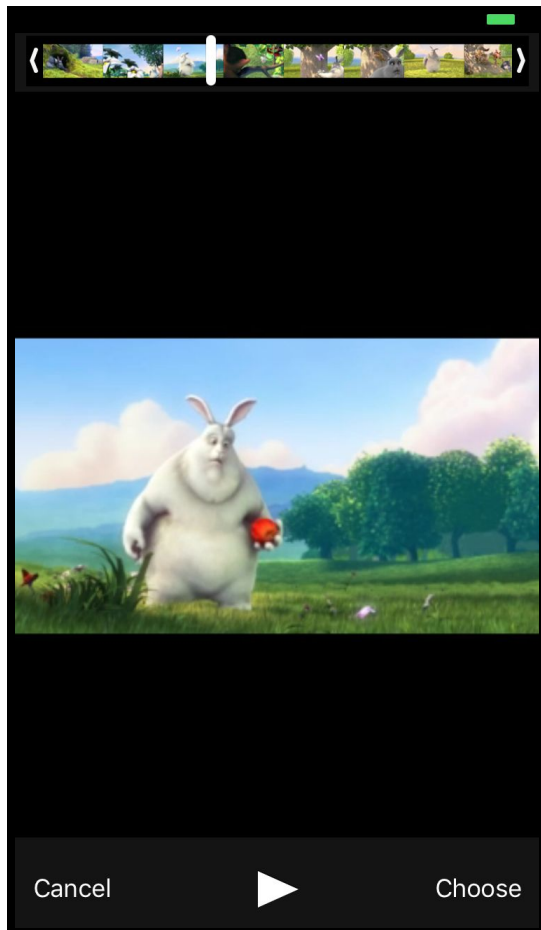
Create New | Reply To Message

By tapping the Create New icon or selecting an active message from the messages screen, user can compose a message by tapping the 'Enter message' input field. The device keyboard displays once the input field is tapped.

Another feature includes the ability to send images or video by selecting the image icon displayed in the left in the input field.

Details

In the View folder, the ChatInputContainer view controls UI for the message input field. The UserCell and ChatMessage views control the UI elements for the message content.



Send Image/Video Message

By tapping the image icon from the message input controller, users can select an image or video available on their device. Once a media type is selected, the user is prompted to either cancel or choose - Tapping choose inputs the media into the message and the user can tap send to send the image or video

Details

In the ChatLogController, we utilize an `imagePicker` method to enable selecting an image or video. We utilize *Firebase* Storage to store the media file. A url of its location in *Firebase* is captured and sent to the recipient user. *Firebase*

The app utilizes three services provided by *Firebase*. Authentication, Realtime Database, and Storage.

Our Realtime Database is organized by 3 groups: Users, Messages, and User-Messages.

Messages are assigned unique ids and the contents are stored as child node values of the assigned unique id.

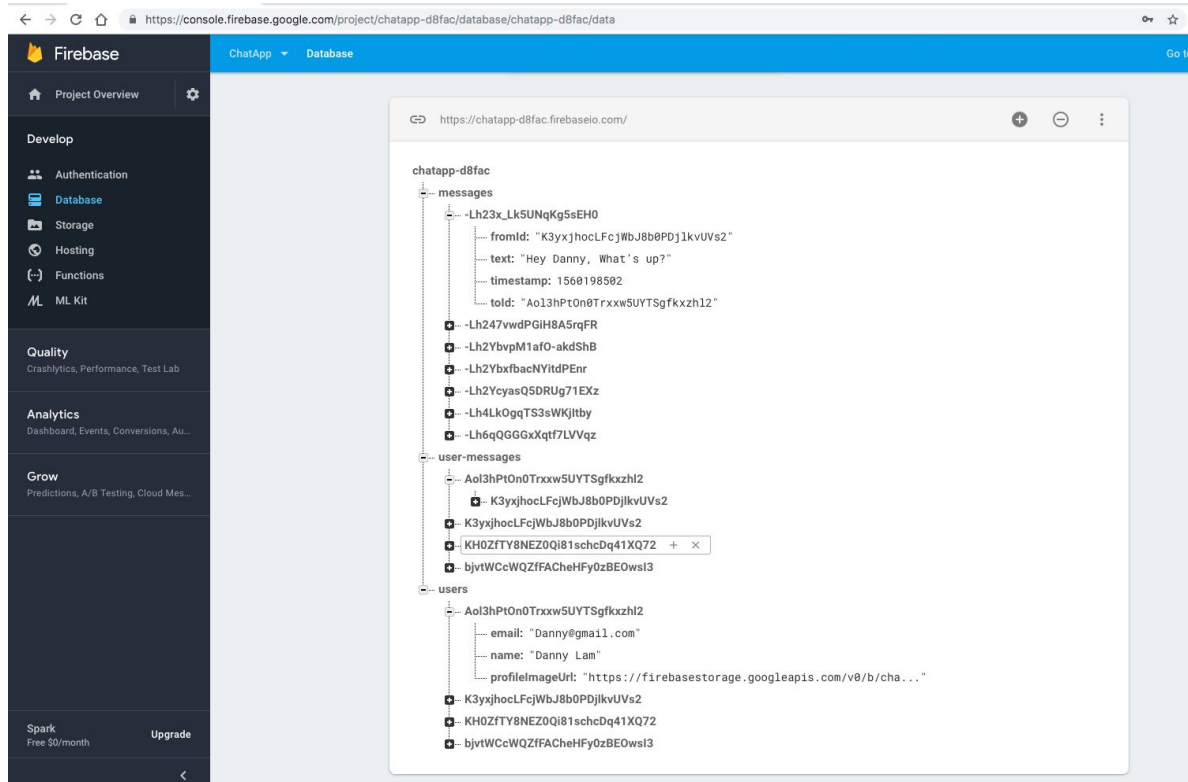
User-Messages are sorted by User and as child nodes to each user, a log of messages are stored

Authentication database is used for Signing in members and those records are stored there.

We use Storage to store images, video, and profile images. It is organized into those three groups respectively.

Firestore Screenshots

Realtime Database



Firebase Storage

The screenshot shows the Firebase Storage console for a project named 'ChatApp'. The left sidebar contains navigation links for Project Overview, Develop (Authentication, Database, Storage, Hosting, Functions, ML Kit), Quality (Crashlytics, Performance, Test Lab), Analytics (Dashboard, Events, Conversions, Au...), and Grow (Predictions, A/B Testing, Cloud Mes...). The main content area is titled 'Storage' and has tabs for Files, Rules, and Usage. The 'Files' tab is active, displaying a file browser interface. At the top of the file browser is a text input field containing the URL 'gs://chatapp-d8fac.appspot.com', an 'Upload file' button, and a folder icon. Below this is a table with columns: Name, Size, Type, and Last modified. The table contains three rows: a blue header row with a star icon and the text 'Default security rules require users to be authenticated' (with 'Learn more' and 'Dismiss' links), and two data rows for folders: 'message_images/' and 'message_movies/'. Below the table, there is a text label 'Location of default bucket: us'. At the bottom of the sidebar, there is a 'Spark' section with 'Free \$0/month' and an 'Upgrade' button.

ChatApp

Storage

Files Rules Usage

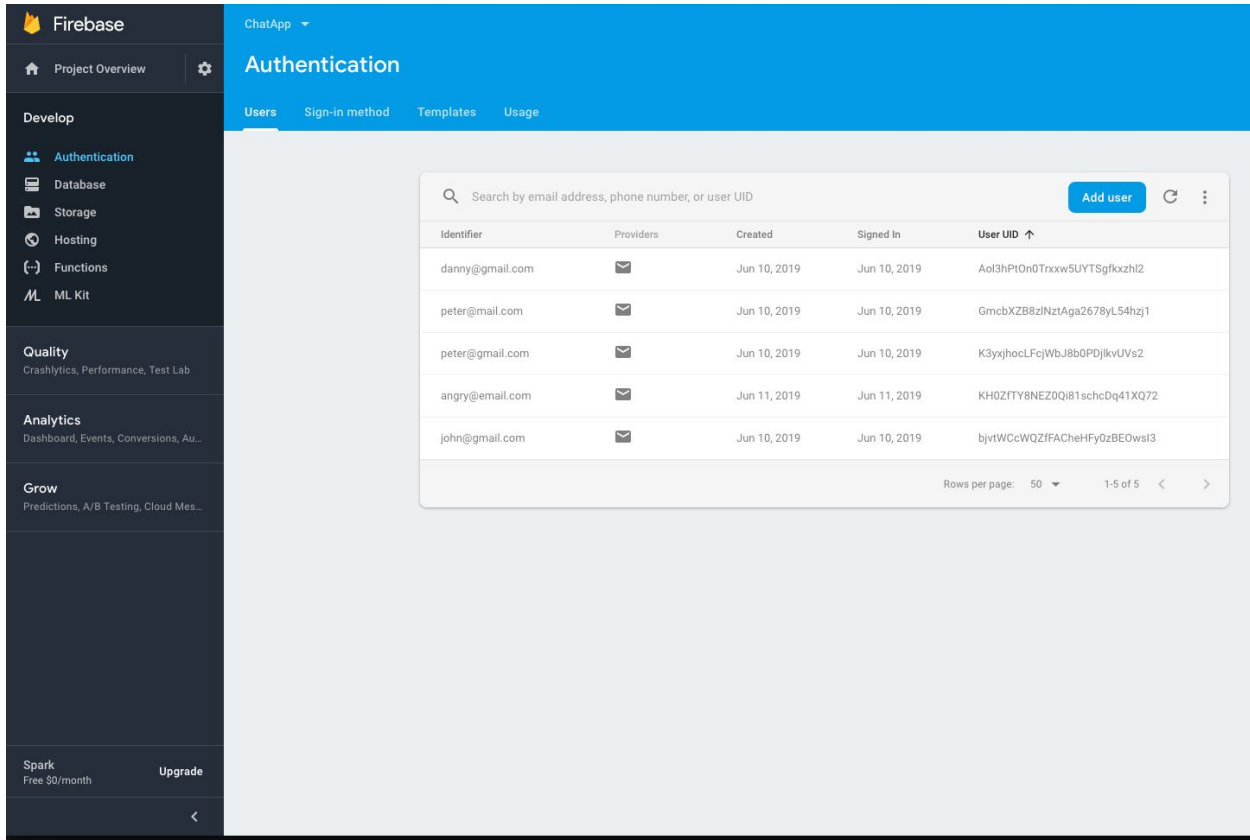
gs://chatapp-d8fac.appspot.com [Upload file](#)

<input type="checkbox"/>	Name	Size	Type	Last modified
<input checked="" type="checkbox"/>	★ Default security rules require users to be authenticated			Learn more Dismiss
<input type="checkbox"/>	message_images/	—	Folder	—
<input type="checkbox"/>	message_movies/	—	Folder	—
<input type="checkbox"/>	profile_images/	—	Folder	—

Location of default bucket: us

Spark
Free \$0/month [Upgrade](#)

Firebase Authentication



The screenshot shows the Firebase Authentication console. The left sidebar contains navigation links for Project Overview, Develop (Authentication, Database, Storage, Hosting, Functions, ML Kit), Quality (Crashlytics, Performance, Test Lab), Analytics (Dashboard, Events, Conversions, Au...), Grow (Predictions, A/B Testing, Cloud Mes...), and Spark (Free \$0/month, Upgrade). The main content area is titled 'Authentication' and has tabs for Users, Sign-in method, Templates, and Usage. The 'Users' tab is active, displaying a table of users with columns for Identifier, Providers, Created, Signed In, and User UID. A search bar at the top of the table allows searching by email address, phone number, or user UID. There is an 'Add user' button and a refresh icon. The table lists five users, all created on June 10, 2019, except for 'angry@email.com' which was created on June 11, 2019. The 'Rows per page' is set to 50, and it shows '1-5 of 5' rows.

Identifier	Providers	Created	Signed In	User UID ↑
danny@gmail.com	📧	Jun 10, 2019	Jun 10, 2019	Aol3hP1On0Trxxw5UYTSgfkxzhI2
peter@mail.com	📧	Jun 10, 2019	Jun 10, 2019	GmcbXZB8zINztAga2678yL54hzj1
peter@gmail.com	📧	Jun 10, 2019	Jun 10, 2019	K3yxjhocLFcjWbJ8b0PDjlkvUVs2
angry@email.com	📧	Jun 11, 2019	Jun 11, 2019	KH0ZfTY8NEZ0QI81schcDq41XQ72
john@gmail.com	📧	Jun 10, 2019	Jun 10, 2019	bjvtWCcWQZfFACheHFy0zBEOWsl3

Rows per page: 50 1-5 of 5