Final Report

By: Aidan Sullivan, Brandon Rau, Cayden Olsen, Dan Habiger
Client - Elvis Kimara
Instructor - Simanta Mitra
TA - Junhyung Shim

Requirements

Project Overview

Throwly is an app that aims to bring a new competitor into the online marketplace game. However rather than using real money Throwly wants to use a point based economy system. Points can be earned through giving things away of your own or just from when you sign up. Then you can use these points to get things that other people want to get rid of. Almost all of the points pricing is through Throwly's own ai system that can determine the value of an item through uploading a picture of it on the app.

Distri is a website that aims to help apps with their analytics. It can take any app and help show the owner of that app how the app is being used through analytics. While also trying to gamify the analytics through the use of a leaderboard to show who the top app users are.

Problem Statement

While Throwly is a fully developed and working app it still has a few things that it needs in order to improve itself. First of all it doesn't introduce the user to the app properly. The first time the app loads up it just takes you to a sign or register account page. Then it also doesn't verify that the new users are legitimate users. Which can cause people to make many accounts to abuse the points system through the free points given to new accounts. While also allowing people to make fake accounts with emails that might not even exist.

Currently there isn't a great way to track the analytics of apps out there on the market. On top of that there are so many different analytics you can track that it's hard to know what is important to track. Then how do we make the analytics more interactive in order to get more people to use the analytics and keep them engaged with what they want to track while also potentially being able to reward top users.

Functional Requirements

- A new onboarding experience for brand new users of Throwly
- Integrate an email service in order to authenticate users
- Create a functioning website for Distri
- Have working analytics tracking
- Have a leaderboard to help keep track of top users in an app
- Ability to add own projects to Distri

Non-Functional Requirements

- Have the onboarding process look professional
- Have Distri look professional
- Changeable point values for events on leaderboard
- Modular analytics package design

- Scalable backend architecture
- Minimal performance impact
- Secure authentication
- Reusable integration process

Design

The overall system design for this project was split into two connected systems: the Throwly mobile application and the Distri analytics platform. Throwly acted as the main application that used the analytics system, while Distri was designed to be a reusable analytics platform that could eventually support multiple applications.

The analytics system followed a one-way event pipeline. User actions inside Throwly generated analytics events which were passed through the `distri-analytics` npm package before being sent to Firebase backend services and Firestore storage. This design helped keep the analytics system modular, reusable, and scalable for future projects.

Overall System Flow

Frontend App
↓
distri-analytics npm package
↓
Backend/API
↓
Firebase Cloud Functions
↓
Firestore Database

Analytics Tracking Design

The analytics system was designed to track several different types of user activity throughout the application. The goal was to better understand user behavior while keeping the tracking system flexible and easy to expand in the future.

Analytics Features

- Session tracking
- Screen navigation tracking

- Page duration tracking
- User activity event tracking
- Event timestamps
- Session IDs
- Retention tracking
- Analytics dashboard support

The analytics package used a standard event structure so that all analytics data would be organized consistently. Every event was stored as an object containing important information about the action being tracked. This made it easier to process analytics data, rebuild user sessions, and analyze engagement patterns later.

Event Object Information

- `type` → category of event
- `name` → action or screen name
- `userId` → identifies the user
- `sessionId` → groups events from the same session
- `timestampMs` → exact time of the event
- `properties` → additional metadata related to the event

The system also supported automatic analytics tracking in several places throughout the app. For example, changing screens automatically generated page events while leaving a screen generated a page duration event. This reduced the amount of manual tracking code required inside the application.

Automatic Tracking Examples

- App opens
- Session starts
- Screen changes
- Time spent on screens
- User interaction events
- Login and logout activity

Firebase Backend Design

Firebase was used as the main backend infrastructure for the project. Firestore was chosen because it works well with React Native applications and allows flexible storage of analytics data. Firebase Cloud Functions were used to process analytics events, manage backend services, and support onboarding and authentication systems.

Firebase Services Used

- Firestore database
- Firebase Authentication
- Firebase Cloud Functions
- Firebase Emulator Suite
- Firebase backend services

The Firestore database structure was designed to keep analytics data organized and scalable. Each user contained multiple sessions, and each session contained multiple analytics events. This structure made it easier to organize long-term analytics data and rebuild user activity throughout the application.

Firestore Structure

- User
    - Sessions
        - Events
            - Event metadata
            - Timestamps
            - Navigation data
            - Duration data

Authentication & Onboarding Design

Another major part of the project focused on improving Throwly's onboarding and authentication systems. The onboarding experience was redesigned to better explain the app's purpose and point-based marketplace system to new users.

The authentication system used custom email verification codes and personalized email templates to improve security and create a more professional user experience. Firebase Cloud Functions and external email services were used to send verification emails and manage account verification logic.

Onboarding & Authentication Features

- New onboarding screens
- Email verification system
- Personalized email templates
- Verification resend cooldowns
- Attempt limits for verification
- Firebase-triggered emails

- HTML email templates

Distri Platform Design

The Distri platform was designed to provide a more interactive analytics experience compared to traditional analytics systems. Instead of only displaying analytics data, Distri also explored gamification concepts such as leaderboards, seasonal systems, and rewarding active users.

The platform was also designed so that multiple projects could eventually connect to the same analytics service using reusable API keys and npm package integration.

Planned Distri Features

- Analytics dashboard
- User leaderboard system
- Seasonal activity tracking
- Reward systems
- User retention analysis
- Project management dashboard
- Multi-project support

Work Done

Daniel Habiger

Analytics Package Development

- Developed the `distri-analytics` npm package
- Designed the package to be reusable across multiple applications
- Built a reusable analytics event schema
- Worked on package architecture and configuration systems

Analytics Tracking Systems

- Implemented session tracking
- Implemented page duration tracking
- Implemented navigation tracking
- Developed automatic event tracking systems
- Added support for timestamps and session IDs

Firebase & Backend Infrastructure

- Built Firebase analytics ingestion pipeline
- Connected analytics events to Firestore backend
- Organized Firestore analytics collections
- Worked on backend analytics architecture
- Integrated Firebase Cloud Functions into analytics systems

Debugging & Integration

- Integrated analytics system into Throwly
- Debugged analytics delivery and backend issues
- Worked extensively with Firebase Emulator Suite
- Improved analytics event reliability and consistency
- Tested analytics tracking across different app flows

Dan mainly focused on the analytics systems and backend infrastructure used throughout the project. This included designing the reusable analytics package, building the analytics event pipeline, and organizing how analytics data was stored and processed through Firebase services. A large amount of time was also spent debugging backend systems, improving event reliability, and making the analytics package flexible enough to support future projects and applications.

Cayden Olsen

Distri User Interface

- Designed and implemented majority of Distri's user interface, pages include:
    - Landing page
    - Signup/login
    - Project management page
    - Project overview page
    - Project users page
    - Project leaderboard page (later refined by Brandon)
    - Project customize page
    - Project settings page

Firebase Structure with Distri

- Worked with firebase to create collections and rules to allow the data management of:
    - Signup/login
    - Project creation/management
    - User management

    - Game season management and rewards

Distri Planning and Design

- Created multiple flowcharts to visualize the common flow that Distri users should experience while using the product
- Used sketches and design programs to help design Distri's pages

Distri Gamification System

- Created the main idea and structure behind Distri's unique gamification element
- Brough the system to life by creating seasons which contain custom dates, leaderboards, and custom rewards for the users. The leaderboard was later refined by Brandon to make the experience overall better

Cayden's work mostly focused on bringing Distri's public website to life. Through planning, designing, building, and backend connection this was made possible. By using the provided research from Brandon and the NPM package from Dan, Cayden used his resources to create a unique and low stress website allowing users to manage their applications analytics with ease.

Brandon Rau

Analytics Development Research

- Researched what different major app developers focus on analytics
- Researched different software development kits
- Did more research into how to track the analytics found to be important
- Made a write up determining what analytics are important
- Made a paper discussing the importance of each analytic and determining why we should track it

Leaderboard Refinement

- Implemented spam identifiers
- Determined good bounds for deciding if someone is spamming
- Connected to Throwly to see how it reacted to the app
- Implemented a way to remove potential spammers from the leaderboard

Debugging

- Tested that flagging works
- Tested that removing from leaderboard works
- Reworked spam bounds
- Removed automatically removing potential spammers to protect users

Brandon mainly focused on researching what analytics we should track throughout the project. This included looking into potential software development kits to mimic, and learning many different companies goals for analytics and what they tracked to reach those goals. Finally made a list of the main analytics we should focus on. Also spent time helping refine the analytics leaderboard, by filtering for potential spammers to prevent users from abusing the leaderboard scoring.

Aidan Sullivan

Onboarding Experience

- Built a new onboarding flow for first-time Throwly users
- Created multi-page screens introducing the points-based marketplace and AI pricing
- Added persistent storage so onboarding only shows on first launch

Email Service Integration

- Integrated the Firebase Trigger Email extension with Mailgun SMTP to handle all outbound email from Throwly
- Built the sendEmail service that renders templates from Firestore and writes to the mail outbox collection
- Designed HTML email templates with Throwly branding for welcome emails and authentication
- Configured the Firebase extension environment and Secret Manager SMTP credentials

User Authentication

- Built a custom 6-digit email verification flow as a replacement for unverified signups
- Implemented sendVerificationCode and verifyEmailCode Cloud Functions with hashed codes
- Created the verification screen with resend cooldown and attempt limits
- Updated AuthContext and CreateAccountScreen to gate access on verified emails
- Added a Firestore trigger that sends the welcome email after verification

Aidan focused on the Throwly app as well as introducing new users to the app with the onboarding, authentication, and the email infrastructure. This included building the first-launch onboarding flow, replacing unverified signups with a 6-digit email verification system, and integrating Mailgun so the app could send branded verification, welcome, and preference-based emails.

Results

For Throwly we ended making a brand new onboarding experience to help new users learn what Throwly is about and how it works. While also adding this to hopefully boost user retention.

Then we also added in an email service through mailgun which is how we now authenticate email addresses. While we can also use this service to send personalized emails to the users about potential updates or promos happening in Throwly.

We made a whole new website for Distri that is super easy to add to any project. As we made it an npm library in order to add it to an app. While also adding the ability to currently track sessions, screen navigation, page duration, user events, event timestamps, and session ids to track these accurately. We also added a leaderboard to each project that is added on Distri to help gamify the analytics so that way it will keep the developer more engaged with the analytics to help them improve the apps.