EduHub Maintenance Report

02.11.2021

—

Team 7:

Chintamani Masthanaiah - 191CS115

Venkata Sravani - 191CS223

Kruthika K Sudhama - 191CS224

V Madhan Kumar - 191CS260

Vinesh S Talpankar - 191CS265

Overview

Software maintenance is 'the process of modifying a software system or component after delivery to correct faults, improve performance or other attributes, or adapt to a changed environment'.

Maintenance is always necessary to keep software usable and useful. Often there are very tight constraints on changing software and optimum solutions can be difficult to find. Such constraints make maintenance a challenge; contrast the development phase, when designers have considerably more freedom in the type of solution they can adopt.

Goals

1. To adapt a tool for Mobile app error tracking, performance monitoring, and reporting run time errors in a single view, giving a holistic overview of our application's health in real-time.
2. Better handling of security updates, bug fixes and audits against the latest versions of iOS and Android.

Specifications

Goal 1:.

Enabled Firebase Google Analytics:

- We are using Firebase authentication for Signin and Signup purposes. So we have enabled firebase google analytics to check users activity, bug reports and other details related to authentication.
- Google Analytics helps us understand how people use our ios, or Android app. The SDK automatically captures a number of events and user properties. Once the data

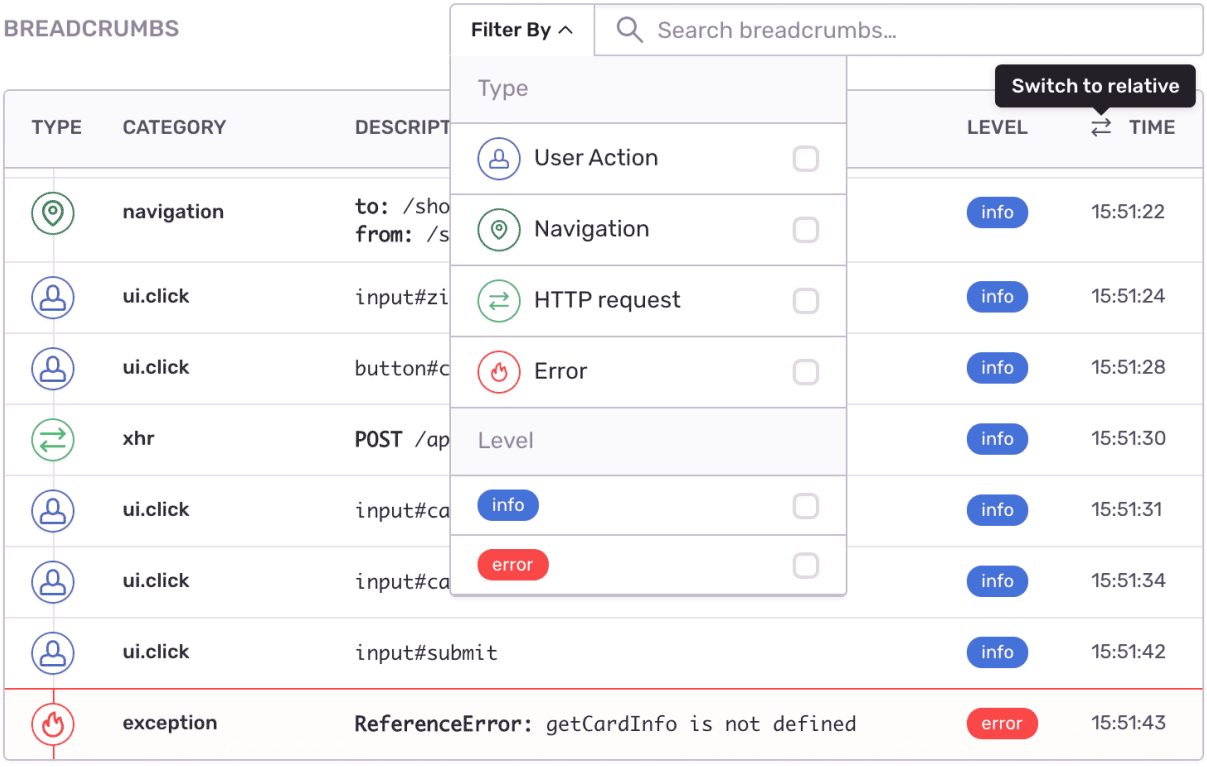
is captured, it's available in a dashboard through the Firebase console. This dashboard provides detailed insights about our data — from summary data such as active users and demographics.

- In our case it can provide the report on the number of logins of users and their frequent usage of our app.

Using sentry for performance monitoring and error tracking:

Using Sentry for the app error tracking, performance monitoring, and reporting run time errors in a single view, giving a holistic overview of your application's health in real-time. Correlating errors with releases, tags, and devices, to solve problems quickly, decrease churn and improve user retention.

The sentry report looks as below



Goal 2.:

For Security updates:

- We keep current with the latest Flutter SDK releases. We regularly update Flutter, and these updates may fix security defects discovered in previous versions. Check the Flutter change log for security-related updates.
- Keep our application's dependencies up to date. We make sure we upgrade our package dependencies to keep the dependencies up to date. Make sure to check periodically to see if our dependencies have had security updates, and update the pin accordingly.
- Keep our copy of Flutter up to date. Private, customized versions of Flutter tend to fall behind the current version and may not include important security fixes and enhancements. Instead, routinely update our copy of Flutter.

About updating our app for the latest versions of iOS and Android.

We can deploy our flutter app to iOS, Android and Web. Flutter supports building ahead-of-time compiled libraries for x86_64, armeabi-v7a, arm64-v8a so that app runs without any problems in new android versions.

Software maintenance activities can be classified as:

1. Corrective maintenance:

   Corrective maintenance removes software faults. It should be the overriding priority of the software maintenance team.

2. Perfective maintenance:

   Perfective maintenance improves the system without changing its functionality. The objective of perfective maintenance should be to prevent failures and optimise the

software. This might be done, for example, by modifying the components that have the highest failure rate, or components whose performance can be cost-effectively improved.

3. Adaptive maintenance:

Adaptive maintenance modifies the software to keep it up to date with its environment. Users, hardware platforms and other systems all make up the environment of a software system. Adaptive maintenance may be needed because of changes in the user requirements, changes in the target platform, or changes in external interfaces.

Basically app maintenance includes:

- ### App Update:

  App updates are something that gives users something more advanced. But it depends on the business owner when they want their app to be upgraded. Depending on the performance monitoring reports, user reports, we will classify the issues/features into three categories

  a. critical and urgent

  b. critical and routine

  c. Non-critical and release updates accordingly for better user satisfaction and ease of use.

- ### Bug-Fixes:

  No matter how superior our app development team is, there is no app that is bug-free at all. And once our app is launched, it becomes tough to fix those bugs so it is important to deeply analyze the app and detect the bugs before its launch.

- ### Design Changes:

  App design is something that actually attracts users but over time it gets out-dated, so it is important to maintain your app and change the design to make it more accessible for the users. On the one hand, any cross-platform framework makes it possible to share codebase between an internal network of programmes. On the other hand, no other application besides Flutter provides a way to share both the UI code and the UI itself between various platforms. Such a rendering process allows you to easily create an app that looks native on every platform. Flutter adapts easily to different screen dimensions, positively affecting the overall experience of its users. In brief, Flutter offers both sharing the UI and business logic between various platforms and adapting to different phone sizes. One codebase for all these principles saves time, effort and money of the company while not affecting the performance of the final product.

Milestones/Bug Fixes

Software maintenance report

There are 3 kinds of maintenance activities

1. Corrective maintenance

- Users of the application who would like to sign in the application need to submit a SHA certificate in the firebase authentication. This problem can be avoided by changing the security group to all the IP addresses.
- When the application gets disconnected to the internet while using the application. Connection to the database is also disconnected, which on reconnecting to the internet doesn't connect back to the database. To connect to the database requires restarting the situation. This problem can be addressed by saving the state of the

application in the database and restoring that state when the connection is back online.

- When a new user is signed in with the help of email and password, the details like contact number and username do not get inserted into the database until the user registers for the event. To solve this, the application needs to take these details as default or they need to be asked during the registration process.
- When the login fails, the user needs to get back to the initial page if the details about the user are not found in the database but found in the firebase. This problem can be eliminated by entering some data into the database when a user creates an account.
- Users are allowed to have 10 active sessions with the database. If it exceeds there is some problem with the data given by the user which is violating. This needs to be captured by monitoring the database sessions frequently.

2. Perfective maintenance

- The place taken by the events in the homepage would tire the user due to long scrolling when the count of the total events increases. This block needs to be shortened either by decreasing the font or by changing the design of the block.
- The application after release, we provide the recommendation system based on the data we have in the database. This can add extra features to the account information which is to be stored in the database. This may significantly change the database schema structure and new tables need to be generated for the recommendation.
- The delay between the transitions needs to be reduced. This can happen to generate parallel connections with the firebase as well as database when that is possible. Few places this may not work especially during registration and sign up. But this can be done when the account information is updated.

3. Adaptive maintenance

- Since both database and firebase is online, we should make sure the application will satisfy the requirements of the new updates
- When the application grows longer, we should be in a state to expand the database which can overcome the limited features with the present database. This can be done either by creating another database.

App Updates:

1. Critical or Urgent

- Delay between the pages needs to be reduced. This can be solved by having a good internet and also choosing faster protocol while communicating with the server.
- The admin who would like to register for any event is not possible with the current edition as we thought the admins would be separate from the users. Now when the admin wants to apply for any event, we should also provide that facility to users.
- Padding problems when the application runs on the different interfaces should be centralised. This can be fixed by choosing the padding size as a ratio of the screen size or extended functionality of the widget.

2. Critical and Routine

- There is a lot of space wasted on the UI of the application which should be developed. But this is not compulsory. Finding a better widget can remove this problem.

- Customer contact facilities need to be implemented or a help centre which would be useful for the customers.

3. Non-Critical and release updates

- Better UI based on the customer request.

Challenges in maintenance

1. Updates in the Database or Firebase authentication need to change the application accordingly to run smoothly. We should be checking the databases and firebase for any updates.
2. New changes in the technology in the servers should be noticed as early as possible.
3. Any updates in the flutter widgets or UI need the application to deploy a new version for the better experience to the user.