# Project Report

# 24-Bits Event Management App

Developed by Tripple Logic Team

Lucky Asenoguan

Theresa Alekwe

Dylan Nchungong

## 1. Project Aim and Description

### Aim:

To develop a cross-platform event management application using Flutter and Firebase, enabling users to discover, create, and manage events, purchase tickets, and allow admins to oversee users and event operations.

### Description:

The app allows users to register/login (via email, Google, or GitHub), browse and search for events, view event details (with map integration), purchase tickets, and manage their profiles. Admins can manage users, events, and tickets, ensuring smooth event operations and user experience. The app supports both web and mobile platforms.

## 2. Functional and Non-Functional Requirements

### Functional Requirements

User registration and login (email, Google, GitHub)

User profile management (edit profile, upload image)

Event creation, editing, and deletion (admin/organizer)

Event listing, search, and filtering

Event detail's view (with Google Maps location)

Ticket purchase and management (unique reference code, QR code)

My Tickets page (view, search, add by reference)

Admin dashboard (user management, ticket management, event management)

Firestore integration for data storage

Email notifications for ticket purchase

## Non-Functional Requirements

Responsive UI for web and mobile

Secure authentication and data access (Firestore rules)

Fast and reliable performance

Cross-platform compatibility (Android, iOS, Web)

Accessibility and usability

Data validation and error handling

Scalability for future features

## 3. User Stories

As a user, I want to register and log in using email, Google, or GitHub so I can access the app.

As a user, I want to browse and search for events so I can find events of interest.

As a user, I want to view event details, including location on a map, so I know where and when the event is.

As a user, I want to purchase tickets and view them in my account.

As a user, I want to edit my profile and upload a profile picture.

As an admin, I want to create, edit, and delete events.

As an admin, I want to manage users (view, promote, deactivate, delete).

As an admin, I want to view and manage all tickets.

As an admin, I want to verify tickets by reference code.

## 4. Test Cases

Test Case ID - Description - Steps - Expected Result

TC01 - User Registration - Register with email/password - User account created, redirected to home

TC02 -- Google Login -- Login with Google -- User logged in, profile created if new

TC03 -- Event Creation - -Admin creates a new event --Event appears in event list

TC04 -- Ticket Purchase -- User buys a ticket -- Ticket appears in My Tickets

TC05 -- Profile Edit -- User updates profile info -- Changes saved and shown

TC06 -- Ticket Search -- User searches ticket by reference -- Ticket found and displayed

TC07 -- Admin User Management -- Admin deactivates a user -- User cannot log in

TC08 -- Map Display -- Event with coordinates shows map -- Map with marker is visible

TC09 -- Web Payment -- User tries to pay on web -- Card input disabled, message shown

TC10 -- Mobile Payment -- User enters valid card on mobile -- Pay Now button enabled

## 5. Individual's Role and Responsibilities

Theresa - Project Manager | Task assignment, timeline, documentation, team meetings

Lucky - Lead Developer | Flutter architecture, authentication, Firestore integration

Dylan -- UI/UX Designer | App design, responsive layouts, user flows

## 6. Task Distribution Document

Lucky---App setup and Firebase configuration, also initialize Flutter project, set up Firebase Configure authentication.

Dylan—Developed the UI/UX Design and and user flows, | Authentication (email/social), Implement login/registration with email, Google, GitHub |

Theresa created CRUD operations which includes Event creation, editing, deletion, Firestore integration.

Lucy developed the Ticketing System which includes Ticket purchase, QR code, reference code, My Tickets page

Dylan integrated the Admin Dashboard and User/event/ticket management screens

Dylan did the Testing & QA and Write/execute test cases, report bugs |

Theresa prepared Documentation Project report, user stories, requirements, README |

## 7. Task Distribution Explanation

Theresa coordinated the project, ensured documentation was up to date, and managed the team's progress.

Lucky handled most of the Flutter and Firebase development, including authentication, event and ticket features, and admin tools.

Dylan designed the app's UI/UX, ensuring a modern and user-friendly experience.

Theresa/Lucky/Dylan were responsible for testing, creating test cases, and ensuring the app met quality standards.