

## Introduction

Two types of performance testing were carried out on the mobile application:

1. Performance Profiling: Used to determine how much of a device's resources would the application consume when being run. The focus was on memory consumption.
2. Load Testing: The database was tested to determine how it would react to an increased level of user activity over a prolonged period.

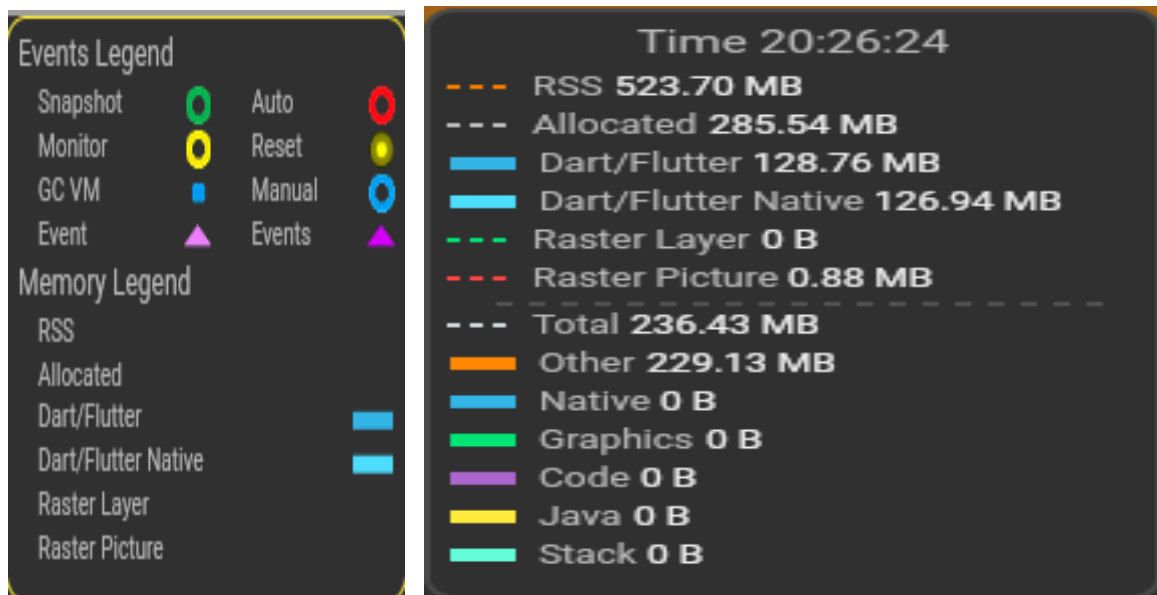
## Performance Profiling Report

**Tool Used:** Flutter DevTools

### Procedure for Testing:

Run the application and use it constantly for 5 minutes performing actions a normal user would. These actions include creating adverts, placing bids, uploading quotes, and communicating via chat.

### Legend



RSS = Resident Set Size

GC = Garbage Collection

### Snapshots of timeline of memory usage

Diagram 1

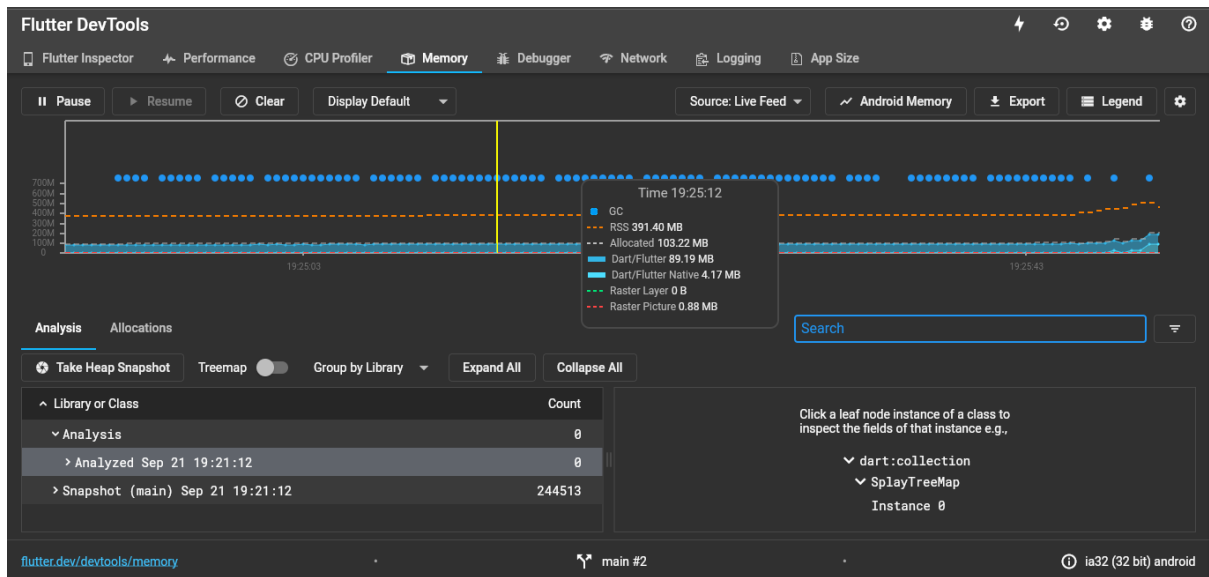
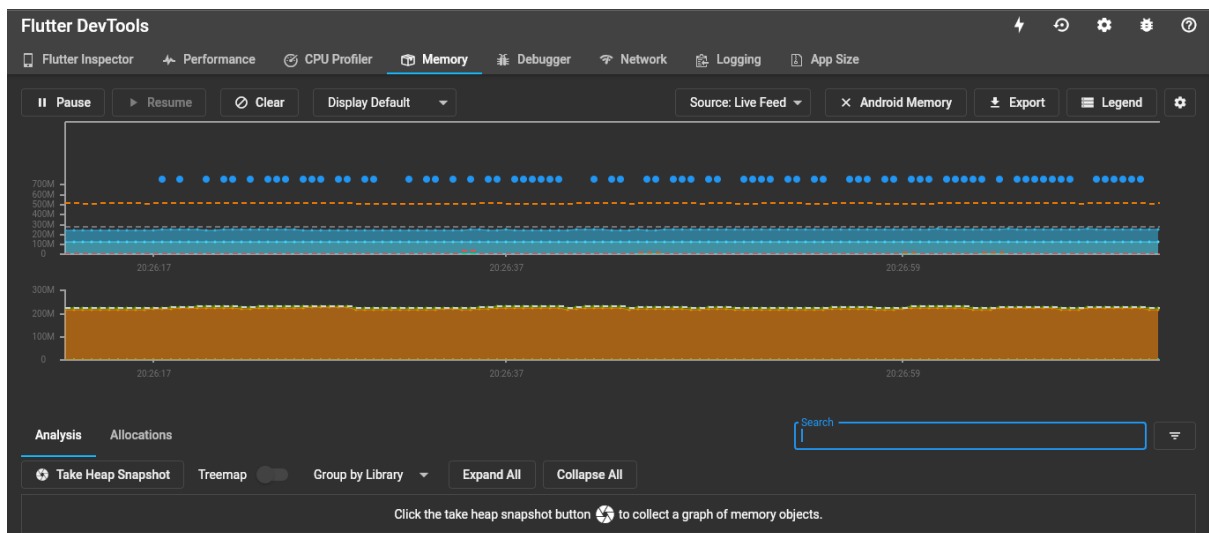


Diagram 2



## Review of Results

There are two things which are of importance in the above graphs, the android memory usage which is represented by the orange part(2<sup>nd</sup> graph) in Diagram 2 and the Garbage Collection rate represented as the blue circles above the orange dotted lines in Diagram 1 and Diagram 2.

The first observation is about the Android Memory. The following 2 things can be observed:

1. The memory usage hovers close to 200 Mb of RAM
2. The memory usage is consistent and there are no sudden spikes in memory usage.

These are very positive indications as it shows that the application is not memory intensive and can be used on lower end devices that do not have much RAM. It is also a good indication that there is no risk of the application crashing due to a sudden spike in memory usage for devices with low amounts of RAM as the usage is consistent.

The second observation is that garbage collection is initiated very frequently therefore allowing for space to be freed up quickly which is a positive for the application.

# Load Testing

## Diagrams



Diagram 3: Timeline of Database Write Capacity

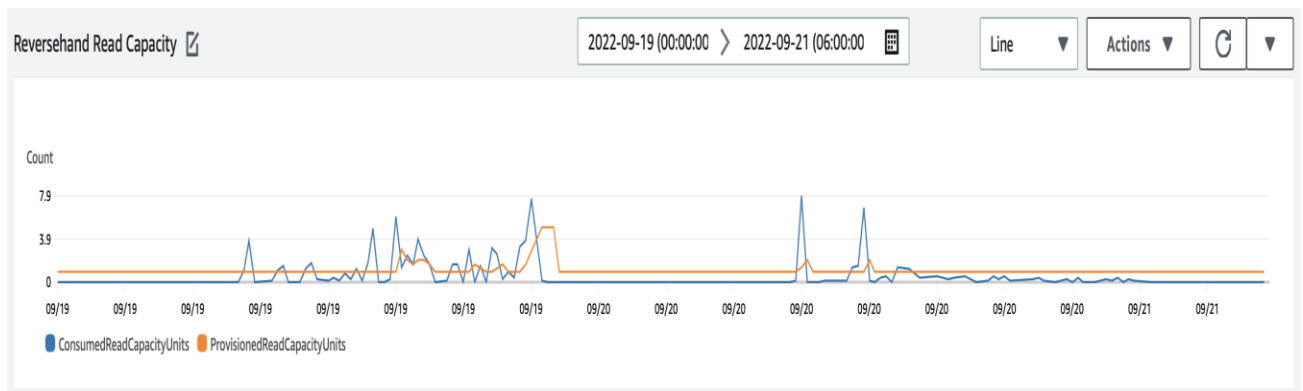


Diagram 4: Timeline of Database Read Capacity

## Procedure to Test

Run a test to simulate the actions of a user and run it consecutively for 5 mins.

## Results

Looking at the second diagram for read capacity for the database, the blue line represents the consumed read capacity for the database and the orange line represents the provisioned read capacity for the database.

It is seen that on the 19<sup>th</sup> of September where there is a sudden spike in the consumed read capacity because of the test that was carried out, DynamoDB automatically increased the provisioned read capacity to cope with increased number of reads.

This confirmed for the application that should the application experience a sudden surge of user activity the database will be able to handle the increased the volume without having to worry about the system being overloaded and crashing.