Mobile TAAS

CMPE 281- Cloud Technologies

Group 4 Members:

Amit Kamboj- Mobile Test Project Management.

Nehal Sharma- Tester Oriented and Project Oriented Community Service, Chatkit Pusher(Chat Integration-3rd party), Google Maps API.

Pranjal Sharma- Tester Runner.

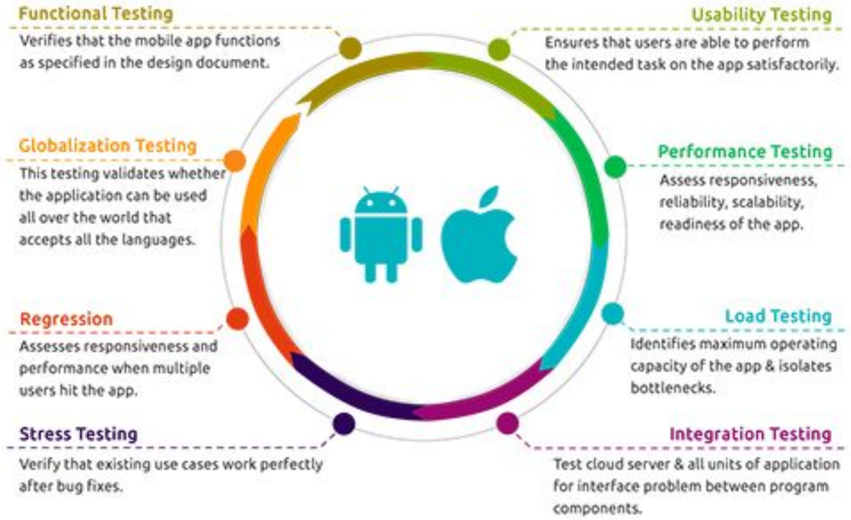
Rachit Saxena- Crowd Sourced Tester Management.

Agenda

- What is MTaas
- Why Taas
- Objectives
- System Infrastructure Diagram
- Database Design
- Extra Credit Activity
- Demo
- Q&A Session

What is MTaas ?

- On Demand testing of application by users on pay as you test basis.
- Scalable Mobile testing in accordance with pre-defined SLAs.


Functional Testing
Verifies that the mobile app functions as specified in the design document.
Usability Testing
Ensures that users are able to perform the intended task on the app satisfactorily.
Globalization Testing
This testing validates whether the application can be used all over the world that accepts all the languages.
Performance Testing
Assess responsiveness, reliability, scalability, readiness of the app.
Regression
Assesses responsiveness and performance when multiple users hit the app.
Load Testing
Identifies maximum operating capacity of the app & isolates bottlenecks.
Stress Testing
Verify that existing use cases work perfectly after bug fixes.
Integration Testing
Test cloud server & all units of application for interface problem between program components.

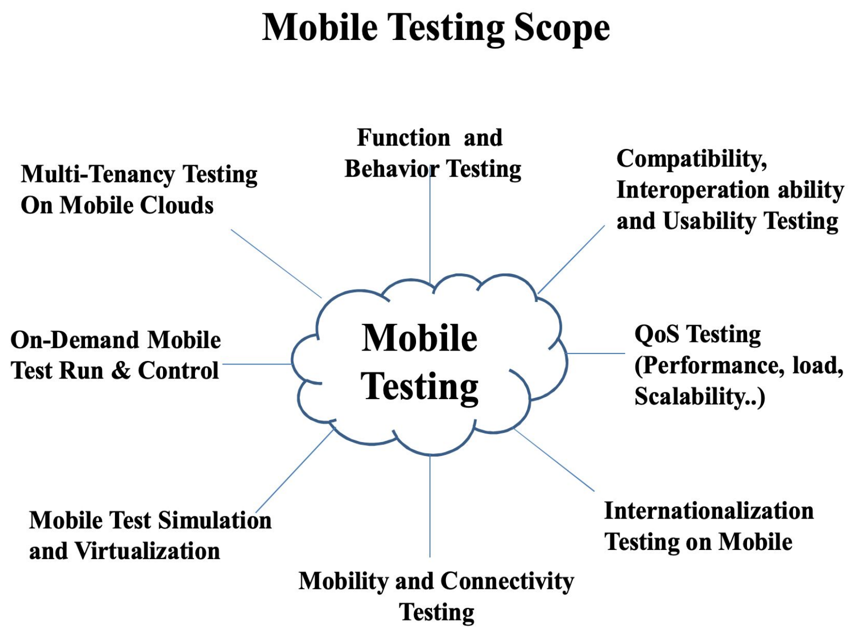
Why MTaas?

Different testers with different domain and experience.

Multi-Tenancy.

Pay-as-you-test-model.

Configuration & Customization


Mobile Testing Scope
Function and Behavior Testing
Multi-Tenancy Testing On Mobile Clouds
Compatibility, Interoperation ability and Usability Testing
On-Demand Mobile Test Run & Control
Mobile Testing
QoS Testing (Performance, load, Scalability..)
Mobile Test Simulation and Virtualization
Internationalization Testing on Mobile
Mobility and Connectivity Testing

Objectives:

To build a crowd based testing community which is scalable.

Manager manages projects and crowd testers.

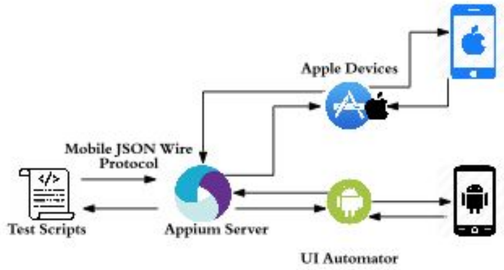
Admin manages infrastructures requirements such as hardware, network and bandwidth.

Tester and Project oriented Community services to facilitate communication and collaborative learning

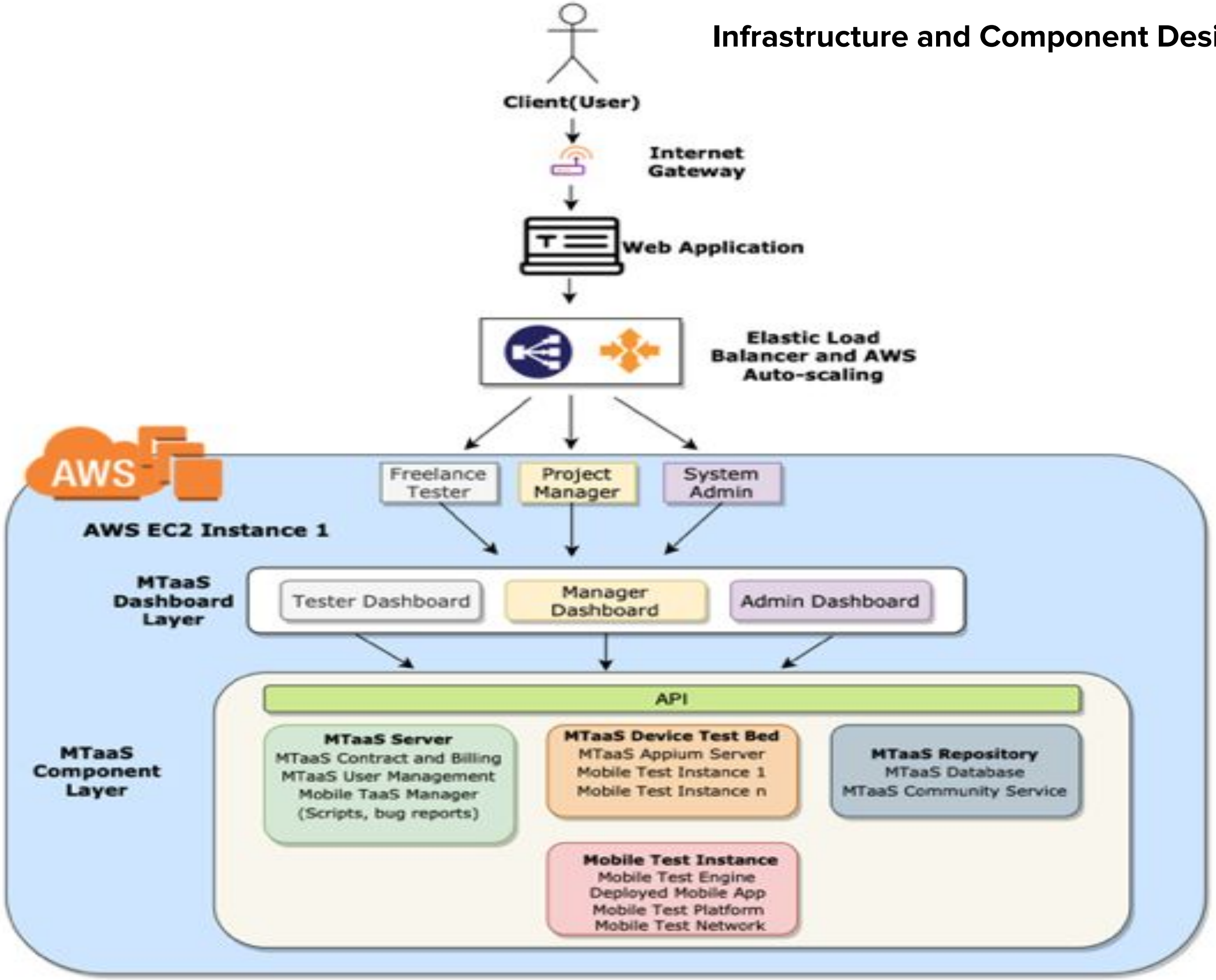
To build an efficient payment metric for services used.

Infrastructure and Component Design

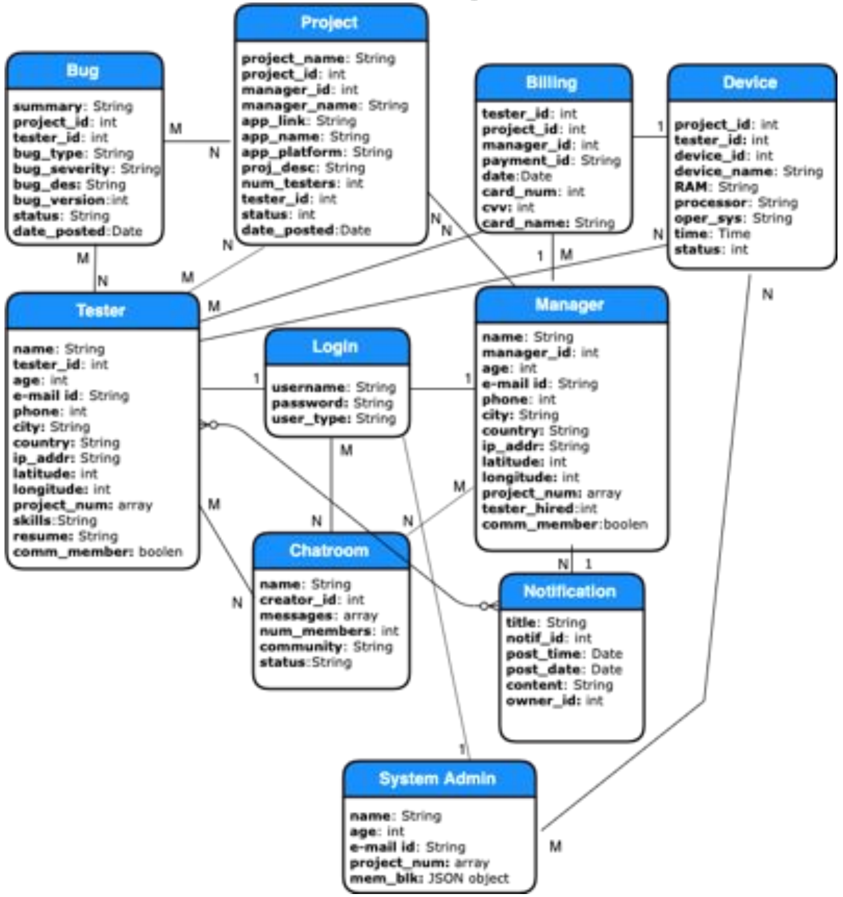
System Diagram:



Testing using Appium Server



Database Design:



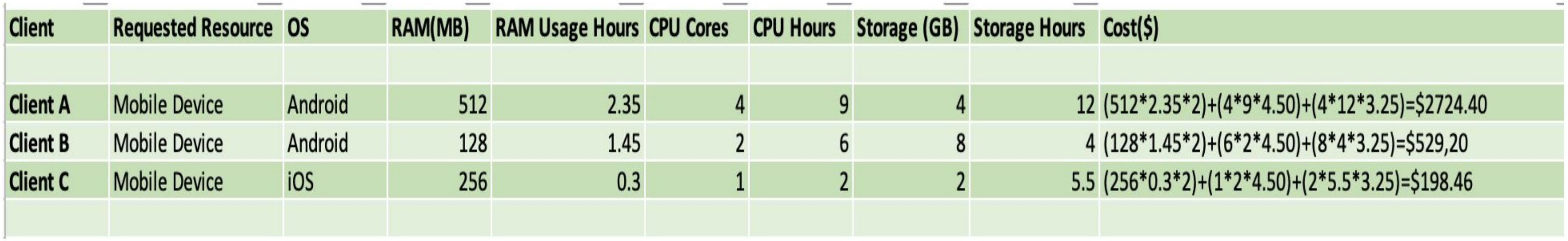
Indexing Strategy used:

- Most of the queries performed by the freelance tester and project managers based on the Project.
- Designed an indexing scheme on the ‘project_id’ field, since it uniquely identifies each project.
- Created a single key index as follows, which greatly increased the query performance:

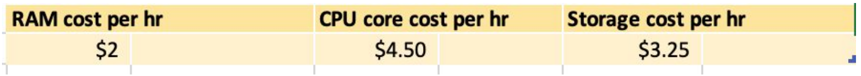
db.projects.createIndex({“project_id”: 2345})

Billing Metric (based on resources and bug severity)

Resource Calculation and Charges:



Client	Requested Resource	OS	RAM(MB)	RAM Usage Hours	CPU Cores	CPU Hours	Storage (GB)	Storage Hours	Cost($)
Client A	Mobile Device	Android	512	2.35	4	9	4	12	(512*2.35*2)+(4*9*4.50)+(4*12*3.25)=$2724.40
Client B	Mobile Device	Android	128	1.45	2	6	8	4	(128*1.45*2)+(6*2*4.50)+(8*4*3.25)=$529,20
Client C	Mobile Device	iOS	256	0.3	1	2	2	5.5	(256*0.3*2)+(1*2*4.50)+(2*5.5*3.25)=$198.46



RAM cost per hr	CPU core cost per hr	Storage cost per hr
$2	$4.50	$3.25

The reimbursement cost for high severity and critical bugs is determined by each project manager and can be different for different projects.

Technologies Used:

Cloud Technologies Used: Amazon Web Services EC2, Classic Load Balancer.

MongoDB: MongoDB Cloud Platform Atlas

Front End: ReactJs, Bootstrap 4

Backend: ExpressJS, Appium, AVD

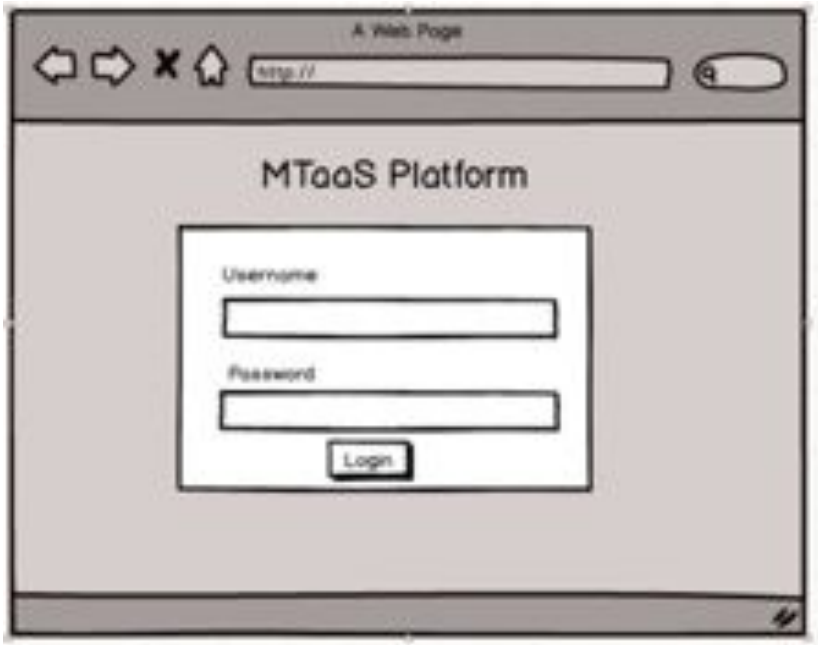
Payment Gateway: Paypal

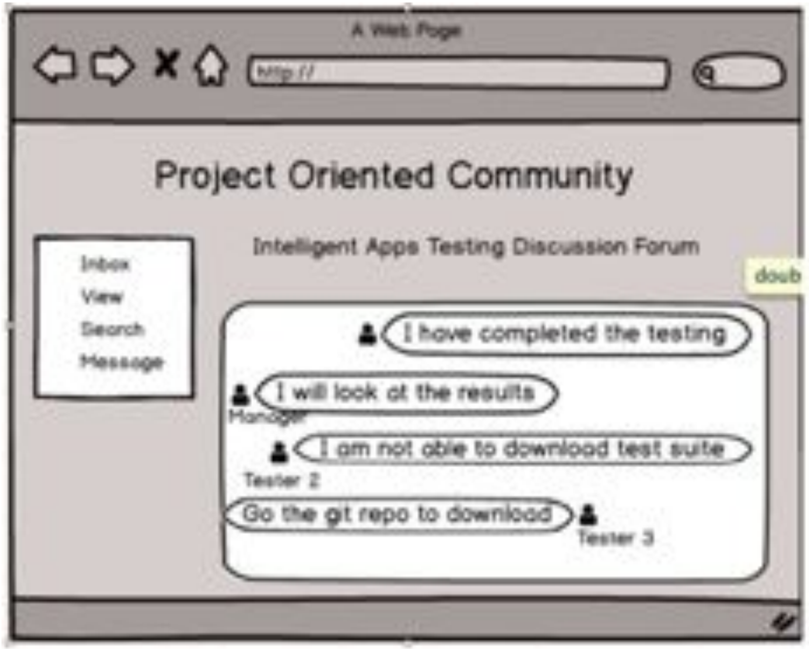
Chat 3rd Party Integration: Pusher Chatkit

Google Maps API for Map View

Graphical User Interface

Low Fidelity Prototype Screens designed using Balsamiq:




A Web Page
http://
Project Oriented Community
Intelligent Apps Testing Discussion Forum
Inbox
View
Search
Message
doub
Manager
Tester 2
Tester 3

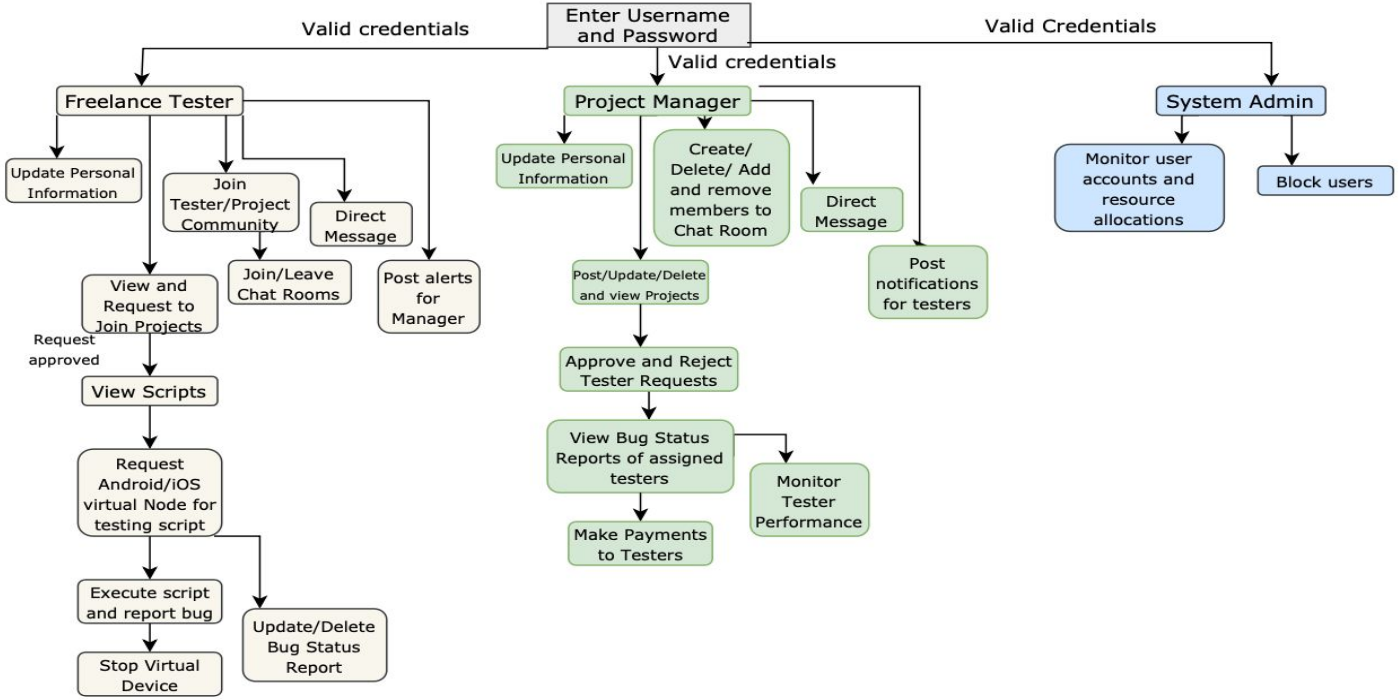
I have completed the testing

I will look at the results

I am not able to download test suite

Go the git repo to download

Graphical User Interface Functions

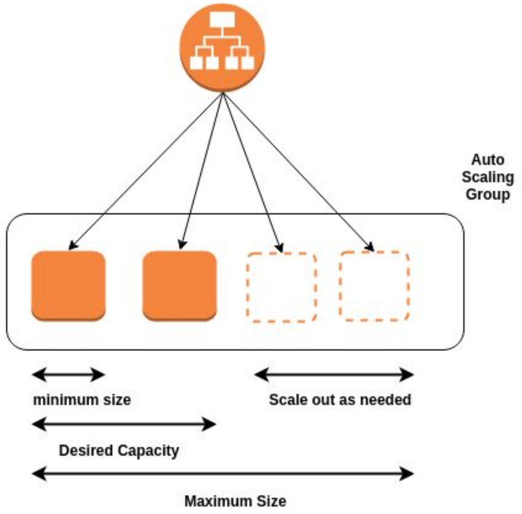

Enter Username and Password
Valid credentials
Valid credentials
Valid Credentials
Freelance Tester
Update Personal Information
Join Tester/Project Community
Direct Message
Post alerts for Manager
Join/Leave Chat Rooms
View and Request to Join Projects
Request approved
View Scripts
Request Android/iOS virtual Node for testing script
Execute script and report bug
Stop Virtual Device
Update/Delete Bug Status Report
Project Manager
Update Personal Information
Create/ Delete/ Add and remove members to Chat Room
Direct Message
Post notifications for testers
Post/Update/Delete and view Projects
Approve and Reject Tester Requests
View Bug Status Reports of assigned testers
Monitor Tester Performance
Make Payments to Testers
System Admin
Monitor user accounts and resource allocations
Block users

Scalability and Load Balancing

Used AWS EC2 auto scaling and classic load balancing algorithms

Auto scaling Condition: When throughput(no of requests served per second) is more than 8000, spin up a new instance(scale out) and shut off a machine when the throughput value is drops to 4000 marks(scale in)-values determined from JMeter load testing of our application

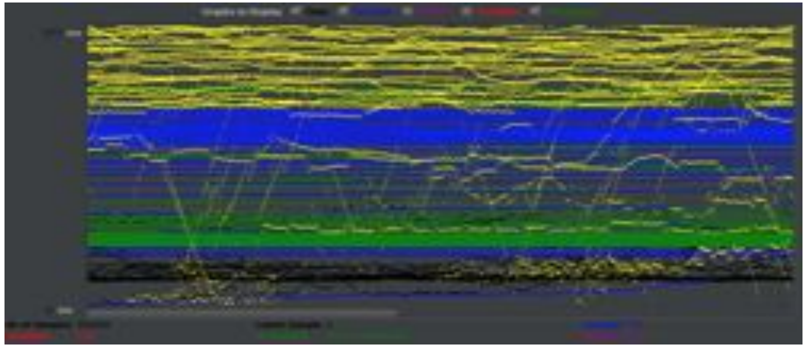
Load Balancing based on autoscaling-direct request to new instance



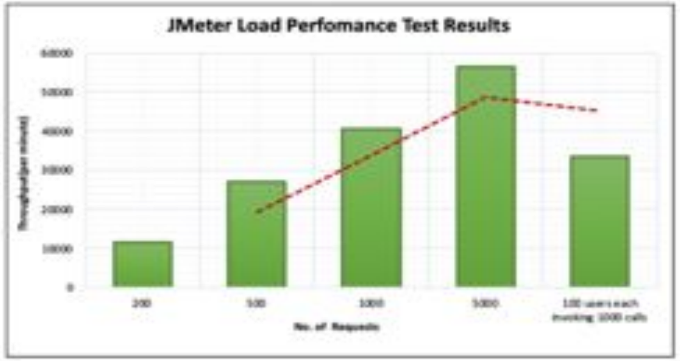
JMeter Load/Stress Testing Results:

JMeter Throughput graph for 1000 calls

JMeter Throughput graph for 5000 calls



JMeter Throughput graph for 100000 calls



JMeter Throughput Load Performance Test Results

Extra Credit Activities:

1)Google Maps API Integration to let the manager view testers based on their location

2)PayPal API Integration to enable transactions

3)Performance Testing of the Application using JMeter

Questions: