 

K SANTHAMURTHY

# PROJECT MANAGER

E-mail- Optimum Info System

Optimum Info System

Mobile- 90030 99941

**Certified Project Management Professional** (**PMP**)®,, **PMI Agile Certified Practitioner (PMI-ACP)® and TOGAF® 9 Combined Part 1 and Part 2 Certified**

**CAREER OBJECTIVE :**

**10.4** years of extensive working experience in Information Technology in Web Based Applications as Senior Consultant, having expertise in Project Management, Product Engineering, Quality Engineering and Quality Management. And now I am looking for challenging project manager position where I can use my skills and knowledge for software product development and release.

**FAMILIAR WITH TEAM RESPONSIBILITIES :**

**Product Owner responsibilities before sprint, during sprint and at the end of the sprint :**

* Before sprint Product owner should define product vision, roadmap, strategy and set sprint goal with team.
* During the sprint Product Owner should take responsibility for prioritization of product backlog.
* During the sprint Product owner should be regularly available for discussion / clarification.
* At the end of the sprint Product owner has the final authority over the product/stories.
* At the end of the sprint Product owner should inspect the product against the story and DoD ( Definition Of Done ) and should close the story every sprint.

# Design/ Dev SPOC responsibilities during the sprint :

Create number of sub task relevant to the user stories (mandatory sub task as below)

* Features covered as part of the user story.
* To upload the design document / wireframe (if not uploaded as part of story grooming).
* To review the design.
* To Perform Code Review.
* To Perform Unit Testing.
* To explicitly test the browser compatibility / performance aspects / security aspects.
* Create Issues / Enhancement request to the framework team if there any dependency and assign those JIRA to the framework team (link those JIRAs against user stories).

# Testing SPOC responsibilities during the sprint :

* Create a sub task with respect to each user story, to prepare the test cases (ST Cases and SIT cases) and get them endorsed by business solution team.
* Create a sub task with respect to each user story, to perform the ST once the UT results are shared.
* Create sub task to ensure non-functional requirements are coat (load test/stress test /and all other performance aspects).

# Scrum Master responsibilities during the sprint :

* Assign the user story JIRAs to the respective member by mentioning the start date & end date for each item.
* Track with the team and help delivering the committed items and facilitate if there is any impediment found.
* Facilitate Sprint planning meetings, daily stand ups, retro and walk through/demos for each sprint and most importantly removes impediments and helps with collaborating with interfacing teams where interface PM does not exist.

# DB SPOC responsibilities during the sprint:

* Create a sub task with respect to each user story, to cater all DB related activities.

# Scrum Team (All) when closing the sprint :

* Generate report, identify the backlog, plan the upcoming sprint and identify the area for improvement.

 

# DEFINITION OF DONE ( DOD )

**SPRINT 0 READINESS**

* User stories Jira links.
* ID ( Indonesia ) fast confluence link.
* ID ( Indonesia ) DB details (cloud).
* Model country VN (Vietnam) DB details (cloud).
* DevOps resource version.
* Services need to deploy versions.
* Sample payloads for the respective services for both model country VN (Vietnam) and ID ( Indonesia ).
* SIT service URL’s for the respective services for both model country VN (Vietnam) and ID ( Indonesia ).
* OCP console details for both model country VN

(Vietnam) and ID ( Indonesia ).

* EKS namespace and environment details.
* External interface system details( Kong External API Gateway / STP Adaptor / AVRO Stream / Solace / Kafka ).
* Connectivity details (Kafka / Solace / TCP IP connect).
* Technical data flow impacted services need to validate.
* Technical data flow impacted tables need to validate.
* Sample test case documents for model country ID ( Indonesia ).
* User story should be clearly defined with description and with acceptance criteria.
* User story should have been groomed well before starting the sprint.
* Story points should be assigned for all user stories as part of estimation, planning exercise.
* User story should be developed and tested, and it meets the acceptance criteria BDD format.
* Unit testing should get completed and unit test results should be attached for evidence.
* All code changes should be checked in and integrated and deployed in test environment and

test results should meet the defined acceptance criteria.

* Test preparation, automation scripting and execution should be completed successfully, and test results are to be attached for evidence.
* All open defects on stories should be closed or agreed with PO defer to subsequent sprint.
* User story status should be updated to closed status in Jira by product owner when above points are satisfied.
* For highly interdependent functions, there should be no associated user stories kept undelivered /open/pending in the same sprint/release.
* Sprint demo/review of sprint outcome discussion with PO should be completed.
* All epics/user stories in scope for the release- detailed requirements added in Jira or equivalent tools.
* Definition of done is met for all the user stories in scope of the release.
* Test execution complete (functional and non- functional) including regression, UAT and end to end testing and test evidence retained for review as per the standards defined in test strategy.

 

# MODERNISATION ACCOMPLISHED PAVING THE WAY TO MICROSERVICES

**REAL TIME DASHBOARDS ( CASH DATA LAKE ) – SCPAY CANONICAL PAYLOAD PUBLISHED IN REAL TIME**

**Support team :**

Instant alerts.

Event based monitoring System health check.

**Business analytics :**

Data analytics and artificial intelligence will provide real-time insights into client's transactions

Product volume analysis Amount based analysis

**Customers :**

Real-time payment status ..End to end tracking.

**Operations team :**

Real-time payment status .Event based monitoring Key-client monitoring

**Others:**

SCPAY events published to Cash ODS (Cash Data lake)

Header details exposed with various fields ..Audit events published

Product Information. Transaction & Event Status

Service & Sub-service Names, Status Exception Information

**Cloud Ready :**

With all technical changes, Atlas is already in AWS cloud.

Can be ported to any cloud. Cl/CD and DevOps implemented

# Support volume growth :

Enhancements done to handle high volume and  high throughput.

Improve response time.

Tested to scale up to 5000 TPS.

# Digital banking enabled :

Restful APIs for easy integration  300+ API Methods are ready

Real Time Event Framework for instant notification

300+ Events being developed

# Enhanced user experience :

Rich User Interface using React JS Chrome or IE compatible

Adopted standard UX Studio model Removed Applet Obsolescence


# Highly flexible :

Any database compliant.

Will work in PostgreSQL or DB2. Removed tight dependency on DB2. Can work with WAS or JBOSS.


# Safe and secure:

Implemented integration controls like TLS 12 ICS solutions like SSO, 2FA

Encryption to ensure access control and data security

 

s

**TYPICAL APPLICATION ONBOARDING ONTO CLOUD:**









**POC -** Proof Of Concept - 6 weeks

**Solution conclusion and approval** - 6 weeks

**Application readiness** - It may vary

**System Testing ( ST) / System Integration Testing ( SIT )** - 5 Weeks













**User Acceptance Testing ( UAT )** - 5 Weeks

**Performance Testing** - 2 Weeks **Application Vulnerability Testing** - 2 Weeks **Regression Testing** - 2 Weeks

**DRH 1 and 2** - 2 Weeks

**Operational Acceptance Testing ( OAT )** - 2 Weeks

* **System Acceptance Testing ( SAT )** - 2 Weeks
* **Setup Production Infrastructure** - It may vary

**MCAP - Materiality Cloud Assessment Profile - Which includes GRC, BRC and PRA approval :**











**Stage 1** Approval - 2 Months

**Stage 2** Approval - 2 Months

**Stage 3** Approval - 2 Months

**GRC - GROUP RISK COMMITTEE**

**BRC - BOARD RISK COMMITTEE**

**DIFFERENT CONTROL FAMILIAR WITH:**















Business Requirements (Functional)

Business Requirements (Non-Functional) Capacity Planning and Monitoring .

Code Quality Review.

IT Service Continuity Management.

Operational Level Agreement (OLA).

High Level ( HL ) System Architecture.





















Operations Manual.

Release Performance and stability testing. Service Level Agreement (SLA).

SIA: Security Control.

Technical Deploy ability confirmed release. Test Closure Report.

Test Evidence. Test Strategy.

Traceability Report.

Unit Testing.

sss

**BREAKING DOWN LEAP STAGES AND LEAP PROCESS :**

* **Stage 1** - **Development -** 11 Controls, 2 Approvers - Cloud Governance and BBMO - 2 Weeks
* **Stage 2 - Staging** - 24 Controls, 6 Approvers - Architecture, Compliance, CISRO, CSS, Resilience, BBMO - 6 Weeks
* **Stage 3 - Production** - 11 Controls, 5 Approvers - Architecture, Compliance, CISRO, CSS, Resilience - 3 Weeks
* **Country Regulator Approval** - 16 Weeks ( It may vary )

**MCAP PROCESS:**

* **MCAP Studio** - 6 Weeks
* **Price Waterhouse Coopers ( PWC ) External Review** - 5 Weeks
* **Senior Management Approval** - **Group CIO and Group CRO** - **Group Risk Committee ( GRC ) - Board Risk Committee ( BRC )** - 1 Week
* **UK Regulator** - 4 Weeks
* **PRA Approval** - 4 Weeks

# CORE COMPETENCIES

* + Extensive working experience in Information Technology as a Senior Consultant in Web Based Applications using Selenium Web Driver.
	+ Extensive experience in **Banking, Payment Platform** and **E-Commerce** Domain.
	+ Expertise in creating test scripts in selenium Web Driver by using java for automating the web-based application.
	+ Having good experience in automation frameworks such as **Cucumber BDD**, **Test NG**, **Junit**, **POM** and **Data Driven** framework.
	+ Expertise in developing **BDD** scripts with Cucumber.
	+ Experience in Build Automation tools like Maven and Continuous Integration Tools like

# Jenkins.

* + Having Experience in **web service** testing with **REST Assured** framework.
	+ Extensive experience in designing Test scenarios, Test Scripts and Test reports of automated tests.
	+ Hands on experience in tracking and reporting of defects using Defect Tracking tool –

# JIRA

* + Expertise in developing **BDD** scripts using Feature Files by **Cucumber**.
	+ Maintaining the checkout and check in the source code from **GIT and GIT Hub.**
	+ Good experience in writing **X-paths and CSS selectors** to identify web elements.
	+ Good Experience in Agile and waterfall methodology.
	+ Involved in **ceremonies** like Daily Scrum call, Sprint planning, Sprint review and Retrospective.
	+ Having experience in **Jenkins** Tool for **Continuous Integration.**
	+ Good Experience in SQL Queries

# TECHNICAL SKILLS

**Language :** Java

**Automation Tool :** Selenium

**Framework :** Cucumber, Junit, Test NG, POM, REST Assured API, Genie, Spring Boot

**DBMS** : SQL,JDBC, PostgreSQL

**Tools and Utilities** : GIT, Maven, Jenkins, MS-Office, Outlook, Bitbucket, Kibana, Swagger, Anaconda

# Project Management Tool : JIRA, Confluence

**Domain Knowledge** : Banking, Payment platform and E-commerce

# PROFESSIONAL EXPERIENCE

From May 2021To August 2022 in **Optimum Info System, Perungudi, Chennai**

From Apr 2012To Feb 2021 in **Energy Soft Solutions, Trichy PROJECTS HANDLED**

# Project # 1:Payment Platform

* + - Project title : Simplified Case Management ( SCM ) and Customer Due

Diligence ( CDD )

* + - Domain : Payment Platform
		- Role : QA Engineer – Backend -Java
		- Architecture : Client Tier – Delivery Tier – Front Tier – Mid Tier – Data Tier
		- Methodology : Agile
		- Platform Name : PAYPAL

# Description:

* Consumer interacts with a merchant website and initiates the checkout process.
* Merchant establishes a checkout session context and redirects to PayPal’s checkout application.
* If necessary, user will be prompted to authenticate via a redirect to Unified Login (UL ). UL interacts with Auth Service to identify and validate users. Risk information is collected and consulted to prevent fraud.
* PayPal’s Checkout service orchestrates user experience and business logic to establish a user session and invoke risk and payment processing.
* Payment service orchestrates payment activities, including compliance and risk checks, funding source selection, FX conversion, and recording of the transaction itself.
* FX rates are retrieved and cached twice daily and are managed by Treasury. Rates are disclosed by the checkout application and recorded as part of the application.
* Payment fulfillment initiates funds movement from cards into

PayPal via Payment Switch, which interfaces with processors using ISO8583 and proprietary gateways. Bank funded transactions are authorized by PayPal risk rules.

* Subject to risk rules, funds are made available to the merchant’s account immediately. Back-end settlement and reconciliation in PayPal’s financial system occurs at least daily.

# PAYPAL DOMAIN MODEL

PayPal's 'Domains" are comprised of the following functional groupings:

* 1. **Products** these are the primary customer facing capabilities of the PayPal system, they typically include user experiences in addition to services
	2. **Supporting Services** these include our Customer Support organization and the tools they use as well as the Marketing content to educate and promote our capabilities to customers.
	3. **Platform Services** these are the primary business functions of PayPal dealing with Identity, Wallet, Payments, Risk and Compliance.
	4. **Infrastructure and Security Domains** - these are common capabilities used across by higher level services
		+ The domain model depicts the domains as groups of Capabilities. So for instance the "merchant" domain has several capabilities including checkout and invoices
		+ In a few cases two domains may have a capability with the same name, in that case the Capabilities should be considered distinct from one another
		+ After the diagram we have a written description of each overall domain and the capabilities illustrated. Where available, hyperlinks to confluence spaces are included with each description. Note: the hyperlinks are considered non normative.

 

# Project#2: Banking

* Project title : SCPay
* Domain : Banking – Corporate, Retail, Credit Card and Value Added

Services (VAS)

* Role : Analyst, Platform Services – Cash Management
* Architecture : Event Driven Micro Services Based Architecture
* Methodology : Agile
* Bank : Bank Indonesia ( BI ) / Standard Chartered Bank ( SCB )

**ABOUT BANK INDONESIA ( BI )**

Bank Indonesia is the central bank of the Republic of Indonesia. It replaced in 1953 the Bank of Java, which had been created in 1828 to serve the financial needs of the Dutch East Indies.

**BUSINESS REQUIREMENT DEFINITION ( BRD )**

* **There are 4 main items to be developed for BI ( BANK INDONESIA ) FAST :**
	1. **Proxy / Alias Management**

BI – FAST will be equipped with an addressing service feature. Addressing service administers the alias data ( proxy address ) for information of account name, account number, bank name, bank code, or recipient participant. This feature provided is to increase the practicability in doing Individual Credit Transfer ( CT ) transaction and Request for Payment (RFP), thus no need to fill in data of the account name, account number, bank name, bank code, or recipient participant. The Addressing service

( Proxy or Alias ) is a central infrastructure system accessible to entitled Participants, exposing APIs for Alias resolution and Alias Maintenance ( modify, suspension, registration, deregistration, activate, porting, and inquiry).

* 1. **BI FAST Payment Transaction**

High Level Transaction Flow for BI FAST Payment

1. **Mechanism for Information Query:**
	1. Remitter will initiate transaction through the available bank channels ( i.e. internet banking, mobile banking, etc.).
	2. Remitting Bank sends inquiry to beneficiary Bank through BI – FAST infrastructure to get the information on Beneficiary.
	3. Beneficiary Bank will validate the account number received and send the Beneficiary information to Remitting Bank. Remitting Bank will pass on the information to Remitter for validation.
	4. **Mechanism for Transaction :**
	5. Upon validation of Beneficiary account information, remitter sends payment instruction to remitter Bank.
	6. Remitter Bank sends the payment instruction to BI- FAST to be further sent to Beneficiary Bank.
	7. Beneficiary Bank credit the funds to Beneficiary on real – time basis.
	8. Beneficiary Bank notification to BI-FAST that funds have been credited to Beneficiary.
	9. BI- Fast will pass on the credit notification to Remitting Bank.
	10. Remitting Bank will debit the Remitter and provide notification


# MICRO SERVICES AND ITS CAPABILITIES

1. **Pre-Processor:**
	* Receives single and bulk payment from channels.
	* Mandatory validation and JSON transformation.
	* Payment type derivation.
	* Route payments to SCPAY or BAU System.

# Router :

* + Rule based prioritization lane engine.
	+ Segregation of high / low value payments.
	+ Segregation of retail and corporate lane.

# Adaptor:

* + Converts messages to standard NPP-JSON.
	+ Split batches into individual transactions.

# Processor:

* + Holiday Cut-off management.
	+ Duplicate check.
	+ Payment enrichment.
	+ Customized narration framework.
	+ Bank code validation.
	+ Account validation.
	+ Amount derivation.
	+ Balance and risk check.
	+ Reg / Compliance check.
	+ Threshold limit verification.
	+ Purpose of payment.
	+ IBAN calculator / validator.
	+ OBO Derivation.
	+ Virtual account.
	+ Auto returns.
	+ Value date derivation.
	+ Beneficiary name verification

# Risk and Compliance :

* + Transaction and document screening.
	+ Sanctions screening ( SaaS )

# Charges:

* + Flexible charge and billing.
	+ Charges handling / calculation ( GPBS )

# FX Booking :

* + S2BX / SCALE / SHAPE Auto FX Booking

# Accounting Generation :

* + Account expression generation.
	+ Account expression Cache.

# Accounting :

* + Accounting entries ( Core Banking ).
	+ Account posting reversal.

# Clearing:

* + Clearing message generation.

# Dispatcher :

* + Clearing message dispatcher.

# Notification :

* + Notification to the channels.
	+ Advices to the customer.

# Returns:

* + Auto returns.
	+ ALM

# Reference Data:

* + RDM - Bank / Branch / Currency / Country.
	+ CADM - Client preferential data.
	+ Virtual accounting data.
	+ Business rules.
	+ System parameter.
	+ Static data PI specific.

# Auxiliary services:

* + CARDS 400 / CCMS - Credit card validation.
	+ S2BR - Receivables Management System.
	+ BNCPR - Payment volume and revenue feed.
	+ GPBS - Client pricing system
	+ MDIS - Customer advices, FAX, Email and SMS.
	+ AML DETICA - Anti money laundering system.
	+ RDM - Bank codes, Currency, Country



**FUNCTIONAL FLOW**

* Transactions received to SCPAY from different channels are passed through various Internal Micro service as per the payment lifecycle.
* When

there

is

any

Exception /

Rejection/Approval / Successful processing of payment based on each of the validations performed in Individual micro services, those will be communicated to channels via notification service.

* Other services within SCPAY shall write in to the respective topics with the sequence of events using the data event store microservice.
* Based on the configuration set, event listener with in Notification (response) micro service shall pick the event (this is based on the configuration that only the events that are eligible for channels notification will be notified) and notify the status codes, description back to channels.

**SOLACE AND KAFKA ECOSYSTEM:**







These two are the messaging systems for SCPay platform.

They both lies in on-prem as well as in Cloud

Solace is basically used for asynchronous communication.

external



















Pre-processor has no Kafka only solace.

On on-prem, Kafka is used only for capturing log metrics. Solace is used for internal communication with the micro services too.

But on cloud, Kafka is used for internal micro service communication and log metrics.

Both have clustered environment

Both helps in achieving event persistency. Both have different properties for getting connectivity.

Instrumentations are done for both monitoring.

Kafka cluster lies in EKS. Kafka Topic Taxonomy

**BUSINESS CASE**

SCB customers wanted to view purpose of transactions in a customized manner to satisfy below requirements:

1. To perform auto reconciliation of transactions by using feeds from channels.
2. These feeds are sent to clients not only by SCB but by other banks as well hence they do not want to configure reconciliation on feeds bank wise, instead have a standardized or uniform auto reconciliation at their end which requires customization at SCB. Statements that are generated by the clients will populate only those details that are important to them which may require further investigation if need to.

**END TO END FLOW WALK THROUGH (INWARD, FAST/NON-FAST ,SINGLE, REQUEST AND OUTWARD)**

* The Payment initiated at the Initiation phase from channel which moved to Router stage where the message routing will happen based on predefined rules of priority and formats.
* The message will move to the next phase i.e. Adapter where it is gets converted from external format (JSON) to internal canonical JSON. Canonical JSON is common understandable language for all transactional participants.
* Then the message (in the internal JSON format) will move to the Processor phase where payment related validations and derivations will takes place.
* After successful processing at Processor stage, the payment will move to next stage, i.e. Accounting Generation which will create the accounting entries required for the transaction posting.
* The details transaction posting will then move to the Accounting (Settlement ) phase where the transaction will be sent to the core banking application(eBBS) for posting.
* Upon successful posting, the payment will move to the Clearing . The clearing phase will process the payload JSON and send the JSON to next stage.
* Dispatcher services get the internal payload and send the GCG specific payload JSON to clearing house for the final payment settlement.

#

**SCPAY OVERVIEW:**

•SCPay is a new payment platform which will replace existing legacy payment applications - STS, DotOpal, Hogan and GCG.

• It is build as a highly resilient single payments platform for the bank, which can scale exponentially and make SCB the fastest payment processing bank in the market.

• It is a system under SCB’s Cash Management business that caters for both individual and corporates.

• SCPay is a purpose built for multi-country and multi product, Global Payment Platform built on a set of atomic and independently deployable microservices.

• The transaction processing is driven by event bus/JSON in SCPay. The payment lands into the event bus and it moves from one stage to the other .The final data is persisted in Oracle database, to be used later for reporting and auditing purpose. The event of the action is based on the status of the payment at that stage, To handle replay scenarios, the internal JSON will be persisted into a separate database i.e Casandra DB.

• There are list of libraries that are built in SCPay to provide a standard way of using technologies like Kafka, Solace, Hazelcast and provide utilities for different functionalities like Messaging, Logging, Caching, Parsing Messages and Sequence Number generation known as SCPay Core Framework.

**SCPAY DESIGN PATTERNS AND ARCHITECTURE:**

Based on 12 factor design pattern as SCPay is a cloud native application.

* Strangler design pattern.
* Adapter design Pattern.
* API Gateway design Pattern.
* Certificate based security design.
* Tokenized security pattern.
* Choreography based Saga design pattern.
* Compensating Transaction pattern(Replay).
* Publisher/Subscriber pattern(solace).
* Retry.
* Throttling.
* Health Endpoint Monitoring.
* Caching

**EXISTING LEGACY SYSTEM AND THEIR TRANSACTION PROCESS:**

* The transaction process for the current legacy system systems i.e., STS, dot Opal, Hogan and GCG is database driven.

**KEY CHALLENGES:**

* + The payment first used to land into the main table and then into multiple sub tables while moving from one stage to other.
	+ Intermediary data like computation, derivations are also stored on database.
	+ These table details are referred from time to time to derive the final list of details for the payment. Multiple tables are maintained in STS , based on payment/product type and region wise.

**NEW THOUGHT:**

* So new thought came up to SCB to use DB very less and that's why they started migrating the legacy system.
* Dot Opal is migrating in to SCPay.

# Project#3 :Banking

* + - Project title : Standard bank of South Africa
		- Domain : Banking
		- Role : Test Engineer
		- Methodology : Agile

# Description:

The Standard Bank of South Africa Limited is a South African financial services group and it is Africa's biggest lender by assets. The company's corporate headquarters, Standard Bank Centre, is situated in Simmonds Street, Johannesburg. The bank now known as Standard Bank was formed in 1862 as a South African subsidiary of the British overseas bank Standard Bank, under the name The Standard Bank of South Africa.

# Project# 4: Travel Domain

* + - * Project title : Origin air
			* Domain : Travel Domain
			* Role : Senior Test Engineer
			* Methodology : Agile

# Description:

Origin air is an airline based in Nelson, New Zealand, that operates domestic flights in New Zealand. It was founded in 2015 by local businessman Robert Inglis, who had previously founded Air Nelson and Origin Pacific Airways. Origin air commenced operations on 12 August 2015 with flights between Nelson and Palmerston North; it started flights between Nelson and Wellington the following month. The airline began its operations with one British Aerospace 31 aircraft, and now operates a fleet of four aircraft, following the addition of another 2 Jetstream 32 aircraft.

# Project#5 : E-Commerce

* + - * Project title : Selfridge
			* Domain : E-commerce
			* Role : Senior Test Engineer
			* Methodology : Agile

# Description:

Selfridges, also known as Selfridges & Co., is a chain of high-end department stores in the United Kingdom that is operated by Canadian group Selfridges Retail Limited, part of the Selfridges Group of department stores. It was founded by Harry Gordon Selfridge in 1908.

# Project#6: E-Commerce

* + - * Project title : New egg
			* Domain : E-commerce
			* Role : Senior Test Engineer
			* Methodology : Agile

# Description:

Newegg Commerce, Inc. is an American online retailer of items including computer hardware and consumer electronics. It is based in City of Industry, California. It is majority- owned by Liaison Interactive, a multinational technology company. In 2004, Newegg established as a private-label reseller of computing and household products from many manufacturers. In 2005, Newegg.com was named one of the Internet's Top 10 retailers by Internet Retailer Magazine, with a 2004 sales revenue of just under $1 billion.

Newegg.com grew an additional 30% in 2005, bringing annual sales to approximately $1.3 billion.

# RESPONSIBILITIES :

* Analyzing the User Stories and planning the User Stories to be completed in the particular Sprint.
* Test manually and identify the defects.
* Read all the documents and understand what needs to be tested.
* Responsible for preparing test scenarios and Test Cases.
* Responsible for preparing REQUIREMENT TRACIBILITY MATRIX.
* Identify the Bugs and report the defects to development team through JIRA.
* Re-testing the Resolved defects.
* Regression Testing will be done to check the existing functionality must not be affected.
* Preparing the Test Summary Report.
* Preparing Weekly and monthly status report.
* Updating project manager regularly about the progress of testing activities.
* Suggested risk based approach for pages to be tested which reduced the effort for testing.
* Ensure timely delivery of the project.

# EDUCATION

**BE (ELECTRICAL AND ELECTRONICS ENGINEERING**) - from Park College of

Engineering and Technology, Coimbatore with 5.8 CGPA in 2005

# DECLARATION

I hereby declare the above mentioned details are true and correct to the best of my knowledge.

# Date :

**Place** : Chennai **(K.SANTHAMURTHYs**