



**JARAMOGI OGINGA OGINGA UNIVERSITY OF SCIENCE AND
TECHNOLOGY
SCHOOL OF INFORMATICS AND INNOVATIVE SYSTEMS
DEPARTMENT OF COMPUTER SCIENCE & SOFTWARE ENGINEERING**

**A project Proposal Submitted to the School of Informatics and Innovative
Systems at Jaramogi Odinga Odinga University of Science and Technology**

DIGITAL VAUCHERS

By

KIPLAGAT KELVIN MUTAI

A Project submitted to the board of undergraduate studies in partial fulfillment of the requirement for the Award of a bachelor degree of science in information technology security and Audit of Jaramogi Odinga Odinga university of science and technology

November 2019

DECLARATION AND APPROVAL

STUDENT

I, the undersigned declare that this research project is my original work and that it has not been presented in any other university or institution for academic credit.

Signature

Date.....

KIPLAGAT KELVIN MUTAI

I132/0644/2016S

UNIVERSITY SUPERVISOR'S APPROVAL

This research project report has been submitted for examination with my approval as university supervisor.

Signature.....

Date.....

Dedication

I dedicate the entire work carried here in to my Mum Mrs. Emily sang for their inspiration all through my course. I would like to appreciate all my lecturers who instilled in me lifelong values and the desire for education. Finally, I would like to appreciate all of you who have

contributed to my success and have not been mentioned above, remember that all your efforts are highly appreciated and you will never be forgotten for your stake in my life, God bless you all.

ACKNOWLEDGEMENT

We would like to acknowledge God Almighty for the strength, sound mind and provision during the period of study. Special thanks to our supervisor Dr. Ogara and Moderator Professor Rodrigues for their immense support, guidance and patience, and without their constructive criticism and advice this work would not have been

complete. We also register our gratitude to our parents and siblings for their continuous encouragement during this journey, my university Administration (Jaramogi University) for providing an enabling environment for us to complete the course and access the labs during weekends and at night. Thank you to all my friends who contributed to the completion of this academic document both directly and indirectly. They provided me with logistical and moral support that gave us every reason to work harder and ensure that this study becomes a success.

To all of you, God bless and increase you immensely.

ABBREVIATIONS & ACRONYMS

API: application programming interface

DFD: data flow diagrams

SDLC: system development lifecycle

TLC: transport layer security

ABSTRACT

As distribution models of information systems are moving to XaaS paradigms, microservices architectures are rapidly emerging, having the RESTful principles as the API model of choice. In this context, the term of API Economy is being used to describe the increasing

movement of the industries in order to take advantage of exposing their APIs as part of their service offering and expand its business model. Currently, the industry is adopting standard specifications such as OpenAPI to model the APIs in a standard way following the RESTful principles; this shift has supported the proliferation of API execution platforms (API Gateways) that allow the XaaS to optimize their costs. However, from a business point of view, modeling offering plans of those APIs is mainly done ad-hoc (or in a platform-dependent way) since no standard model has been proposed. This lack of standardization hinders the creation of API governance tools in order to provide and automate them

The purpose of this research was to design and implement an application programming interface for web-based digitalization of voucher. The study focuses on under 18 generation and those with no identification card like the foreigners and would like to purchase goods and services. Our target being students and the foreigner and also the local was based on the same capability of not owning the identification card. The application programming interface will be able to create an environment whereby there will be a generation of a voucher having a given set of code which are unique for both the client and merchandise. The client would deposit amount in the float account and immediately given the deposit code and a voucher. It will involve the float account, customer interface, server interface and internet service provider. At the merchandise section one will pick the goods and then make payment through the given unique voucher the amount will be deducted from the voucher and awaits the next purchase.

CHAPTER ONE

1.0 INTRODUCTION

While modern web APIs were officially born with Roy Fielding's dissertation Architectural Styles and the Design of Network-based Software Architectures in 2000. Web APIs first appeared in the wild with the introduction of Salesforce on February 7th, when the company officially launched its API at the IDG Demo 2000.

Application programming interface (API) is accessible using Representational State Transfer (REST) and provides methods for accessing public and private data and terminology extraction methods by using agile development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams.

The api, will act has the integrating link between a third-party platform for applications to access its features and services. In other words, an api, will delivers your request for information to the operating system and then delivers its response back to you. we decide to apply api basing on the fact that API specifies how software components should interact. Additionally, APIs are used when programming graphical user interface (GUI) components.

1.1 BACKGROUND OF THE STUDY

The main reason that APIs matter so much in modern markets is that they allow for faster innovation. Barriers to change are reduced and more people can contribute to an organization's success. They offer two-fold benefits: the company can create better products while standing out from the competition. Digitalizing of voucher tickets depends much on api key generation with help of a push key. Api key in our voucher will be used by client's customers in purchasing good and service using random generation of keys.

1.2 PROBLEM STATEMENT

In our country Kenya there are rules that govern the rights of having the identification cards and is of more challenging to the under 18 and the foreigners also few locals who visit our country more often. With digitalization of voucher our web-based system will create an environment where both the foreigner and the under 18 will be able to purchase good and service without the need of them having a phone or a simcard. Digital voucher with the help of php coding and java environment will help create a system that let say the parent to the student will send money to the voucher card of a given amount and the student being the owner of a given ticket will be given the api key which will be use it to purchase good and service and more so the foreigners. It will ease the foreigner ability of just purchasing the good and a well-advanced service without following all procedures.

The following problems are to be solved.

i)under 18 youths will be able to access digital voucher without having identification cards.

ii) Foreigner who visits our country will be able to purchase and have services offered without much strain of looking for a bank to change their currencies or identification procedures.

iii) local with lost identification card are able to interact with our system till when the id card is processed.

1.3 OBJECTIVES

1.3.1 MAIN OBJECTIVE

To develop a digital voucher for purchase of good and services. The main objective is creating a web-based system that will use api to achieve its goals.it will be accessed anywhere as provided network is available.

1.3.2 SPECIFIC OBJECTIVES

i) To develop a system that will be competitive and that can integrate with the m-pesa, PayPal and pioneer.

ii)To develop a system that will store all the client records and monitoring how they interact with the merchandise sector.

iii) To develop a system that will enhance secure and allow flow of purchase of good and services.

1.4 JUSTIFICATION AND SIGNIFICANCE OF THE SYSTEM

Upon the completion of design of the system, the various users of the system will benefit from the system in the following ways:

1. Improvement of security since the system unsure that clients are well secured through TLS is a standard that keeps an internet connection private and checks that the data sent between two systems (a server and a server, or a server and a client) is encrypted and unmodified. rough

2. The system will eliminate a situation norm that one need to have identification card to access a given service.

1.5 ASSUMPTION

1. The system has the assumption that the system users are connected to internet.

2. All system users have the knowledge of using smart phone, tablets, PDA and computer.

3. There is network connection wherever the system users are.

1.6 SCOPE OF THE STUDY

Generally, the idea digital voucher cuts across the Merchant Centers or with the API client's endpoint for a single project when creating an API client.

Involves the under 18 youths some locals and foreigner, also the client and the merchant where service will be offered. The system can accommodate more than 70 -90, 000 both clients and merchants.

CHAPTER TWO

2.0 LITERATURE REVIEW

A number of analyses of web services in the industry and, especially, of RESTful APIs, have been presented. They usually focus on characteristics inherent to the API design. This work presents a new research direction by developing a system-atic study of RESTful APIs focusing on how providers deal with non-functional properties in plans by establishing limitations, such as rates and quotas. The point out that hardly any of the services claiming to be RESTful is truly RESTful. analyzes a dataset which comprises 45 Web APIs in total, primarily chosen from Programmable Web directory, and provide conclusions about common de-scription forms, output types, usage of API parameters, invocation support, the level of reusability, API granularity and authentication details.

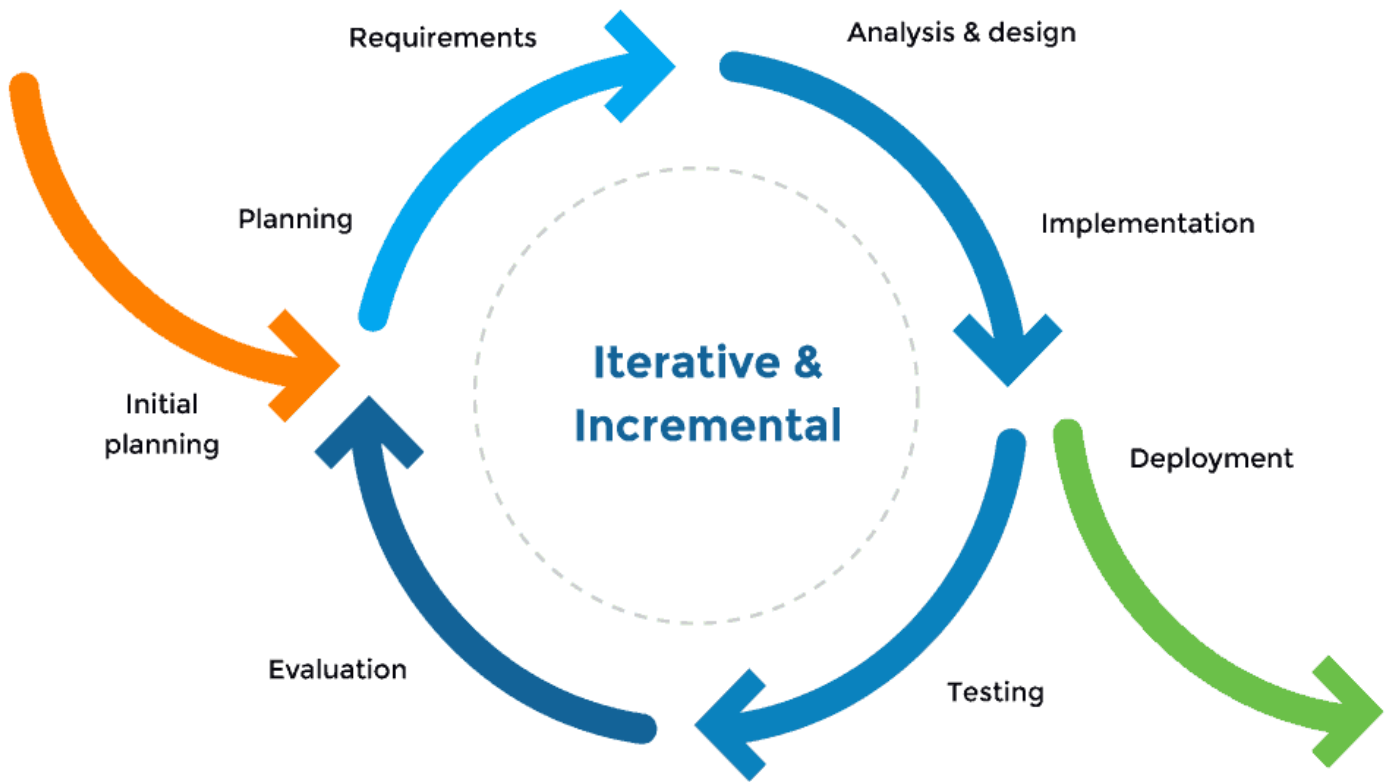
CHAPTER THREE

3.0 METHODOLOGY

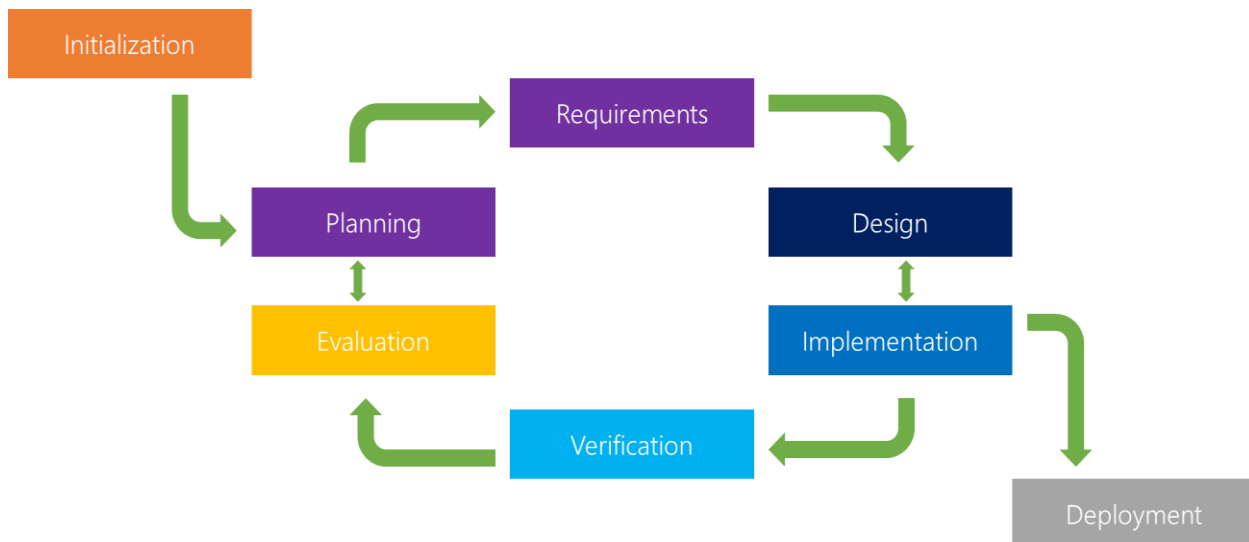
3.1 Iterative Methodology

This methodology focuses on an initial, simplified implementation, which then progressively gains more complexity and a broader feature set until the final system is complete. Incremental alterations are made during the design and implementation of each new iteration.

This model is best thought of as a cyclical process. A handful of stages are repeated over and over (after the initial planning phase) with each completion of the cycle incrementally improving and iterating on the software. This makes enhancements be quickly recognized and implemented throughout each iteration, and hence allowing the next iteration to be at least marginally better than the last.



Iterative Model 1



Iterative Model 2

- **Planning & Requirements:** The first step is to go through an initial planning stage to map out the specification documents, establish software or hardware requirements, and generally prepare for the upcoming stages of the cycle.
- **Analysis & Design:** this phase is performed to nail down the appropriate business logic, database models, and the like that will be required at this stage in the project soon after the planning phase. The design phase also occurs here, establishing any technical requirements (languages, data layers, services, etc.) that will be utilized in order to meet the needs of the analysis stage.
- **Implementation:** The actual implementation and coding process can now begin soon after the planning and analysis phases. All planning, specification, and design docs up to this point are coded and implemented into this initial iteration of the project.
- **Testing:** after building iteration has been fully coded and implemented, the next step is to go through a series of testing procedures to point out (identify) and locate any potential bugs or issues that have come up since the last iteration.
- **Evaluation:** Once all prior stages have been completed, it is time for a thorough evaluation of development up to this stage. Everyone is now involved, including clients or other outside parties, to examine where the project is at, where it needs to be, what can or should change, and so on. These are then documented and taken as requirements for the iterative process.

Advantages of the Iterative Model

- **Inherent Versioning:** Iterative model makes this even easier to version by ensuring that newer iterations are incrementally improved versions of previous iterations. Moreover, in the event that a new iteration fundamentally breaks a system in a catastrophic manner, a previous iteration can quickly and easily be implemented or “rolled back,” with minimal losses; a particular boon for post-release maintenance or web applications.
- **Rapid Turnaround:** In the iterative process, each stage can effectively be slimmed down into smaller and smaller time frames; whatever is necessary to suit the needs of the project or organization. While the initial run through of all stages may take some time, each subsequent iteration will be faster and faster, lending itself to that agile moniker so very well, and allowing the life cycle of each new iteration to be trimmed down to a matter of days or even hours in some cases.
- **Suited for Agile Organizations:** The iterative model really starts to shine when it is in the hands of a smaller, more agile team. Particularly when combined with the power of modern version control systems, a full “iteration process” can effectively be performed by a number of individual team members, from planning and design through to implementation and testing, with little to no need for outside feedback or assistance.
- **Easy Adaptability:** The ability to rapidly adapt to the ever-changing needs of both the project or the whims of the client. Even fundamental changes to the underlying code structure or implementations (such as a new database system or service implementation) can typically be made within a minimal time frame and at a reasonable cost, because any detrimental changes can be recognized and reverted within a short time frame back to a previous iteration.

Disadvantages of the Iterative Model

- **Costly Late-Stage Issues:** in some cases; due to the minimal initial planning before coding and implementation begin, when utilizing an iterative model, it is possible that an unforeseen issue in design or underlying system architecture will arise late into the project. Resolving this could have potentially devastating effects on the time frame and costs of the project as a whole, requiring a great deal of future iterations just to resolve one issue.
- **Increased Pressure on User Engagement:** Unlike other old methodologies, which emphasizes nearly all user/client engagement within the initial stages of the project during a brief crunch time period, the iterative model often requires user engagement throughout the entirety of the process. This is sometimes an unfortunate obligation, since each new iteration will likely require testing and feedback from users in order to properly evaluate any necessary changes.
- **Feature Creep:** Iterative model requires user feedback throughout the process, this means it also inherently the project may be subject to undesired feature creep, whereby users experience the changes in each iteration, and are inclined to constantly put forth new requests for additional features to be added to future versions.

Error are supposed to be handled well and it takes time to gather and discern the errors that matter most.

CHAPTER FOUR

4.0 REQUIREMENT ANALYSIS

CHAPTER FIVE

5.0 SYSTEM DESIGN

CHAPTER SIX

6.0 IMPLEMENTATION

CHAPTER SEVEN

7.0 TESTING AND EVALUATION

CHAPTER EIGHT

8.0 DISCUSSION AND CONCLUSION

REFERENCE