Justification document

**Senior Software Engineer – Safaricom Garage**

Version 1.0

Contents

1. [Revision History 3](#_Toc489425542)
2. [Definitions, Acronyms and Abbreviations 3](#_Toc489425543)
3. [Document Reference 3](#_Toc489425544)
4. [What’s New in this Version 4](#_Toc489425545)
5. [Introduction 4](#_Toc489425546)
6. [Choice of System 4](#_Toc489425547)
7. [System overview 5](#_Toc489425548)

[Connectivity 6](#_Toc489425549)

[Oracle Java JDK version 6](#_Toc489425550)

[OS User 6](#_Toc489425551)

1. [Deployment Activities 7](#_Toc489425552)

[Java Clients 7](#_Toc489425553)

[Step 00 – creating a workbench directory 7](#_Toc489425554)

[Step 01 – copying files 7](#_Toc489425555)

[Step 02 – Configuring Java Keystore 7](#_Toc489425556)

[Cron configuration 8](#_Toc489425557)

[Database 9](#_Toc489425558)

1. [Rollback plan 9](#_Toc489425559)
2. [Deployment Impact 9](#_Toc489425560)
3. [Artifacts 10](#_Toc489425561)

[Database Details 10](#_Toc489425562)

[MERCHANT SETTLEMENT PRODUCTION DB 10](#_Toc489425563)

[Deployment Files 10](#_Toc489425564)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Revised By |
| 1st August, 2017 | 1.0 | 1st Draft Version | Steve Mutungi |
|  |  |  |  |
|  |  |  |  |

# Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| Abbreviation |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Document Reference

|  |  |
| --- | --- |
| **Document Title** | **Author(s)** |
|  |  |
|  |  |
|  |  |

# What’s New in this Version

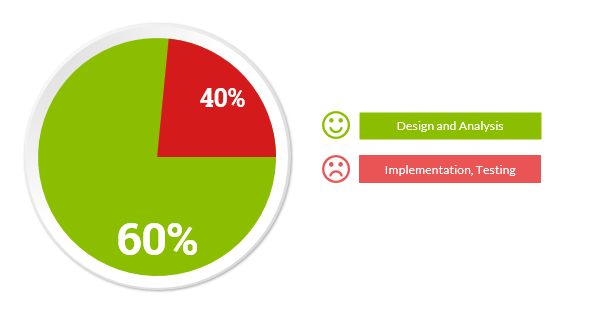
|  |
| --- |
| Notes: |
| * Initial justification document for choice of system, the rationale behind it and the delivery approach. |

# Introduction

This document highlights the delivery approach, choice of system, rationale for the decisions made given the constraints etc.

# Effort Used and Rationale behind

Most of the effort has been spent doing design, 60%. 40% has been used developing/implementing the system. This is based on industry experience in getting it right the first time.



As a senior software Engineer, this is 60:40 percentages still applies. Software engineers can develop the 40% as long as the design is clear.

# Choice of System

With the current political agenda on top of every Kenyan mind, I have chosen to develop and showcase IEBC system.

In any successful enterprise, we must consider the 3 items:

* People
* Systems
* Processes

This justification focuses mostly on systems aspect as the rest are already taken care of.

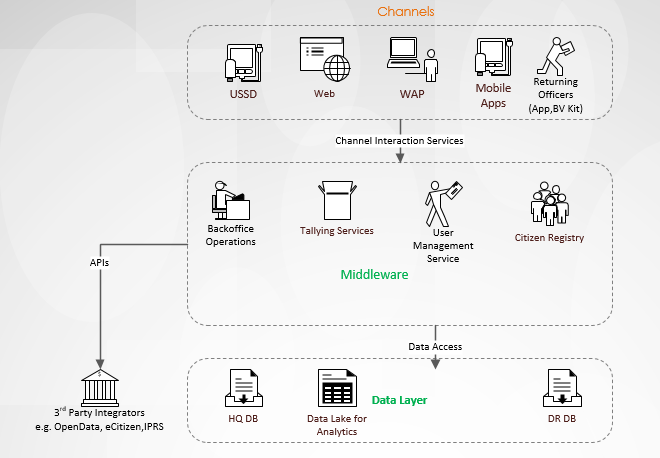
For any growing and maturing democracy, having a system that is robust, scalable, with omni-channel is preferred.

The IEBC system will utilize several APIs.

|  |  |  |
| --- | --- | --- |
| API Name | API Source | Comments |
| Authentication | Use of Kenyan government OneID for eCitizen | This uses Auth2.0 |
| Publishing IEBC results for different categories e.g. Presidential, governors, MP, Senators | OpenData system by Kenyan Govt |  |
| Facebook and Twitter posts | By Facebook and Twitter(OpenAPIs) |  |
| Google Maps | Gets your current location and the nearest point of interest – your voting polling station |  |
| Uber API | For estimating fare to the nearest polling station. |  |
| Internal APIs(private APIs) | Used to get data from the field by returning officers |  |

# System overview

The figure below highlights the key components of the system.



|  |  |  |  |
| --- | --- | --- | --- |
| Component | Category | Justification | Comments |
| USSD | Channels | To be used by Citizens to confirm registration details.Acts as backup for returning officers. | Should be accessible via one short code for all the Telcos |
| WEB | Channels | This has several facets: Backend portal for administration functions 🡺 **Spring Java Web highly recommended due to security considerations.**  Citizen portal to query registration details, search for IEBC official results etc. 🡺 **Laravel framework recommended.** | Internal portal is only available internally (local network). Public portals are accessible via public interfaces e.g. internet or public IPs. |
| WAP | Channels |  | This is a responsive version of the Web. |
| Mobile Apps (iOS, Android) | Channels |  |  |
| Micro-services | Middleware | This uses microservices architecture that can be deployed on Docker to serve several functions e.g. tallying, authentication, external communication with other APIs e.g. IPRS etc. |  |
| Data and Analytics layer | This constitutes of both RDBMS and non-structured DBMS. Data lakes are used for data aggregation and visualization to establish and answer the WHY (goes beyond WHAT). |  |  |
|  |  |  |  |

# Delivery Package

The delivery package committed on GitHub includes the following:

|  |  |  |
| --- | --- | --- |
| Component | Details | Comments |
| Mobile App | Uses Google Map API to get your coordinates, fetches the nearest polling station |  |
| Uber API | To estimate fare to the polling station. |  |
| WEB/WAP | Has analytics and dashboards. |  |