**Software Requirement Specification**

**NAME OF SYSTEM**

**DATE**

Version

**Logo**

**Presented To:**

**Name (client or class)**

**Submitted By:**

Group names

**TABLE OF CONTENTS**

**REVISION HISTORY**

| **Date** | **Author** | **Distributed to** | **Version** | **Description** |
| --- | --- | --- | --- | --- |
| Mmm DD, YYYY | name | who | 0 | template |

**1 Introduction**

Brief explanation of project (from SDP)

Brief explanation of purpose and goal of this document

**1.2 Scope**

Context Diagram

Personas / Actors

Use Case Diagram

**2 Related Documents**

Software Development Plan reference

Requirements Traceability matrix reference

Other diagrams references

**3 Requirements**

For each Actor, list all the user stories:

As an Actor1 I want to…..

* Bulleted list

As an Actor2 I want to…..

* Bulleted list

**3.1 Punch List**

The following list are the items still to be resolved:

1. ……

**3.2 Use Case 1 (example: comes directly from SDP)**

Use use Case Description Template and complete for **every** use case. Some use cases from the SDP don’t need any more requirement detail and come over as is. See 3.3 if more detail is needed.

Also define the non-happy path flows for all use cases. Most go in “other detail requirements” section“, some will go directly in”flow of events"

**3.2.1 Use case description**

**Description:**

**Use Case**:

1. **Name**:
2. **Participating Actor(s)**:
3. **Entry**:
4. **Exit**:
5. **Flow of events**:
6. **Special Requirements**:

**3.2.2 user interface requirements**

**3.2.3 other details requirements**

**3.3 Use Case 2 (example: expands to more use cases)**

Define the next level of detail for larger use cases, breaking it down into its own use case diagram at the lower level.

**<new use case diagram>**

**3.3.1 new use case name 1**

**3.3.1.1 new use case description**

**3.3.1.2 new use case user interface requirements**

**3.3.1.3 new use case other details requirements**

**3.3.2 new use case name 2**

**3.3.2.1 new use case description**

**3.3.2.2 new use case user interface requirements**

**3.3.2.3 new use case other details requirements**

**3.4 Use Case N**

**4 Non Functional Requirements**

**4.1 Other Systems**

* **Database server:** to store users and advertisements information.
* **Payment methods:** the system provide three ways to pay for advertisement Vodafone wallet, Orange wallet and Etisalat wallet.

**4.2 Security**

* **Authentication:** Each user in the website has a unique username and a password, users are enforced to make strong password (longer than 8 digits including alphabets, numbers and special symbols like $, @, &, %) so that malicious attackers cannot easily guess them.
* **Data Encryption:** By Implementing HTTPS for all communication between the user's browser and your payment gateway server
* **SMS 2FA (SMS-based Two-Factor Authentication):** by sending a code in SMS to the phone number which the user entered its wallet and ask the user to enter this code in its specified place in the website

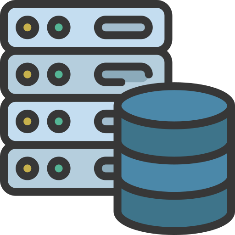
**4.3 Performance**

* **Availability:** Users must be able to access the website 365 days a year, 24 hours a day
* **Throughput:** The number of transactions processed by the payment gateway within a specific time period should be between 8 and 12 (TPS) or between 480 and 720 (TPM).
* **Third-Party Integrations:** Ensuring that the used APIs (Vodafone wallet, Orange wallet , Etisalat wallet , SMS and Email) do not negatively impact website performance
* **Response time:** The total response time should not exceed 3 seconds (including server processing time, network latency, and client-side rendering time)

**4.4 Maintainability**

* **Modularity**: By organizing the website into modular components or modules, each responsible for a specific functionality or feature.
* **Dependency Management**: The software should manage dependencies effectively, minimizing tight coupling between components and external dependencies.
* **Search Engine Optimization (SEO):** Implement SEO to improve the website's visibility and ranking in search engine results pages by optimizing meta tags, headers, URLs, and content for relevant keywords
* **Performance Optimization**: Optimize the website's performance by implementing magnification of CSS and JavaScript files, and leveraging content delivery networks (CDNs) to reduce latency and improve page load times.

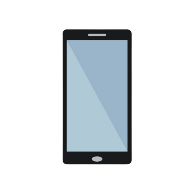
**5 Architecture**



Database Server

Web Server

Developers Computers



End user devices

**Developers Computers:**

Developing the website by creating user interface and connecting them with database

**Web Server:**

Getting hosting for the website on a web server to allow user to access the website and use its features

**Database Server:**

Getting hosting for the website on a database server to store users and advertisements info

**End user Devices:**

End users can access website features by PCs and smart phones

**Signature Page**

AGREED TO:

**Customer?**

**Signature**

**Name:**

**Title:**

**Date:**

AGREED TO:

**Team members**

**Signature**

**Name:**

**Title:**

**Date:**