Software Requirements Specification for < Bottle Technology >

Version 1.0

Prepared by < Utsav Shrestha >

<2020.11.22>

Table of Contents

Revision History

Name	Date	Reason for Changes	Version

1. Introduction

1.1 Purpose

Construct a working system with proper AWS services integration to achieve the following flow:

- A user should be able to enter their email in a simple web interface, and hit the "Subscribe" button
- If the email provided is not already present in our database table "Subscription emails", the email should receive a confirmation link. If the email is present, the email should instead receive an email saying "You are already our member!". This first API should respond 200 OK if successful.
- Upon clicking the confirmation link (second API), the user's email should be added to the database table "Subscription emails", and the user should be redirected to the website: www.google.com

1.2 Document Conventions

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

1.3 Intended Audience and Reading Suggestions

- Mr. Bipal Shakya
- Mr. Vivek Kansakar

1.4 Project Scope

For team lead assessment

1.5 References

N/A

2. Overall Description

2.1 Product Perspective

AWS based service intended for the collection of email address to be added to the mailing list for newspaper subscription

2.2 Product Features

The product lets users enter their email and if it is not present sends a confirmation link to their email.

2.3 User Classes and Characteristics

N/A

2.4 Operating Environment

AWS Cloud with integrated lambda functions.

2.5 Design and Implementation Constraints

As the product has been developed in a sandbox environment, currently email facilities are only available from verified email sources.

2.6 User Documentation

N/A

2.7 Assumptions and Dependencies

N/A

3. System Architecture

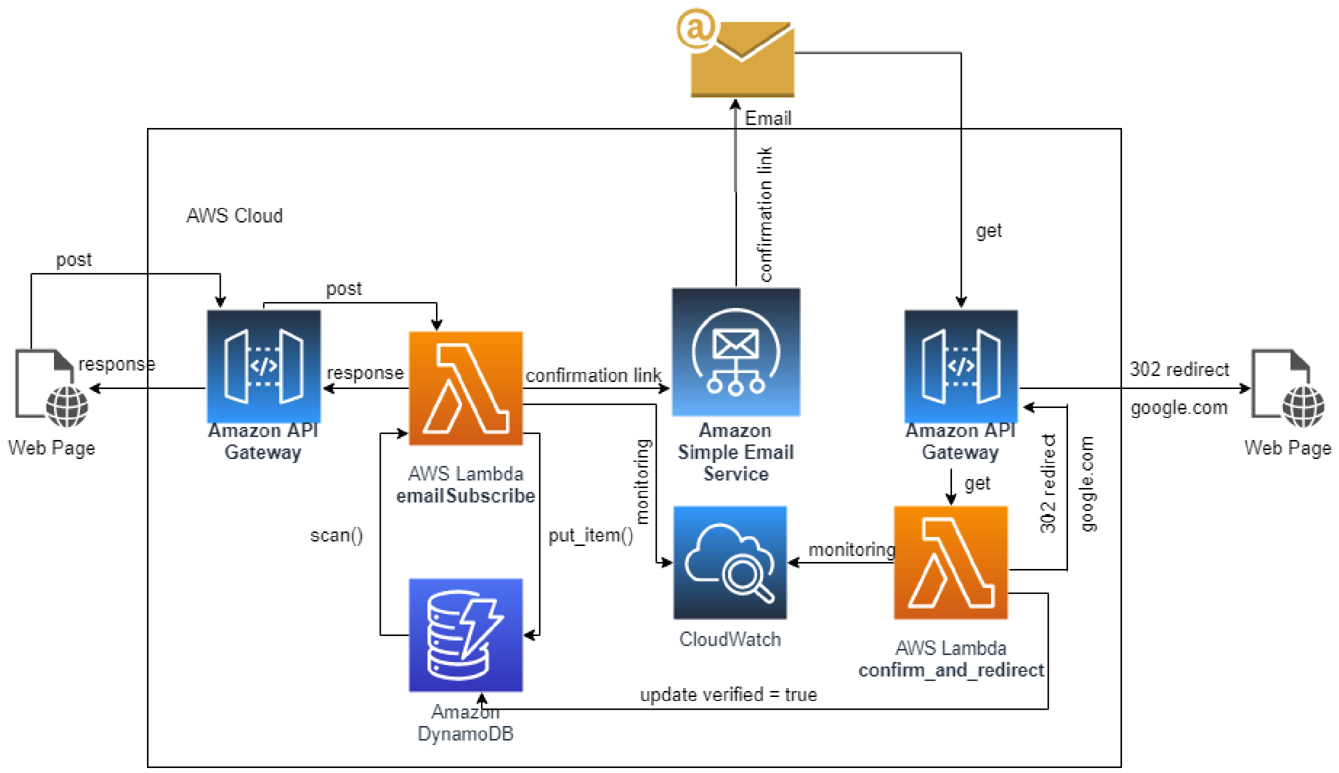

Figure 1 System Architecture

3.1 System Feature 1

<Don't really say "System Feature 1." State the feature name in just a few words.>

3.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

3.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

3.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

3.2 System Feature 2 (and so on)

4. External Interface Requirements

4.1 User Interfaces

Simple web user interface which allows the user to input their email address.

4.2 Hardware Interfaces

N/A

4.3 Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

4.4 Communications Interfaces

Communication will be done through two Amazon API Gateways.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

N/A

5.2 Safety Requirements

N/A

5.3 Security Requirements

N/A

5.4 Software Quality Attributes

N/A

6. Other Requirements

N/A

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: Issues List

< This is a dynamic list of the open requirements issues that remain to be resolved, including TBDs, pending decisions, information that is needed, conflicts awaiting resolution, and the like.>