Software Requirements Specification (SRS) document

Group 20
Kenrick Lam
Steven Tran
Lora Koo
Ritvik Shah

Contents

1. Introduction

1.1. Purpose

The purpose of Digital Health Inc.'s software application is to allow users to take pictures of food items and provide them with the caloric and nutrient make up of those food items with a minimum accuracy of 94.11%. Digital Health Inc. believes that this will be the most convenient, practical, and affordable method for logging daily food intake.

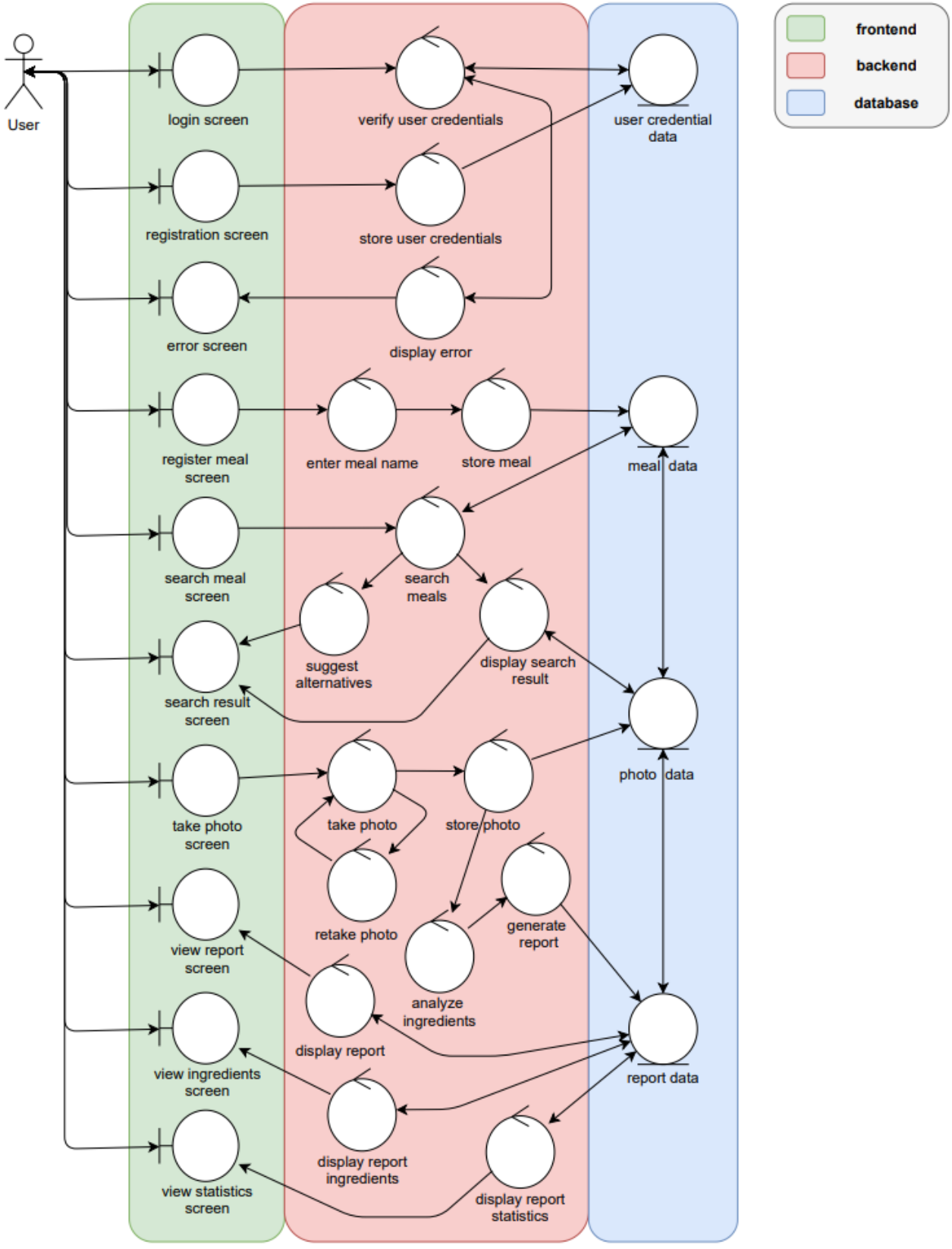
1.2. Scope

The software application is named "Personal Food Logger". This application allows the user to take a picture of a food item using the phone's camera, after which it will analyze the picture with the help of a machine learning algorithm to accurately display the caloric and nutrient makeup of the food.

The benefit of this application for the primary user is that it allows them to accurately and conveniently track the food they are eating to understand their eating patterns through a variety of statistics. Viewing relevant information such as the recommended dietary allowance percentage of macronutrients and micronutrients in the user's meal can allow them to adjust their diet accordingly to positively affect their health. The benefit of this application to Digital Health Inc. is that they can then potentially sell this application to healthcare providers, diet and health clinics, athletes, and individuals who either suffer from overweightness or simply want to stay fit.

As previously mentioned, the objective of Personal Food Logger is to act as a convenient vision-based measurement smartphone application that provides results with at least 94.11% accuracy. The accuracy requirement is necessary for Digital Health Inc. in order to convince clinics and professional athletic businesses to purchase this product.

1.3. Product overview



1.3.1. Product perspective

Digital Health Inc. is only looking to develop a software application that can automatically run on popular smartphones.

Hardware interfaces

The application will directly interface the phone's camera to allow the user to take photos. It will also directly interface with the memory and the processor to allow computation and storage of information.

User interfaces

The user will interface using a touch screen display where they will provide input using their fingers by touching the screen. The software will listen for user inputs and perform functions accordingly.

Communication requirements

The application will need to communicate with Digital Health's servers using a wireless application protocol that uses Wi-Fi signals to be able to send and receive data.

Operations

The application must meet the operation requirements for modern Android and iOS smartphones and must be easily accessible through the respective stores.
The user-initiated operations include taking a picture and selecting pictures to be analyzed. Background operations can include analyzing diet patterns based on earlier results and generating various graphs which help the user understand their eating patterns. The images taken from the camera will have to be pre-processed before they can be analyzed to ensure as accurate of a result as possible (adjustment can include white balance shift and exposure changes).

1.3.2. Product functions

The application will allow the user to take a photo of the food item using their phone camera. The application will then analyze the photo to reveal a comprehensive result showing the nutrient and caloric makeup of the food with a minimum accuracy of 94.11%. The application will also allow the creation of multiple user profiles so that various family members can separately measure their energy intake. The application will also provide various statistics that help the user understand their eating patterns over a long term.

1.3.3. User characteristics

Application user

This is the primary user of the application who uses it to measure their caloric intake by taking photos of their food and having the application analyze them. This type of user cannot view the source code of the application nor can they modify it.

Application maintainer

This person is employed by Digital Health Inc. and they are responsible for ensuring that the application’s services work as expected. This type of user can view the source code of the application and modify it if required.

1.3.4. Limitations

Regulatory requirements and policies

The application must meet the privacy laws of all the countries where it will be made available, which may or may not limit the functions of the application.

Hardware limitations

The accuracy of the results generated will be largely limited by the quality of the camera present on the smartphone as an unclear and low-resolution is significantly harder to analyze.

Quality requirements

The application will have to provide results with a minimum accuracy of 94.11% under ideal conditions.

Safety and security considerations

The application will have to securely store user data using the AES 256-bit encryption system on secure servers.

Physical/mental considerations

The application should not constantly remind the user that what they are eating is unhealthy or bad to avoid any mental difficulties caused.

1.4. Definitions

AES 256-bit encryption - Refers to an encryption standard that uses a 256-bit long key to encrypt data.

2. Requirements

2.1. Functions

a) Validity checks on the input shall be done against the database. Since nearly all of the system's functions require a unique ID, these IDs that are passed into the functions shall be matched against existing IDs in the database to determine if they exist and are unique. For inputs requiring strings from the user like entering a new meal name, the system can send an error response back to the user for using invalid characters or names.

b) The exact sequence of operations will depend on which functions are called. However, it shall mainly be composed of interactions between the frontend, backend, and database components of the application. The user will first trigger an event to happen by clicking some user interface component. The backend

system shall then determine which function was called and with any given inputs, it may then access the database for receiving, sending, or validating data. After interactions with the database, the system's function shall return something back to the user.

c) For abnormal situations with processing the inputs and outputs regarding overflow, the system shall do if-checks to determine if the ID values are within the correct data type range. If there is an operation causing overflow the system shall detect this issue and report the issue to the development team. The system shall also store the issue with a timestamp to the database, so the development team has records of potential recurring issues. With communication failures, these issues can be alleviated by having the system report which components are the point of failure in an operation. For example, if the user is attempting to log into their account but the system cannot return a response because it has an issue verifying the user credentials, the system shall report that there was an error when accessing the database. The only source of hardware in the application would be the computer that runs the application and its servers. Therefore, a response should be sent to the development team when the computer experiences a failure. Error handling and recovery can be handled both automatically and manually. For small issues like incorrect user input, these can be handled as mentioned above. However, for larger issues that require recovery, they will have to be handled manually on a case-by-case basis and would require more effort to fix.
d) Parameters for the system functions are primarily the IDs for the report, meal, photo, and user IDs. Since the IDs are unique, changing the passed in ID to its respective system function shall provide you a unique report, meal, photo, or user object that matches the passed in ID.
e) The other output sequences are the UI outputs which display the different webpages of the application. These shall be displayed by the system depending on which user prompt is clicked. For example, when the user triggers the view report feature, the system shall return a new page which displays the selected report.

2.2. Performance requirements

Static requirements:

- The application must initially support at least 50000 concurrent users.
- The database's storage capacity must be at least 2 TB in size.
- The application must be able to support at least 100,000 concurrent transactions.
- The software must hold at least 100 meal and report data objects per user.

Dynamic requirements:

- The software shall return a requested application page to the user in no more than 5 seconds when the load on server resources is <= 75%.
- The software shall remain active for at least 30 minutes when the load on server resources is <= 95% on average.
- Error messages shall be displayed to the user in < 1 second on average.
- 95% of application transactions shall be processed in < 1 second on average.
- The image recognition software shall analyze a meal's ingredients in <= 5 seconds on average.

2.3. Usability requirements

Effectiveness:

- The software shall correctly display whatever pages the user wishes to open.
- The image recognition software's accuracy will be continuously improved through user feedback.
- The functions of the application shall be protected by invalid user inputs by sending specific error messages back to the user, and by suggesting alternative inputs.

Efficiency:

- The software shall return views and results to the user with minimal delay.
- The software shall display exactly what the user requests to see and the most important information in that context.
- The application shall minimize the number of operations that are necessary to complete a single task e.g., creating a meal.

Satisfaction:

- The software shall allow users to retake their meal photo as many times as they want until they are satisfied with their selected photo.
- The user can send feedback for they want to see improved by the application and its image recognition software.
- The software shall allow for many opportunities for users to customize their profile with personal meals, photos, reports, etc.

Intuitiveness:

- The application shall break down large functions such as generating reports into numerous smaller tasks that users can follow one by one.
- The software shall provide UI triggers for each important function of the application.
- Each page shall describe clearly what its purpose using concise language.

Error tolerance:

- The application shall send error messages to the user for using invalid inputs and tries to suggest alternative, valid inputs when possible.
- The image recognition software shall learn from its errors via user feedback and incorporate changes for its next use.
- The software shall trace where the point of failure is during an error and send a response back to either the user if it is input related, or to the development team if it is an issue with the output.

2.4. Interface requirements

2.4.1. External interface requirements

- The software shall use an artificial intelligence API for image recognition.
- The software shall connect to AWS for uploading photos from the application.

2.4.2. Internal interface requirements

- The software shall utilize programming languages that support mobile application development.
- The programming language(s) used for the front-end design shall be simple and intuitive to use for the development team.

- The programming language(s) used for the back-end design shall support AWS and an API for image recognition.
- The database used for the software shall connect to AWS since photos will be uploaded there.

2.5. Logical database requirements

The following list contains information on what types of data will be stored in the system's databases. Although data will be stored using Amazon Web Services, it will still be beneficial to have backups of the food and user databases in physical databases. "Use" as defined in this section is the frequency at which a portion of the database is read from and written to by developers.

- Nutritional information of foods: used to generate nutritional information reports. Will be used extremely frequently by employees.
- Statistics on number of active users daily: used in conjunction with report generation data to assess success of the application. Will be used infrequently by employees.
- Statistics on number of reports generated daily: used in conjunction with active user data to gauge success of the application. Will be used infrequently by employees.
- User information:
    - Used infrequently by employees
- Logs of server downtime
    - Used infrequently by employees
- Logs of changes made to the database
    - Used infrequently by employees

2.6. Design constraints

The system is designed with a few external constraints in mind. The project is required to utilize Amazon Web Services (AWS), as Digital Health Inc. already has preexisting infrastructure that is built with AWS. Additionally, the system must be highly scalable and

perform its machine learning incrementally. This means that every new addition into the database should not require a complete retraining of the AI. As the size of the database will grow very large very quickly, it is not feasible for the AI to take long periods of time to train in accordance with new data.

2.7. Software system attributes

Reliability

To ensure the food detection is reliable, a vigorous training and validation process will have to occur. As we will have access to publicly available food datasets prior to the deployment of the software, we will have an opportunity to allow the AI to train itself until it is one of if not the most accurate food detection system in the market.

The storage capabilities of the system's servers will also be stress tested to meet the requirement of 100,000 concurrent transactions at a 90% uptime rate.

Security

To protect the system's information from being misused, a variety of failsafes will be in place. First, all passwords and credit cards will be encrypted so that in case of a data breach, it is unlikely that users will be divulged. Secondly, database access and alteration will be logged, so any suspicious behaviour will be quickly discovered, and appropriate countermeasures can be taken. Finally, not all employees will have the same permissions to prevent accidental misuse of the data, with more experienced staff having more permissions.

Maintainability

The system's level of maintainability will fall mostly on the difficulty of incrementally training the AI. It is required that the AI can be trained as new images and foods get processed, without having to completely retrain itself, so monitoring data entries that could potentially reduce the accuracy of the food detection system will be of utmost importance.

Portability

As much of the system is handled server-side, with the report generation being done entirely through AWS and the food database, this system has high amounts of portability. The applications shall be written in Swift and Kotlin for Apple and Android respectively, but these will only handle the UI and the API calls to AWS and the food database.

2.8. Supporting information

AWS is a cloud service that allows data to be stored remotely as opposed to having the information in local databases. This will be used when users upload photos during the nutritional report generation process, as required by Digital Health Inc. as they already have pre-existing infrastructure that uses AWS.

Input from the user will be in the format of a photo. This photo will then be uploaded to the AWS cloud server, where it will be processed by the food database and the user will receive the output of a nutritional report that includes a variety of information depending on subscription type.

3. Verification

3.1. Functions

The functionality of validation on the input against the database shall be tested through unit and functional testing. The implementation logic shall be inspected during design and code review. The returning error response in case of invalid input shall be inspected to ensure it contains all necessary information.

The sequence and logic of various operations will be tested for correct functionality and results through unit and functional testing. The expected results will be compared with actual results from the tests to ensure that the operation is performed correctly. The database backend will also be inspected to ensure that the correct expected data insert

or update has been done. The operation logic will also be inspected during design and code review to ensure correctness.

The protocol for abnormal situations and operations will be established and inspected during design, as well as continually maintained after deployment. If a manual operation such as bug fixes, error handling and recovery is needed, the development team will carry out the operation first in the staging environment to validate the correctness of logic and the results. Once the logic and results have been inspected, the development team will carry out the operation in the production environment.

The uniqueness of the system function parameters such as IDs will be validated through continuous database and code inspection. The functionality of any system function involving these parameters will also be unit and functionally tested for correctness of logic and output. The acceptance criteria for parameter correctness is 100%. The functionality for UI sequences will be functionally tested and the resulting output sequence will be compared against actual output sequence for correctness. The acceptance criteria for correct UI sequences is 100%.

3.2. Performance requirements

Static requirements

The system shall measure load by measuring the total concurrent users able to use the system at one time. The system shall test by sending a various number of users to perform a transaction at the same time. The acceptance criteria for simultaneous user testing is greater than or equals 50000 concurrent users. The acceptance criteria for simultaneous transaction testing is greater than or equals 100,000 concurrent transactions.

The system shall measure capacity by measuring the total storage capacity available by the system as well as per user. The system shall test by storing a various capacity of data within the system. The acceptance criteria for database’s storage capacity is greater than

or equals 2TB. The acceptance criteria for storage capacity per user is greater than or equals 100 meal and report data objects.

Dynamic requirements

The software shall measure performance by measuring time taken to return a requested application page to the user while load testing the server for various percentages of resources used. The acceptance criteria for request performance time is less than or equals 5 seconds when the load on server resources is less than or equals 75%.
The system shall test endurance of the software by testing the server for various load percentages and measuring how long the system performs at a stable level. The acceptance criteria for active endurance is greater than or equals 30 minutes when the load percentage is less than or equals 95%.

The system shall measure average response times for tasks such as displaying error messages, processing application transactions or analyzing meal ingredients from a photo. The acceptance criteria for response time is less than or equals 1 seconds for displaying error messages or processing application transactions, and less than or equals 5 seconds for analyzing meal ingredients from a photo.

3.3. Usability requirements

Effectiveness

The system shall be functionally and unit tested to ensure that tasks such as displaying of pages are done correctly. The acceptance criteria for task effectiveness is a 100% match between expected results and actual results. The image recognition software will receive a variation of different inputs and the resulting calculated outputs will be compared against the expected outputs for accuracy. The system will also be tested with various random inputs, both valid and invalid, to ensure that the system responds to user error correctly by detecting the error and displaying some error message. The acceptance criteria for effective error detection is a 100% match of times of invalid input and return of error having been detected.

Efficiency

The system shall be tested through both UI and an automated test suite to ensure lack of bottlenecks or delay within. Performance metrics such as the time taken for an action or user requests will be measured and compared against predetermined standards for performance acceptance criteria for each of the actions. The number of operations needed for each task will also be calculated and compared against predetermined standards for necessary operations needed for each of the tasks. The acceptance criteria for efficiency is that measured performance metrics such as time taken for an action should be equal to or less than the predetermined standard 95% of the time.

Satisfaction

There will be a team of test users who will try out using the system before the product is released. The test users will be asked a series of questions each on their user experience and satisfaction with the system for a user experience survey. If there is any feedback, they will be taken into account and implemented as well. The acceptance criteria for user experience is 95% of the test users being satisfied with the system.

Intuitiveness

Some randomly selected users from the team of new test users will not be given any instructions regarding the system before or during they use the system. The time they take to navigate the application and complete certain tasks will be recorded and compared against the time taken for new users with the instructions to have performed the same actions. The acceptance criteria for intuitiveness is a 15% difference in time taken for actions in users without any instructions compared to the users with the instructions.

Error tolerance

The system will be tested with various samples of invalid inputs. The system will offer the opportunity to validate input each time before submission. Once the input has been corrected and submitted, the results will be compared against the expected output. The

acceptance criteria for error tolerance is a 100% match from corrected output and expected output.

3.4. Interface requirements

3.4.1. External interface requirements

The system and its server shall be inspected for all required frameworks for utilizing an artificial intelligence API for image recognition. The system shall go through a complete set of code reviews and input-testing to ensure that all dependencies are installed and working as well as the correctness of the output from the API.

The system and its server shall be inspected for all required frameworks as well as connection to interact with AWS for uploading photos from the application. The system shall go through a complete set of code reviews and random input-testing to ensure that all dependencies are installed and working. The system shall go through connection testing to ensure that the connection between the application and AWS is working properly.

3.4.2. Internal interface requirements

A set of technical standards shall be set for development in regard to appropriate programming language choices. The technical standards will be enforced strictly by the development team throughout the development process to keep technical design consistent.

The system and its server shall be inspected for connection to AWS for uploading photos from the application. The system shall go through connection testing to ensure that the connection between the system database and AWS is working properly.

3.5. Logical database requirements

The database content requirements and constraints for each database will be ensured through inspection of the database during initial creation of the database, as well as regular maintenance of the database afterwards. By creating the database first in a staging environment, a team of database administrators will inspect the content, schema and other constraints of the database before deployment to the production environment.

Database tables that will be used frequently by users or employees will be stored on the Solid State Drive portion of the server while the tables that will be infrequently used will be stored on the Hard Disk Drives. The database will be regularly inspected and maintained by the database administration team.

3.6. Design constraints

The system will ensure the use of AWS and validate the system's connection to the service through connection, unit and functional tests. The scalability of the machine learning functionality of the system will also be validated through inspection of the machine learning system design and the functional testing of the system.

3.7. Software system attributes

Reliability

The system shall ensure food detection reliability by continually verifying the outputs from the AI using food datasets during training as well as after deployment. The AI team will inspect and verify if the results of the AI are correct.

The system shall measure load by measuring the total concurrent transactions able at one time. The system shall test by sending a various number of transactions at the same time at varying uptime rate. The acceptance criteria for concurrent transaction testing is greater than or equals 100,000 concurrent transactions when uptime rate is greater than or equals 90%.

Security

The information security protocols will be strictly established and enforced. All user, account and payment information in the database will be encrypted, and that will be verified regularly by inspection of the code, the database and the communication to the server. Logs of database access and alterations will be regularly inspected for any suspicious activities to ensure security of the data. All users and employees with any access to any part of the system will have differing roles and permissions given to them, and they will be given the minimum number of permissions possible for them to do their tasks. No user or employees will have access to any permissions that they do not need. The protocol of creating and assigning new permissions and roles to each user will be strictly enforced.

Maintainability

The training plan of the AI incrementally will be inspected before being initiated. Once the AI training begins, the process will be monitored by the AI team. The continuing trend for the accuracy of the AI will be analyzed. The acceptance criteria of the AI accuracy is 98%.

Portability

The code will be inspected during development and regularly after deployment to ensure that the correct programming language of Swift for Apple and Java for Android is used.

3.8. Supporting information

No verification information is needed for the supporting information section of the document.

4. Appendices

4.1. Assumptions and dependencies

No assumptions or dependencies were required in this document.

4.2. Acronyms and abbreviations

- VBM: Vision-based measurement
- UI: User Interface
- AI: Artificial Intelligence
- AWS: Amazon Web Service
- API: Application Programming Interface