

**New York University Abu Dhabi
CS-UH 2012: Software Engineering**

(Group 5)

**Software Requirements Specification
for
< *Dietade* >**

Version: (2)

Date: (03/2/2023)

Table of Contents

1. Introduction	4
1.1 Purpose	4
1.2 Scope	4
1.3 Definitions, Acronyms, and Abbreviations	4
1.4 References	5
1.5 Overview	6
2. The Overall Description	6
2.1 Product Perspective	7
2.1.1 System Interfaces	7
2.1.2 Interfaces	8
2.1.3 Hardware Interfaces	9
2.1.4 Software Interfaces	9
2.1.5 Communications Interfaces	11
2.1.6 Memory Constraints	11
2.1.7 Operations	11
2.1.8 Site Adaptation Requirements	12
2.2 Product Functions	13
2.3 User Characteristics	13
2.4 Constraints	14
2.5 Assumptions and Dependencies	15
2.6 Apportioning of Requirements	16
3. Specific Requirements	17
3.1 External interfaces	18
3.2 Functions	21
3.3 Performance Requirements	21
3.4 Logical Database Requirements	22
3.5 Design Constraints	23
3.5.1 Standards Compliance	24
3.6 Software System Attributes	25
3.6.1 Reliability	25
3.6.2 Availability	26
3.6.3 Security	26
3.6.4 Maintainability	27
3.6.5 Portability	28
3.7 Organizing the Specific Requirements	29
3.7.1 System Mode	29
3.7.2 User Class	30
3.7.3 Objects	30
3.7.4 Feature	31
3.7.5 Stimulus	33

3.7.6 Response	33
3.7.7 Functional Hierarchy	33
3.8 <i>Additional Comments</i>	35
4. Change Management Process	
5. Document Approvals	
6. Supporting Information	35

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) is to define the requirements and features of the Dietaide platform, which is a smart diet-building platform designed for hospital health professionals and patients. The SRS outlines the functional and non-functional requirements, user interfaces, system interfaces, performance requirements, and other specifications needed for the successful development and implementation of the software.

The intended audience of this SRS includes the development team, project stakeholders, and quality assurance team responsible for building and testing the software. This document serves as a reference guide for all parties involved in the software development lifecycle, ensuring that the final product meets the requirements of the hospital health professionals and patients who will use it.

By defining the scope and objectives of the Dietaide platform, this SRS will enable the development team to deliver a high-quality software solution that improves the efficiency and accuracy of diet planning and communication between health professionals and patients.

1.2 Scope

(1) The software product to be produced is Dietaide, a smart diet-building platform designed for hospital health professionals and patients.

(2) In general, Dietaide will:

- Allow health professionals to provide and access appropriate diets for patients based on their medical diagnosis.
- Enable users to create customized diet plans and facilitate communication between health professionals and patients regarding dietary needs.

However, Dietaide will not:

- Provide medical advice, diagnosis, or treatment recommendations.

(3) Dietaide's Benefits: Improve patient outcomes by ensuring that patients receive tailored dietary recommendations that are aligned with their medical diagnosis

Dietaide's Objectives: Aim to streamline the diet planning process, reduce manual effort, and improve accuracy.

Dietaide's Goals: By providing an intuitive and efficient platform for diet planning and communication, Dietaide will enable health professionals to focus on delivering better care to patients.

1.3 Definitions, Acronyms, and Abbreviations.

1. IP: Internet Protocol
2. TCP: Transmission Control Protocol
3. SQL: Structured Query Language
4. NoSQL: Not only SQL
5. HTTP: Hypertext Transfer Protocol Secure
6. HTTPS: Hypertext Transfer Protocol Secure
7. API: Application Programming Interface
8. JSON: JavaScript Object Notation
9. RAM: Random Access Memory
10. GB: Gigabytes

1.4 References

1. IP/TCP Protocol Suite:
Title: TCP/IP Illustrated, Volume 1: The Protocols Author: W. Richard Stevens
Date: 1994
Publishing Organization: Addison-Wesley Professional
Source: <https://www.amazon.com/TCP-Illustrated-Protocols-Addison-Wesley-Professional/dp/0201633469>
2. HTTP/HTTPS Protocol:
Title: Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing
Report Number: RFC 7230
Date: June 2014
Publishing Organization: Internet Engineering Task Force (IETF)
Source: <https://tools.ietf.org/html/rfc7230>
3. Amazon Web Services (AWS):
Title: AWS Documentation
Publishing Organization: Amazon Web Services, Inc.
Source: <https://aws.amazon.com/documentation/>
4. Microsoft SQL Server:
Title: Microsoft SQL Server Documentation
Publishing Organization: Microsoft Corporation
Source: <https://docs.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver15>
5. NoSQL:
Title: NoSQL Databases Explained
Publishing Organization: IBM
Source: <https://www.ibm.com/cloud/learn/nosql-databases-explained>
6. MongoDB:
Title: MongoDB Documentation
7. Publishing Organization: MongoDB, Inc.
Source: <https://docs.mongodb.com/>

8. Flask:
Title: Flask Documentation
Publishing Organization: Flask Documentation Project
Source: <https://flask.palletsprojects.com/en/2.1.x/>
9. UAE Ministry of Health and Prevention website (MOHAP):
Title: MOHAP
Publishing Organization: Government of the United Arab Emirates
Source: <https://www.mohap.gov.ae>

1.5 Overview

The requirements and features of the Dietaide platform are thoroughly described in this Software Requirements Specification (SRS) paper. The SRS details the performance requirements, user interfaces, system interfaces, functional and non-functional requirements, and other specifications required for the software's effective development and implementation.

The SRS is divided into a number of sections that offer in-depth details about the undertaking. Section 2, which provides comprehensive details about the functional requirements and features of the Dietaide platform, will be of particular interest to customers and prospective users of the software. This part describes the functionality and intended use of the software, as well as any constraints or limitations.

The material in Section 3 about the technical requirements and non-functional requirements of the software will be of most interest to developers and other technical stakeholders. The architecture, design, and implementation of the software are all covered in this part, along with any performance or security needs.

The SRS's other sections contain more details, like a glossary of terms, references, and appendices. Overall, the SRS acts as a guidebook for everyone involved in the software development lifecycle, ensuring that the finished product satisfies the needs of the hospital health workers and patients who will use it.

2. The Overall Description

Dietaide is a web application created to give medical staff members and patients an easy method to obtain and administer the right diets based on their diagnoses. By offering patients pertinent and individualized food advice depending on their medical circumstances, the platform is designed with the goal of improving patient outcomes.

The platform is made to help hospital health professionals overcome the difficulties they have when planning and overseeing the right diets for their patients. Conventional food management techniques can be time-consuming and error-prone, which results in less than ideal patient outcomes. Dietaide seeks to improve accuracy and efficiency while lessening the workload on hospital workers by streamlining the dietary management process.

Dietaide includes a variety of features to fulfill the demands of patients and medical professionals in order to accomplish these goals. These features include direct patient-doctor connection via chat channels, patient login and access to pertinent

diagnoses, automatic diet advice generation, doctor review and approval of generated diets, and automatic diet recommendation generation.

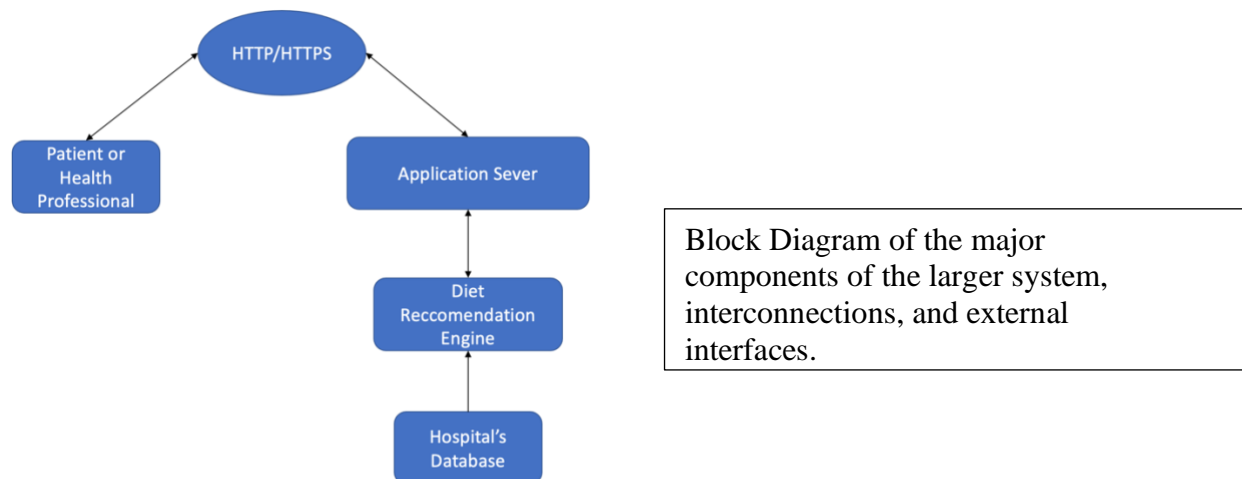
The platform has an intuitive user interface and is designed with usability and accessibility in mind. Overall, Dietaide is an excellent tool created to assist hospital health professionals and patients in better managing nutritional needs, enhancing patient outcomes, and lessening the workload on hospital staff.

2.1 Product Perspective

Dietaide is a standalone web application built specifically to function in a hospital setting. Even though it is not a part of a bigger system, it might communicate with other hospital systems, like the electronic medical record system, to access patient data and medical records.

Although there are a number of diet and nutrition-related web applications on the market, Dietaide is special in that it was created with medical staff and patients in mind. With a focus on streamlining and streamlining the process as much as possible, it offers an intuitive platform for accessing and offering recommended meals based on diagnosis.

Dietaide operates in accordance with all applicable hospital policies and regulations, including those pertaining to data security and privacy. The platform is made to effortlessly integrate with the infrastructure and workflows already in place at the hospital, causing no disruption or alteration to current procedures.



2.1.1 System Interfaces

As the Dietaide platform is a self-contained system, there are no external systems that it needs to interact with. However, the platform does have interfaces for different types of users, such as health professionals and patients, to interact with the system. These interfaces include:

1. **User interface:**

This interface allows users to interact with the system through a graphical user interface. Users can log in to the system, select relevant diagnoses, view generated diets, and communicate with health professionals.

2. Database interface:

The system database stores diagnosis-dietary correlations and patient profiles. The database interface allows the system to access and manipulate this data as needed to generate appropriate diets for patients.

3. Diet recommendation engine interface:

The diet recommendation engine uses the diagnosis-dietary correlations stored in the database to generate appropriate diets for patients based on their selected diagnoses and any additional conditions/severity. The engine interface allows the system to interact with this engine to generate and modify diet recommendations as needed.

Overall, these interfaces allow the different components of the Dietaide platform to interact with each other to provide an intuitive platform for health professionals and patients to access appropriate diets based on diagnosis. Assuming the hospital's database is managed with SQL, the API used to interact with the database will be SQLAPI++ if C++ is used or SQLite3 if Python3 is used. On the other hand, it is assumed that the database is managed with NoSQL if not SQL; in that case, the API is MongoDB.

2.1.2 Interfaces

(1) The logical characteristics of each interface between the Dietaide software product and its users include:

- Graphical User Interface (GUI) for patients and health professionals to interact with the system. The GUI should be intuitive and easy to navigate with clear instructions and feedback messages.
- A login interface for patients to access their hospital profile and upload their selected diagnoses relevant to their current dietary needs.
- An interface for health professionals to access patients' profiles and medical records within the hospital, view their selected diagnoses, generate diet recommendations, and make adjustments or other recommendations.
- A chat interface for direct patient-doctor communication on diet recommendations.

(2) Aspects of optimizing the interface with the person who must use the system:

- The GUI should be designed to be user-friendly and visually appealing to the target audience, which includes patients and health professionals.
- The system should provide clear instructions and feedback messages to users to help them navigate the system and understand how to use it effectively.

- The chat interface should be designed to facilitate clear and effective communication between patients and health professionals, with features such as real-time messaging, message notifications, and/or message history.

2.1.3 Hardware Interfaces

For Dietade, the software product will have the following interface with the hardware components of the system:

1. Server hardware: The software product will require a server to run on. The server must meet the minimum hardware requirements to run the software and support the expected number of concurrent users.
2. User hardware: The software product will support the use of standard web browsers on desktop computers.
3. Network interface: The software product will communicate with the hospital's internal network and internet to access patient data and provide diet recommendations. The system will use standard protocols such as HTTP and HTTPS to communicate over the network.
4. Data storage hardware: The software product will use a database management system to store patient data and diet recommendations. The database management system must meet the minimum hardware requirements to support the expected number of concurrent users and data storage capacity. The software product will support both SQL and/or NoSQL database management systems.

The software product will optimize its interface with the hardware components of the system by implementing efficient algorithms, minimizing the use of system resources, and following best practices for web development to ensure compatibility hardware and software configurations. The software product will also provide clear and concise error messages to help diagnose any hardware or software compatibility issues that may arise.

2.1.4 Software Interfaces

Dietade will interact with the following software products and interfaces:

1.
 - Name: MongoDB (if NoSQL is used)
 - Community Server Mnemonic: MongoDB
 - Specification Number: N/A
 - Version Number: 4.4.6
 - Source: <https://www.mongodb.com/>

- Purpose: MongoDB will be used as the database management system for Dietade. The system will interact with the MongoDB server through the MongoDB API to perform operations such as data insertion, querying and deletion.
- Interface: The interface between Dietade and MongoDB will be through the MongoDB API. The API will communicate with the database server using the BSON (Binary JSON) format. The data to be exchanged will be in the form of JSON objects.

2.

- Name: Microsoft SQL Server (if SQL is used)
- Mnemonic: SQL Server
- Specification number: N/A
- Version number: 2019
- Source: Microsoft
- Purpose: The system must use SQL Server as its database component. Communication with the DB is through ODBC connections.
- Interface Definition:
 - The system must provide SQL data table definitions to be provided to the company DBA for setup.
 - The system must connect to SQL Server using ODBC connections.
 - SQL queries must be executed using the appropriate syntax for SQL Server.
 - Data returned from SQL Server must be in a format compatible with the system's data structures.
 - SQL Server errors must be handled appropriately by the system, and appropriate error messages must be displayed to the user.

3.

- Name: Flask
- Mnemonic: N/A
- Specification Number: N/A
- Version Number: 2.0.1
- Source: <https://flask.palletsprojects.com/>
- Purpose: Flask will be used as the web framework for Dietade. It will handle incoming HTTP requests and route them to the appropriate application functions. It will also be responsible for generating HTTP responses to send back to the client.
- Interface: The interface between Dietade and Flask will be through the Flask API. The API will receive requests in the form of HTTP requests and send responses in the form of HTTP responses. The API will also use the Jinja2 templating engine to generate HTML templates for dynamic web pages.

4.

- Name: AWS S3
- Mnemonic: S3
- Specification Number: N/A

- Version Number: N/A
- Source: <https://aws.amazon.com/s3/>
- Purpose: AWS S3 will be used to store and manage user-uploaded files for Dietade. This includes profile pictures, meal plans, and other user-generated content.
- Interface: The interface between Dietade and AWS S3 will be through the AWS SDK for Python (Boto3). The SDK will communicate with the S3 server using the REST API over HTTP/HTTPS. The data to be exchanged will be in the form of binary data or multipart/form-data.

2.1.5 Communications Interfaces

1. TCP/IP Protocol:
Dietade requires support for TCP/IP protocol for communication between the client application and the server application.
2. HTTP Protocol:
Dietade requires support for HTTP protocol for communication between the client application and the server application.

These are referenced in the *References* section above.

2.1.6 Memory Constraints

The database management system must meet the following minimum memory requirements to support (by assumption) up to 100 concurrent users:

- Primary memory:
The system must have a minimum of 16GB of RAM available to allocate to the database management system.
- Secondary memory:
The system must have a minimum of 500GB of hard disk space available for storing data files.

These memory constraints are based on the expected number of concurrent users and the anticipated amount of data storage required. If additional users or data storage is needed in the future, the memory requirements may need to be adjusted accordingly.

2.1.7 Operations

1. Modes of operation:
 - User login and authentication
 - Data entry and editing

- Search and retrieval of data
 - Generating reports
 - Data analysis
2. Various modes of operations:
 - Dietade will be used in a continuous mode of operation, with users accessing the system throughout the day.
 - The system must be able to handle peak usage periods, which are expected to occur during the morning and lunchtime hours.
 3. Periods of interactive and unattended operations:
 - Dietade will primarily operate in interactive mode, with users accessing the system through a web-based interface.
 - However, some automated background processes will also be running periodically, such as data imports and exports.
 4. Data processing support functions:
 - Dietade must be able to import and export data in various formats, including CSV and Excel.
 - The system must also be able to generate reports on user activity, meal plans, and other metrics.

2.1.8 Site Adaptation Requirements

1. The system requires specific data to be provided by the customer, including:
 - User information: The system requires a list of authorized users and their access levels to be provided by the customer.
2. Site-related features that should be modified to adapt the software to a particular installation:
 - The system requires a reliable internet connection with a minimum bandwidth of 1 Mbps to ensure smooth operation.
 - The system should be installed on a server with a minimum of 8 GB RAM and 500 GB hard disk space.
 - The customer must ensure that the server is located in a secure and climate-controlled environment.
 - The system requires daily backups to be taken and stored off-site to ensure data protection.
 - The customer must ensure that adequate training is provided to the staff who will be using the system.
 - The customer must provide access to their existing database server for the integration of the system.

- The system requires a dedicated domain name or IP address for hosting the application.

2.2 Product Functions

1. User Management:

- Allow users to create and manage their accounts with login credentials and personal information
- Provide the ability for users to update their account information and reset their passwords if needed
- Authenticate users and control access to system features based on user roles and permissions

2. Nutrition Tracking:

- Allow users to input their dietary goals and restrictions
- Enable users to track their daily food intake and calculate nutritional values based on the provided product information
- Provide users with visual representations of their progress towards their nutritional goals

3. Meal Planning:

- Generate meal plans for users based on their dietary goals and restrictions, using existing meal plans provided by the customer as well as new plans generated by the system
- Allow users to customize meal plans by swapping out meals or adjusting portion sizes
- Generate grocery lists for users based on their chosen meal plans

5. Customer Support:

- Provide users with a help center containing resources and frequently asked questions
- Allow users to submit support requests and track their progress
- Enable customer support staff to view and manage support requests and provide assistance to users

6. System Administration:

- Allow administrators to manage user accounts and access permissions
- Provide tools for managing product information, including adding new products and updating nutritional values
- Enable administrators to view system usage data and generate reports on user activity and system performance.

2.3 User Characteristics

The intended users of Dietade are individuals who are seeking to improve their health through better nutrition. As such, the general characteristics of the users are as follows:

1. Educational level:

There are no specific educational requirements for using Dietade. However, users are expected to have a basic understanding of nutrition and healthy eating habits.

2. Experience:

Users of Dietade are likely to have a range of experience with nutrition and dieting. Some may be experienced in tracking their food intake and understanding nutritional information, while others may be relatively new to the concept.

3. Technical expertise:

Dietade is designed to be user-friendly and accessible to individuals with varying levels of technical expertise. However, users are expected to have basic computer skills such as the ability to navigate a web-based application and enter data using a keyboard.

The characteristics of the users will impact the design of Dietade in several ways. For example, the system will need to be designed with a user interface that is intuitive and easy to use for individuals with varying levels of technical expertise. Finally, the system *may* need to provide guidance and educational resources to users who are less experienced with nutrition and healthy eating habits.

2.4 Constraints

1. Regulatory policies:

The system must comply with relevant food safety regulations and guidelines, such as MOHAP requirements.

2. Hardware limitations:

The system must be able to operate on standard desktop and laptop computers with a minimum of 2GB RAM and 250GB hard drive space.

3. Parallel operation:

The system must be able to handle a maximum of 500 concurrent users without significant performance degradation.

4. Audit functions:

The system must log all user actions and transactions for auditing purposes.

5. Control functions:

The system must restrict access to certain features and data based on user roles and permissions.

6. Higher-order language requirements:

The system must be coded in Python 3.8 or higher, or C++, or Java.

7. Signal handshake protocols:
The system must support the ACK-NACK protocol for data transmission between the server and client applications.
8. Reliability requirements:
The system must have a 99.9% uptime guarantee and be able to handle spikes in traffic and usage.
9. Criticality of the application:
The system is critical to the customer's business and any downtime or errors could result in significant financial losses.
10. Safety and security considerations:
The system must adhere to industry-standard security practices, including secure data transmission and access controls based on user roles and permissions.

2.5 Assumptions and Dependencies

1. The customer will provide accurate and up-to-date nutritional information for all products to be used in the system.
2. The customer will provide necessary access to any third-party applications or systems that the software must interface with.
3. The customer will provide adequate hardware and network infrastructure to support the software.
4. The software will be developed and tested using the specified operating system and programming languages.
5. Any changes to the regulatory policies or standards will be communicated to the development team in a timely manner.
6. The customer will provide necessary training to their employees on the use of the software.
7. The customer will ensure that all data entered into the system is accurate and complete.
8. The software will be deployed and used in a safe and secure manner, in compliance with relevant laws and regulations.
9. The development team will have access to all necessary resources and personnel to complete the project within the given timeframe.
10. The system is dependent on the availability and reliability of the internet connection at the customer's site.
11. It is assumed that the customer's existing meal plans are in a format that can be easily integrated into the system.

12. The system is dependent on the performance and capacity of the hardware and infrastructure provided by the customer, such as servers and databases.
13. It is assumed that the customer will provide adequate training to their staff on how to use the system effectively.
14. The system is dependent on the availability and compatibility of any third-party software or services used in conjunction with it, such as payment processing or API integrations.
15. It is assumed that the customer will comply with any relevant regulations or standards related to the storage and handling of sensitive user data.
16. The system is dependent on the security measures implemented by the customer, such as firewalls and user authentication, to protect against unauthorized access or data breaches.
17. It is assumed that the customer's database system will be able to support the amount and type of data that the system will store and manipulate.
18. It is assumed that the customer's network infrastructure will be able to handle the expected traffic and communication between the system's components.
19. It is assumed that the customer's hardware and operating system configurations will be compatible with the software product and any third-party libraries or tools it relies on.
20. It is assumed that the customer's web server or hosting environment will meet the minimum requirements for the software product to run smoothly and securely.
21. It is assumed that the customer's payment gateway or financial service provider will provide reliable and secure payment processing services that integrate with the system's billing functionality.
22. It is assumed that the customer will be responsible for providing any necessary training or support to their own end-users who will be using the system.
23. It is assumed that the system will only support a rough maximum of 100 concurrent users.
24. It is assumed that the intended client and end-users have at least a limited working proficiency in the English language.

2.6 Apportioning of Requirements.

1. Must-Have Requirements:

These requirements are essential for the core functionality of the system and must be included in the initial release. Any delay in these requirements could impact the system's primary purpose. These requirements include:

- Patients should be able to login to the system with their hospital profile.
- Doctors and other medical professionals should be able to access patients' profile and medical record within the hospital.
- The system should be able to contain a database storing diagnosis-dietary correlations, and automatically generate diet recommendations based on the patient's selected diagnoses.

2. Should-Have Requirements:

These requirements are important but can be postponed until future versions of the system. They provide additional value to the system but are not critical for the initial release.

These requirements include:

- Patients should be able to select specific diagnoses from their hospital record that they find relevant to their current dietary needs, and upload them to the system.
- Doctors and other medical professionals should be able to review generated diets, make adjustments and/or other recommendations as they see fit, and then make approvals.
- Venues such as chat channels should be available for direct patient-doctor communication on diet recommendations.
- Additional features such as tracking and monitoring of patient progress, personalized alerts and reminders, and gamification elements to encourage adherence to prescribed diets.

3. Could-Have Requirements:

These requirements are desirable but are not necessary for the initial release. They can be postponed until later versions of the system.

These requirements include:

- Patients should be able to view all generated diets both before and after approval, attached to their own profile.
- The system should be able to attach the approved diet recommendations to the patient's profile for future reference.
- The system should be able to track the progress of patients' dietary adherence and provide recommendations accordingly.
- Integration/expansion into other hospitals to ensure completeness of medical records.

3. Specific Requirements

This section contains all the software requirements for Dietade at a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements. Throughout this section, every stated requirement should be externally perceivable by users, operators, or other external systems.

3.1.1 Authentication and Patient Profile Management Requirements:

- The system shall allow patients to create an account and log in using their hospital profile.
- The system shall provide patients with the ability to update their personal information in their profile.

3.1.2 Diagnoses Selection and Upload Requirements:

- The system shall allow patients to select specific diagnoses from their hospital record that they find relevant to their current dietary needs.
- The system shall provide patients with the ability to upload their selected diagnoses to the system.

3.1.3 Medical Professionals Access Requirements:

- The system shall provide medical professionals with access to patients' profiles and medical records within the hospital.
- The system shall allow medical professionals to view patients' selected "relevant" diagnoses.

3.1.4 Correlation Database and Diet Recommendations Requirements:

- The system shall contain a database storing diagnosis-dietary correlations.
- The system shall automatically generate diet recommendations based on the patient's selected diagnoses and any specific conditions/severity.
- The system shall attach generated diets to the patient's profile.

3.1.5 Diet Review, Adjustment, and Approval Requirements:

- The system shall allow medical professionals to review generated diets.
- The system shall allow medical professionals to make adjustments to generated diets. The system shall allow medical professionals to make other recommendations as they see fit.
- The system shall allow medical professionals to approve generated diets.

3.1.6 Patient Access to Generated Diets Requirements:

- The system shall allow patients to view all generated diets, both before and after approval, attached to their own profile.

3.1.7 Communication Requirements:

- The system shall provide venues such as chat channels for direct patient-doctor communication on diet recommendations.

3.1 External Interfaces

1. User Login

- Purpose: Allows patients to access their hospital profile and upload their dietary needs.
- Source of input: User input
- Destination of output: Dietade system
- Valid range: User credentials must be valid
- Timing: At system start-up or when accessing patient's profile
- Relationships to other inputs/outputs: N/A
- Screen formats/organization: Login screen with fields for username and password
- Window formats/organization: N/A
- Data formats: User ID and password (text)
- Command formats: Login button
- End messages: Error message if credentials are invalid, success message if login is successful

2. Diagnosis Selection

- Purpose: Allows patients to select specific diagnoses from their hospital record that they find relevant to their current dietary needs and upload to the system.
- Source of input: User input
- Destination of output: Dietade system
- Valid range: User-selected diagnoses must be valid and available in patient's medical record
- Timing: After user login and when updating dietary needs
- Relationships to other inputs/outputs: Linked to patient's hospital record
- Screen formats/organization: Diagnosis selection screen with a list of available diagnoses
- Window formats/organization: N/A
- Data formats: Selected diagnoses
- Command formats: Submit button
- End messages: Success message if selection is successful

3. Patient Profile

- Purpose: Allows doctors and other medical professionals to access patients' profile and medical record within the hospital, as well as view their selected "relevant" diagnoses.
- Source of input: Dietade system
- Destination of output: User interface
- Valid range: N/A
- Timing: At any time
- Relationships to other inputs/outputs: N/A
- Screen formats/organization: Patient profile screen with patient's personal and medical information, as well as selected diagnoses
- Window formats/organization: N/A
- Data formats: Patient information and selected diagnoses
- Command formats: N/A
- End messages: N/A

4. Diet Recommendations

- Purpose: Contains a database storing diagnosis-dietary correlations, automatically generates diet recommendations based on the patient's selected diagnoses as well as any specific conditions/severity, and attaches them to the patient's profile.
- Source of input: Patient's selected diagnoses
- Destination of output: Patient's profile in the Dietade system
- Valid range: N/A
- Timing: After patient's diagnosis selection
- Relationships to other inputs/outputs: Linked to patient profile and selected diagnoses
- Screen formats/organization: N/A
- Window formats/organization: N/A
- Data formats: Recommended diet plan
- Command formats: N/A
- End messages: Diet recommendations

5. Diet Approval

- Purpose: Allows doctors and other medical professionals to review generated diets, make adjustments and/or other recommendations as they see fit, and then make approvals.
- Source of input: Dietade system
- Destination of output: User interface
- Valid range: N/A
- Timing: After diet recommendations are generated
- Relationships to other inputs/outputs: N/A
- Screen formats/organization: Diet approval screen with recommended diet plan
- Window formats/organization: N/A
- Data formats: Approved diet plan and any recommended adjustments
- Command formats: Approve button
- End messages: Success message if approval is successful

6. Patient-Diet Communication

- Purpose: This interface allows patients and doctors to communicate directly in real-time for diet recommendations and consultations.
- Source of Input or Destination of Output: Patients and doctors both provide input and receive output through the chat interface.
- Valid Range, Accuracy, and/or Tolerance: The chat interface should allow for text-based communication with no specific limitations on character count or message frequency.
- Units of Measure: N/A
- Timing: The chat interface should be available for use 24/7 but response time may vary depending on availability of doctors.
- Relationships to Other Inputs/Outputs: The chat interface is related to the patient profile, medical record, and diet recommendations.
- Screen Formats/Organization: The chat interface should be organized into a conversation view with individual messages displayed chronologically. It should also include a text input field for sending messages and a scrollable message history.
- Window Formats/Organization: N/A

- Data Formats: The chat interface should support text-based messages.
- Command Formats: The chat interface should support basic commands such as sending messages, viewing message history, and ending the conversation.
- End Messages: The chat interface should include a message to confirm the end of the conversation.

3.2 Functions

1. The system shall allow patients to create a hospital profile with a unique username and password.
2. The system shall authenticate patient login credentials and allow access to their profile.
3. The system shall display a list of all the diagnoses from the patient's hospital record.
4. The system shall allow patients to select specific diagnoses from their hospital record and upload them to the system.
5. The system shall store the patient's selected diagnoses in their profile and link them to dietary recommendations.
6. The system shall allow doctors and other medical professionals to access patient profiles and view their selected diagnoses.
7. The system shall allow doctors and other medical professionals to review dietary recommendations generated based on the patient's selected diagnoses.
8. The system shall provide doctors and other medical professionals with the ability to make adjustments and/or other recommendations to generated diets.
9. The system shall allow doctors and other medical professionals to approve generated diets.
10. The system shall store all generated diets in the patient's profile, including those before and after approval.
11. The system shall provide chat channels or other communication tools for direct patient-doctor communication on diet recommendations.

3.3 Performance Requirements

Static numerical requirements:

- The system should support 100 combined personnel of health professionals and patients combined of the hospital.

Dynamic numerical requirements:

- The system should be able to handle a peak workload of 100 simultaneous users.
- 95% of login attempts by patients and health professionals shall be processed in less than 5 seconds.
- The system should be able to generate diet recommendations and attach them to patient profiles within 20 seconds of diagnosis selection.
- The system should be able to display patient's generated diets for viewing within 5 seconds of the request.
- The chat channels for direct patient-doctor communication should have a response time of less than 10 seconds for 95% of messages.

All requirements should be stated in measurable terms and ensure the security and privacy of the system and user data. The possible integration/expansion into other hospitals to ensure completeness of medical records should also be considered.

3.4 Logical Database Requirements

The following logical database requirements should be met:

1. The database should store patient profiles including personal information, medical history, and selected diagnoses.
2. The database should store diet recommendations generated by the system for each patient.
3. The database should contain a correlation table between diagnoses and dietary recommendations.
4. The database should be able to handle a large amount of data, as it will store information for all patients in the hospital.
5. Access to the database should be restricted to authorized personnel only.
6. The database should be able to handle frequent updates and queries from both patients and medical professionals.
7. Data entities in the database should be properly defined and their relationships should be established.

8. The integrity of the data in the database should be maintained through the use of appropriate constraints and validations.
9. The database should be designed to allow for easy expansion and scalability as the number of patients and the amount of data grows.
10. The system should ensure that data retention requirements are met as per relevant legal and regulatory requirements.

Now that the restrictions are stated, below is the relationships between the entities in Dietade and how information is exchanged between them.

Assuming the following entities:

1. User: representing both patients and medical professionals, with attributes such as name, email, password, and user type.
2. Diagnosis: representing medical diagnoses, with attributes such as name and description.
3. Diet: representing dietary recommendations, with attributes such as name and description.
4. Recommendation: representing the association between a diagnosis and a diet, with attributes such as severity level and date of recommendation.
5. Chat: representing direct communication between users, with attributes such as date and message content.

The relationships between these entities can be represented as follows:

- User-Diagnosis: many-to-many relationship between users and diagnoses, representing the selection of relevant diagnoses by patients and the access to patient profiles and medical records by medical professionals.
- Diagnosis-Recommendation: one-to-many relationship between diagnoses and recommendations, representing the association between a diagnosis and a dietary recommendation based on its severity level.
- Recommendation-Diet: one-to-one relationship between recommendations and diets, representing the automatic generation of diet recommendations based on the patient's selected diagnoses and severity level.
- User-Chat: many-to-many relationship between users and chats, representing the availability of chat channels for direct communication between patients and medical professionals on dietary recommendations.

3.5 Design Constraints

1. The software must be designed to meet relevant industry standards and regulations, such as HIPAA for patient data privacy and security.
2. The software must be compatible with the hospital's existing hardware and software infrastructure.
3. The software must be designed to work within the hospital's network security policies and protocols.
4. The software must be designed to handle high volumes of data and traffic, while maintaining optimal performance.
5. The software must be designed to be easily scalable and expandable to accommodate future growth and changes.
6. The software must be designed to work with multiple browsers and devices, to ensure accessibility for all users.
7. The software must be designed to be user-friendly and intuitive, to ensure ease of use and minimize training requirements.
8. The software must be designed to handle potential errors and exceptions in a graceful and informative manner, to minimize disruption and confusion for users.
9. The software must be designed to minimize data entry errors through the use of appropriate input validation and data formatting.

3.5.1 Standards Compliance

The Dietade software system must comply with the following standards and regulations:

1. Report format: All reports generated by the system must follow the standard format specified by the hospital.
2. Data naming: The system must follow the hospital's naming conventions for all data elements used in the system.
3. Accounting procedures: The system must comply with all accounting procedures and regulations specified by the hospital.
4. Audit Tracing: The system must provide an audit trail of all user activities, including logins, data access, modifications, and deletions. The audit trail must include the user ID, date, time, and action performed.
5. Privacy regulations: The system must comply with all relevant privacy regulations, including HIPAA, to ensure the confidentiality and security of patient information.

6. Data retention regulations: The system must comply with all relevant data retention regulations to ensure that patient information is stored for the appropriate length of time.
7. Security regulations: The system must comply with all relevant security regulations to ensure the security and integrity of patient information and system data.
8. Software development standards: The system must comply with software development standards, such as ISO/IEC 12207, to ensure high-quality software development practices are followed.

3.6 Software System Attributes

3.6.1 Reliability

Reliability is an important attribute for Dietade to ensure that the system functions as expected without errors or failures.

The following factors are required to establish the required reliability of the software system at the time of delivery:

1. MTBF (Mean Time Between Failures): The system should have a minimum MTBF of 500 hours to ensure that it operates without any significant downtime or failure.
2. Error Handling: The system should be designed to handle errors gracefully and provide informative error messages to users. The system should also have a mechanism to recover from errors and resume normal operation without any loss of data.
3. Backup and Recovery: The system should have a backup and recovery mechanism to ensure that data is not lost in case of system failure. The system should be able to recover from backups quickly and efficiently.
4. Testing: The system should undergo extensive testing to identify and fix any bugs or defects before deployment. This includes unit testing, integration testing, and system testing.
5. Maintenance: The system should be designed to be easily maintainable, with well-documented code and a modular architecture that allows for easy updates and upgrades.
6. Monitoring: The system should have a monitoring mechanism to detect any failures or errors in real-time. The monitoring system should notify the relevant personnel and take appropriate actions to prevent further damage or downtime.
7. Security: The system should be designed with security in mind to prevent unauthorized access, data breaches, or other security threats. This includes

implementing authentication and authorization mechanisms, data encryption, and secure communication protocols.

By adhering to these factors, Dietade can establish the required reliability of the software system at the time of delivery, ensuring that the system functions as expected and meets the needs of the users.

3.6.2 Availability

The availability of Dietade is a critical factor in ensuring uninterrupted access to the system for its users. The availability requirements for the system will depend on its usage pattern and the criticality of the functions it provides.

The system shall be available for access 24 hours a day, 7 days a week, with an expected uptime of 99.99%. This uptime expectation includes all scheduled maintenance windows.

To achieve the required availability level, the following factors are considered:

- **Checkpointing**: The system shall include periodic checkpointing to ensure that system state can be restored in case of a failure. Checkpointing intervals shall be configurable and set to a default value of every 5 minutes.
- **Recovery**: The system shall be able to recover from a failure state within a maximum of 30 minutes.
- **Restart**: The system shall allow users to restart the application after a failure with the loss of at most 2 minutes of data entered prior to the failure.

Additionally, the system shall provide a mechanism for monitoring system availability and performance. This will enable quick detection and resolution of any issues that may affect the availability of the system. The system shall log all system outages and notify the appropriate support personnel in the event of a system failure.

The availability requirements specified in this section are subject to change based on system usage patterns and feedback from users.

3.6.3 Security

1. The system shall require authentication for all users with a valid username and password.
2. The system shall implement role-based access control to restrict access to sensitive data and functionality based on the user's role.
3. The system *might* utilize encryption to protect the confidentiality of all sensitive data transmitted over the network.

4. The system shall maintain a log of all user activity, including login attempts, access to sensitive data, and changes to the system configuration.
5. The system shall enforce password complexity rules and require users to change their password periodically.
6. The system shall perform input validation and sanitization to prevent SQL injection and other types of attacks.
7. The system shall restrict access to critical system resources to authorized personnel only.
8. The system shall implement procedures to ensure the integrity of the data stored in the database, including regular backups and data validation checks.
9. The system shall provide a mechanism for reporting security incidents and vulnerabilities to the system administrator.
10. The system shall be designed to comply with relevant security standards and regulations.

3.6.4 Maintainability

The maintainability of the software system is crucial to ensure that it can be updated and modified easily by maintenance personnel.

The following factors are required to ensure the maintainability of the Dietade system:

1. **Modularity:** The system should be modularized so that different components can be modified independently of each other without affecting the overall system functionality.
2. **Documentation:** The system should be well-documented, including design documents, user manuals, and other necessary documents to facilitate maintenance and updates.
3. **Code Clarity:** The code should be written in a clear and concise manner, with well-defined interfaces and standard coding practices.
4. **Testability:** The system should be designed to facilitate testing, including unit tests, integration tests, and acceptance tests, to ensure that modifications and updates do not affect the system functionality.
5. **Version Control:** The system should be placed under version control, to allow maintenance personnel to track changes and revert to previous versions if necessary.

6. Error Handling: The system should be designed to handle errors and exceptions gracefully, with appropriate error messages and logging mechanisms.
7. Upgradability: The system should be designed to facilitate upgrades, including backward compatibility, so that new features and functionality can be added without affecting the existing system functionality.
8. Portability: The system should be designed to be portable to different platforms, operating systems, and environments, to ensure that it can be maintained in a variety of environments.

3.6.5 Portability

The portability of the Dietade system refers to its ability to be easily adapted and run on different host machines and/or operating systems. The following characteristics have been identified as critical for ensuring the portability of the system:

1. Percentage of components with host-dependent code (High Priority):
This characteristic refers to the percentage of the Dietade components that rely on specific host-dependent code. It is important to minimize the amount of host-dependent code to ensure that the system can be easily ported to different platforms.
2. Use of a proven portable language (High Priority):
This characteristic refers to the use of a programming language that is known to be highly portable across different platforms. The use of a proven portable language will minimize the amount of platform-specific code and make the system more adaptable to different environments.
3. Percentage of code that is host dependent (Medium Priority):
This characteristic refers to the percentage of the Dietade code that is dependent on specific host features or functionalities. While it is important to minimize the amount of host-dependent code, it may not be possible to eliminate it entirely. Therefore, this characteristic is given a medium priority.
4. Use of a particular operating system (Low Priority):
This characteristic refers to the requirement to run the Dietade system on a specific operating system. While the system should be designed to be as platform-independent as possible, it may be necessary to support a specific operating system. However, this requirement is given a low priority as it should not significantly impact the portability of the system.

To ensure the portability of the Dietade system, the following test cases will be executed:

1. Build and run the system on different operating systems (Windows, macOS, Linux) and evaluate the compatibility of the system with each platform.

2. Measure the percentage of components with host-dependent code and code that is host dependent to ensure they are within acceptable limits.
3. Evaluate the use of a proven portable language and assess its impact on the portability of the system.
4. Test the system on different hardware configurations to ensure compatibility and performance across a range of platforms.

The priority levels of these characteristics reflect their relative importance to the overall portability of the system. The high-priority characteristics must be addressed to ensure the system can be easily ported, while the medium-priority characteristic should be considered to minimize the amount of host-dependent code. The low-priority characteristic may be necessary in some cases but should not significantly impact the portability of the system.

3.7 Organizing the Specific Requirements

3.7.1 System Mode

Patient Mode

In this mode, patients have access to the following functionalities:

1. **Login:** Patients can access the system with their hospital profile, ensuring secure access to their dietary information.
2. **Selecting diagnoses:** Patients can select specific diagnoses from their hospital record that they find relevant to their current dietary needs and upload them to the system.
3. **Viewing generated diets:** Patients can view all generated diets, both before and after approval, attached to their own profile.
4. **Direct communication:** Venues such as chat channels are available for direct patient-doctor communication on diet recommendations.

Medical Professional Mode

In this mode, doctors and other medical professionals have access to the following functionalities:

1. **Access to patient profile and medical record:** Medical professionals can access patients' profiles and medical records within the hospital, as well as view their selected "relevant" diagnoses.
2. **Generating diets:** The system can contain a database storing diagnosis-dietary correlations and automatically generate diet recommendations based on the patient's selected diagnoses as well as any specific conditions/severity, and attach them to the patient's profile.

3. Reviewing and approving diets: Medical professionals can review generated diets, make adjustments and/or other recommendations as they see fit, and then make approvals.

3.7.2 User Class

1. Health Professionals:

This class of users includes doctors, nurses, and dietitians. They have access to the following functions:

- Access to patient profiles and medical records within the hospital
- Ability to view patients' selected relevant diagnoses
- Ability to review generated diets, make adjustments and/or other recommendations as they see fit, and then make approvals
- d. Ability to communicate with patients directly through chat channels regarding diet recommendations

2. Patients:

This class of users includes individuals who are admitted to the hospital and require dietary recommendations.

They have access to the following functions:

- Ability to login to the system with their hospital profile
- Ability to select specific diagnoses from their hospital record that they find relevant to their current dietary needs, and upload to the system
- Ability to view all generated diets both before and after approval, attached to their own profile
- d. Ability to communicate with health professionals directly through chat channels regarding diet recommendations

3. System Administrators:

The individuals responsible for managing and maintaining the Dietade system. They will have access to all features and functions of the system and will be responsible for managing user accounts, ensuring the security and privacy of the system and user data, and integrating or expanding the system into other hospitals to ensure completeness of medical records.

3.7.3 Objects

1. User

- Attributes: UserID, Username, Password, Email, Role
- Services: Login, Logout, View Profile, Edit Profile, Delete Profile

2. Patient

- Attributes: PatientID, Name, Age, Gender, Diagnosis, Medical Record, Profile Picture

- Services: View Profile, Edit Profile, Delete Profile, Upload Diagnosis, View Diagnosis, Generate Diet Recommendation, View Diet Recommendation, Communicate with Doctor
3. Doctor
 - Attributes: DoctorID, Name, Age, Gender, Specialization, Profile Picture
 - Services: View Profile, Edit Profile, Delete Profile, View Patients, View Patient Diagnosis, Approve Diet Recommendation, Communicate with Patient
 4. System Administrator
 - Attributes: AdminID, Name, Age, Gender, Profile Picture
 - Services: View Profile, Edit Profile, Delete Profile, Manage Users, Manage Diagnoses, Manage Diet Recommendations, Manage System Settings
 5. Diagnosis
 - Attributes: DiagnosisID, Name, Description, Dietary Recommendation
 - Services: Add Diagnosis, Edit Diagnosis, Delete Diagnosis
 6. Diet Recommendation
 - Attributes: RecommendationID, PatientID, DiagnosisID, Recommendation, Approval Status
 - Services: Generate Recommendation, Approve Recommendation, View Recommendation, Edit Recommendation, Delete Recommendation
 7. Chat Channel
 - Attributes: ChannelID, Participants, Messages
 - Services: Create Channel, Edit Channel, Delete Channel, Send Message, View Messages.

3.7.4 Feature

Feature 1: User Login Description

Stimulus:

1. The user clicks on the “Login” button.
2. The user enters their username and password
3. The user enters incorrect credentials

Response:

1. The system displays the login page.
2. The system validates the user’s credentials and grants access to the system.
3. The system displays an error message and prompts the user to enter valid credentials.

Feature 2: Doctor and Medical Professional Access

Stimulus:

1. The doctor or medical professional logs in to the system.
2. The doctor or medical professional selects a patient.

3. The doctor or medical professional views the patient's selected “relevant” diagnoses.

Response:

1. The system displays the doctor or medical professional’s profile page.
2. The system displays the patient's profile page and medical record.
3. The system displays the patient's selected “relevant” diagnoses.

Feature 3: View and Edit Patient Information

Stimulus:

1. User selects the “View Patient Information” option from the main menu.
2. User inputs the patient’s unique identifier.
3. System retrieves the patient’s information from the database.
4. System displays the patient’s information on the user interface.

Response:

1. System displays the patient’s name, age, gender, medical history, current diagnoses, allergies, and any other relevant information.
2. User selects the “Edit” option.
3. System enables the editable fields on the user interface.
4. User modifies the patient’s information as necessary.
5. User selects the “Save” option.
6. System validates the modified information and updates the database.
7. System displays a confirmation message to the user.
8. System disables the editable fields and displays the updated patient information.

Feature 4: Generate Diet Recommendations

Stimulus:

1. User selects the “Generate Diet Recommendations” option from the main menu.
2. User inputs the patient’s unique identifier.
3. System retrieves the patient’s information from the database.
4. System retrieves the patient’s selected diagnoses and any other relevant information from the database.
5. System accesses the diagnosis-dietary correlation database.
6. System generates a list of recommended diets based on the patient’s selected diagnoses and any specific conditions/severity.
7. System attaches the generated diets to the patient’s profile.

Response:

1. System displays a list of recommended diets on the user interface.
2. User selects a diet from the list.
3. System displays the details of the selected diet, including the allowed and restricted foods, and any additional notes.
4. User selects the “Approve” option.
5. System sends a notification to the doctor or health professional in charge of the patient.
6. Doctor or health professional reviews the recommended diet, makes adjustments and/or other recommendations as necessary, and approves or rejects the diet.

7. System updates the patient's profile with the approved diet.
8. System displays a confirmation message to the user.

Feature 5: Direct Communication

Stimulus:

1. User selects the "Direct Communication" option from the main menu.
2. User inputs the patient's unique identifier.
3. System retrieves the patient's information from the database.
4. System displays a list of available communication channels, such as chat, email, or video call.
5. User selects a communication channel.
6. System connects the user to the selected communication channel.

Response:

1. User and doctor or health professional communicate directly through the selected communication channel.
2. System records the communication and saves it to the patient's profile.
3. User selects the "End Call" option.
4. System terminates the communication channel.

Feature 6: Diet Approval

Stimulus:

1. The doctor or medical professional logs in to the system.
2. The doctor or medical professional selects a patient.
3. The doctor or medical professional reviews the generated diet recommendations.
4. The doctor or medical professional approves the diet. Response: The system updates the patient's profile with the approved diet.

Response:

1. The system displays the doctor or medical professional's profile page.
2. The system displays the patient's profile page and medical record.
3. The system displays the generated diet recommendations.
4. The system updates the diet recommendations.

3.7.5 Stimulus

N/A

Reason:

The dynamic of data, features, and methods are best described in a Function Hierarchy as detailed below in 3.7.7

3.7.6 Response

N/A

Reason:

The dynamic of data, features, and methods are best described in a Function Hierarchy as detailed below in 3.7.7

3.7.7 Functional Hierarchy

The functionality of Dietade can be organized into a hierarchy based on common inputs and outputs as follows:

1. User Management
 - 1.1. Login
 - 1.2. Logout
 - 1.3. Create Account
 - 1.4. Delete Account
 - 1.5. Update Account
2. Patient Profile Management
 - 2.1. Create Profile
 - 2.2. Update Profile
 - 2.3. View Profile
 - 2.4. Delete Profile
3. Diagnosis Management
 - 3.1. View Diagnoses
 - 3.2. Add Diagnosis
 - 3.3. Remove Diagnosis
4. Diet Recommendation Management
 - 4.1. Generate Diet Recommendation
 - 4.2. View Generated Diet Recommendation
 - 4.3. Approve Diet Recommendation
 - 4.4. Modify Diet Recommendation
5. Communication Management
 - 5.1. Patient-Doctor Communication
 - 5.2. System Notifications
6. System Administration
 - 6.1. Database Management
 - 6.2. User Management
 - 6.3. System Configuration

The above hierarchy shows the primary functions of the system, organized based on common inputs and outputs. The User Management function handles all user-related tasks, while the Patient Profile Management function deals with the management of patient profiles. The Diagnosis Management function enables doctors to add or remove diagnoses from patient profiles. The Diet Recommendation Management function generates and manages diet recommendations, including approving or modifying them. The Communication Management function handles all communication between patients and doctors and the system's notification mechanism. The System Administration function manages the database, system configuration, and user management for the system. Data flow diagrams and data dictionaries can be used to show the relationships between and among the functions and data.

3.8 Additional Comments

N/A

Reason:

All supporting information have been provided externally in separate documents from this SRS, as they do not directly affect the outcome of this document (and as per instructed by the professor)

4. Change Management Process

1. **Identification of Changes:** Changes can be identified by any member of the project team, the customer or other stakeholders. Changes must be documented in writing and submitted to the project manager.
2. **Evaluation of Changes:** The project manager, technical leads, and product owner will evaluate the change request and determine its impact on the project scope, timeline, and budget. They will also assess the feasibility and technical viability of the proposed change.
3. **Approval of Changes:** The product owner will approve or reject the change request based on its impact on the requirements and whether it aligns with the project goals. The project sponsor will also be consulted for any significant changes that may impact the project budget or timeline.
4. **Communication of Changes:** Once a change has been approved, the project manager will communicate the change to the relevant team members and stakeholders via email. This may include updating the SRS, project plan, and other project documentation.
5. **Implementation of Changes:** The technical leads will implement the change request in the system and ensure that it meets the customer's requirements and quality standards.
6. **Monitoring and Control of Changes:** The project manager will monitor the implementation of changes and ensure that they are completed within the agreed-upon timeline and budget. Any deviations from the plan will be reported to the project sponsor and product owner, and appropriate corrective action will be taken.

To ensure that changes are managed effectively, the project team will hold regular meetings to review the status of the project, identify risks and issues, and discuss any proposed changes. The change management process will be documented in the project plan and communicated to all relevant stakeholders. It is important to note that the customer may suggest changes to the requirements, but all changes must be evaluated and approved by the project team before they can be implemented. The team will reach a consensus on whether a change is necessary and if it can be accommodated within the existing project plan

5. Document Approvals

The following are the identified titles of the approvers of this SRS document:

1. Project Manager
2. Business Analyst
3. Technical Lead
4. Quality Assurance Lead
5. Customer or Stakeholder Representative

Name 1: _____

[Insert Name of Approver]

Signature 1: _____

[Insert Signature]

Date: _____

[Insert Date]

Name 2: _____

[Insert Name of Approver]

Signature 2: _____

[Insert Signature]

Date: _____

[Insert Date]

Name 3: _____

[Insert Name of Approver]

Signature 3: _____

[Insert Signature]

Date: _____

[Insert Date]

Name 4: _____

[Insert Name of Approver]

Signature 4: _____

[Insert Signature]

Date: _____

[Insert Date]

Name 5: _____

[Insert Name of Approver]

Signature 5: _____

[Insert Signature]

Date: _____

[Insert Date]

Any changes to the SRS document for the Dietade project must be approved by the same individuals listed above. Changes must be submitted through the Change Management Process outlined in the SRS document.

This Document Approvals section ensures that the SRS document for the Dietade project has been thoroughly reviewed and approved by the appropriate individuals before proceeding with the development phase. It also ensures that any changes to the document are properly reviewed and approved by the same individuals to maintain consistency and accuracy throughout the project.

6. Supporting Information

N/A

Reason:

All supporting information have been provided externally in separate documents from this SRS, as they do not directly affect the outcome of this document (and as per instructed by the professor)