****

**Design and Implement of** **Health Management Platform for the College Groups**

**CPS 4951 SPRING 2022**

Software Requirements Specification Document

**Feng Xinlei (Selena) 1098297**

**Ling Ziang (Eric) 1098506**

**Xu Haiyang (Ambrose)** **1098129**

# Version: (1) Date: (2/27/2022)

**Table of Contents**

1. [Introduction *5*](#_TOC_250045)
   1. [Purpose 5](#_TOC_250044)
   2. [Scope 5](#_TOC_250043)
   3. [Definitions, Acronyms, and Abbreviations 6](#_TOC_250042)
   4. [References 6](#_TOC_250041)
   5. [Overview](#_TOC_250040) 6
2. [The Overall Description *6*](#OLE_LINK6)
   1. [Product Perspective 7](#OLE_LINK61)
      1. [System Interfaces 7](#OLE_LINK62)
      2. [Interfaces 8](#OLE_LINK62)
      3. [Hardware Interfaces 9](#OLE_LINK63)
      4. [Software Interfaces 9](#OLE_LINK64)
      5. [Communications Interfaces 9](#OLE_LINK65)
      6. [Memory Constraints 9](#OLE_LINK66)
      7. [Operations 9](#OLE_LINK67)
      8. [Site Adaptation Requirements 9](#OLE_LINK68)
   2. [Product Functions 10](#OLE_LINK621)
   3. [User Characteristics 10](#OLE_LINK623)
   4. [Constraints 10](#OLE_LINK624)
   5. [Assumptions and Dependencies 11](#OLE_LINK625)
   6. [Apportioning of Requirements](#OLE_LINK626) 11
3. [Specific Requirements *12*](#OLE_LINK7)
   1. [External interfaces 12](#OLE_LINK71)
   2. [Functions 13](#OLE_LINK3Funcion)
   3. [Performance Requirements 14](#OLE_LINK3Performance)
   4. [Logical Database Requirements 15](#OLE_LINK3Logical)
   5. [Design Constraints 19](#OLE_LINK3Design)
   6. [Software System Attributes 21](#OLE_LINK3Software)
   7. [Organizing the Specific Requirements 22](#OLE_LINK3Organizing)
   8. [Additional Comments](#OLE_LINK3Additional) 24
4. [Change Management Process](#OLE_LINK4Change)  *24*
5. [Document Approvals](#OLE_LINK5Document)  *24*
6. [Supporting Information *25*](#_TOC_250000)

# Introduction

# 1.1 Purpose

Health Management Platform (HMP) is a Java and Web-based system designed for college groups, to collect, visualize, and analyze the physical examination data, improving the health status of the group members.

1.2 Scope

# For the front desk functions:

# 1. Teachers' diet management: provide each teacher a recommended food menu based on their daily consumption and relevant information.

# 2. Teacher health diary: users’ sleep, drinking, dimensionality, eating, weight can be self-recorded every day.

# 3. Teachers' physical examination management: each result in the physical examination can be demonstrated on the platform; it also shows a trend chart over time.

# 4. Disease prediction and evaluation: mathematical linear regression is used to predict the probability of teachers' illness in terms of diabetes and hypertension.

# For the background functions:

# 1. Administrator can log in to the system via a specific ID.

# 2. Teacher regulation function: administrator can modify teachers’ general information.

# 3. Physical examination results management: administrator could change and check the records from teachers.

# 4. Health news: publish or delete available health news.

# 1.3 Definitions, Acronyms, and Abbreviations.

HMP: Health Management Platform, is the name of our project, and we are specifically constructed for college groups.

**1.4 References**

This part is shown in the appendix

**1.5 Overview**

Section 2 clarifies the broad information about the project, including hardware and software interfaces, constraints, and end-user features. Section 3 contains explicit functions and requirements, as well as the attributes to realize them.\

1. **The Overall Description**

Nowadays health is getting more and more important in our daily life. To take care of college groups’ health conditions, we plan to design and implement a health management platform. The Java language and SSH framework (Spring, Struts, Hibernate) will be used. Some basic functions include eating and health diary, medicine record, disease forecast. When teachers document these items of data on the website, they can keep a watchful eye on their physical state, and adjust their eating, sleeping, and exercising accordingly. Moreover, this system also provides reminds to prevent hypertension and diabetes.

The HMP system applied J2EE three-tier structure, containing three layers called the presentation layer, middle layer, and data service layer. Clients are separated from accessing directly to the database, they instead react with the database by the middle tier, which defines basic business rules, how to access data, and certificate verification.

**2.1 Product Perspective**

**2.1.1 System Interfaces**

1. Login function: it provides a text field to enter username and password, and users can log in or register an account.
2. Link to backstage: it has a button that leads the administrator to manage the backstage.
3. Health news: health news from all around the world will be picked to enrich teachers’ knowledge.
4. Main module 1 — teacher physical examination management.
5. Main module 2 — teacher daily meal management.
6. Main module 3 — teacher health log.
7. Main module 4 — teacher disease accurate prediction and evaluation.

HMP has a main system interface demonstrated in Figure 2.1 below:

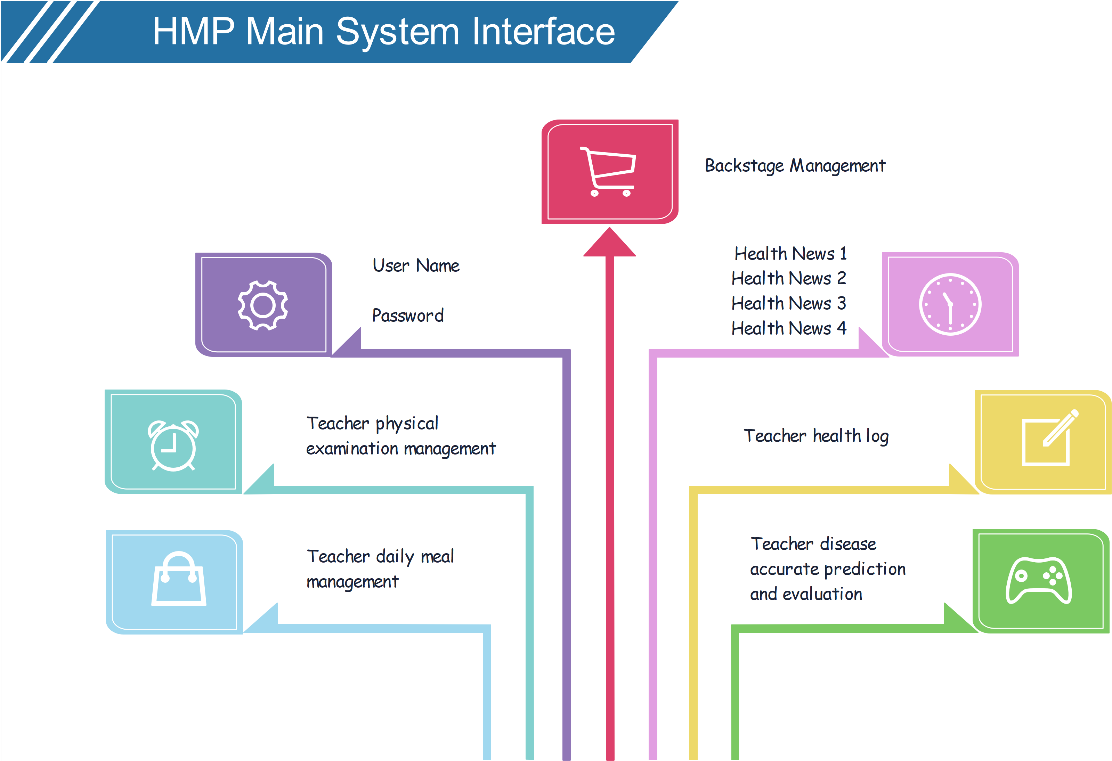


Figure 2.1 HMP main system interface

**2.1.2 Interfaces**

1. Registration function interface. New users need to register an account by filling in some text fields. Each of them has some hints, and the entered text will be verified with a particular regular expression to make sure the validation of the information.
2. Teacher physical examination management interface. Import the physical examination data directly from the college health center backstage. Each time of the physical examination results can be displayed on the platform, by using Highcharts generated in JavaScript. It is useful in showing dynamic and interactive charts on web pages, and non-commercial usage is popular (Hua et. al, 2018). Right now, At present, the HMP involves curve charts, area charts, histograms, pie charts, scatter charts and comprehensive charts are the types of charts in favor of Highcharts. Furthermore, with the advantages of excellent compatibility, most present browsers can perfectly run it (Shahid, 2014).
3. Teacher daily meal management interface. It primarily finds the calories that each teacher needed every day referring to their age, gender, height, weight, and activity, and provides a weekly suggested recipe.
4. Teacher health log interface. There is some text field that requires users to record their health journal, containing information about sleeping hours, mood, drinking volume, smoking number, blood pressure, weight. Some common diet is also available for users to click, or they can add their dishes.
5. Teacher disease accurate prediction and evaluation interface. The target diseases are diabetes and hypertension. According to Chi & Lee, the risk factors such as obesity and anthropometric indices should be highly considered (2022). Beyond that, genetic influence, smoking, age, exercise frequency, blood pressure, and other physical examination data are also included in the linear regression prediction model.

**2.1.3 Hardware Interface**

This project is designed to run on a computer and use a browser to enter the HMP website, so the minimum requirement for its hardware is Pentium III Microprocessor for the processor; 0.8 GHz for the main frequency; 256 MB for the memory; 10GB for the hard disk.

**2.1.4 Software Interface**

We will use JDK, Tomcat, My Eclipse, Oracle, and other software to build the development environment.

**2.1.5 Communication Interface**

The communication or interaction between each layer is that in the Web layer, it has interaction between the business logic layer and the user interface; and the communication between Data Access Object (DAO) and the Persistence Object (PO).

**2.1.6 Memory Constraints**

Designing a website, developing and deploying a server, running My Eclipse need 10 GB for memory.

**2.1.7 Operations**

DAO layer has the main goal to interact with PO. This layer encapsulates the operations of data addition, deletion, query, and modification. PO maps the database with objects based on their entity-relationship, then the administrator could manage the database conveniently.

**2.1.8 Site Adaptation Requirements**

JQuery is strong and compatible with mainstream browsers, such as IE, Firefox, Chrome, Safari, etc, only old versions of IE 6, IE 7, IE 8 cannot be supported (Raigoza & Thakkar, 2016). Under the structure of JavaScript, jQuery provides developers with an easy way to write plug-ins.

Asynchronous JavaScript and XML (AJAX) is also a technique used in the system because a faster, diverse, and user-friendly website can be developed. The main idea is to asynchrony data communication between the browser and the webserver.

**2.2 Product Functions**

Beyond what is mentioned in 2.1.2 the interfaces on the main page, which can be accessed by the user, this system also has robust functions for backstage administrators to operate. The backend functions are in the following.

1. The backend administrator login function asks administrator to enter their ID and password.
2. Backend manages and checks administrator function: it shows the administrator’s name and password and contains functions to add, delete, and modify administrator information.
3. Administrators manage physical examination results function. It showed all the records of teacher physical examination on the webpage, as well as the function of uploading data from Excel sheet via the HSSF interface.
4. The administrator manages the health news function. News content can be added, delete, and modified; Struts is used to deal with the upload of pictures.

**2.3 User Characteristic**

User scope involves professors and staff members who work in colleges or universities. Since they are facing high pressure in teaching and researching, fast working pace, irregular living habits, and tension from social relationships, some of them are in the high-risk group of catching diseases of hypertension, diabetes, and other chronic diseases.

**2.4 Constraints**

1. This system plan to use Java programming language as a developing tool; with the help of the open-source J2EE structure, and JSP technology.
2. Database access: Oracle database
3. Server access: Tomcat series server
4. Web programming technology: DIV + CSS, JavaScript, AJAX, etc.
5. Background processing technology: Struts 2 + Hibernate + Spring
6. Hardware requirements: PIII / 512M / 10 GB and above configuration
7. Operating system: Windows 2000 and above

**2.5 Assumptions and Dependencies**

One assumption about this application is that there are requirements for the performance of the mobile phone. If the phone does not have enough hardware resources available for the application, for example, the user may have assigned it to another application, there may be situations where the application does not perform as expected or even not at all.

Another assumption is user uses Android operating system, if the OS change from Android to IOS, there may be some problems. IOS system is mainly installed on iPhone, iPad, iTouch and other devices, while Android system is mainly installed on Android smartphones. There are more types of Android smartphones on the market today because of the diversity of types and the greater openness of Android software development, while Apple software development is more specialized.

**2.6 Apportioning of Requirements**

The system will follow the sequence of requirements:

1. Teachers have easy access to the five basic functions.

2. Independent of the specific database platform. Full consideration of compatibility for the parts that interact with the database.

3. Easy secondary development on the original basis.

4. The shapelessness of the intermediate base platform. Try to adopt open standard technology to achieve the effect of cross-platform operation.

# Special Requirements

**3.1 External Interfaces**

**3.1.1 User Interfaces**

Front-end software:

Back-end software:

**3.1.2 Hardware Interfaces**

Windows

A browser which supports CGI, HTML & JavaScript

**3.1.3 Software Interfaces**

Following are the software used for the management online application. <Include the software details as per your project>

|  |  |
| --- | --- |
| Software used | Description |
| Operating system | We have chosen Windows operating system for its best support and user-friendliness. |
| Database | To save the flight records, passengers records we have chosen SQL+ database. |
| Eclipse | Feature-rich Java EE integrated development environment , including a complete coding , debugging , testing and publishing features , complete support for HTML, Struts, JSP, CSS, SQL, Javascript, Spring, Hibernate. |

**3.1.4 Communication Interfaces**

This project supports all types of web browsers. We are using simple electronic forms for the reservation forms, data entry, etc.

**3.2 Functions**

**3.2.1 Front-end main page**

The main page of the health management platform has the login and registration functions for teachers, the entrance to the background interface, the entrance and brief introduction of the four main functions, the display of health news, the dynamic display of the campus landscape and the links to the websites of relevant departments.

**3.2.2 Teacher diet management function**

The teacher diet management function mainly calculates the daily calorie requirement of teachers according to their height, weight, age, gender and activity level, and gives a recommended menu for a week.

**3.2.3 Teacher health diary function**

Users can record their own health diary, but also record each day's sleep, drinking, smoking, blood pressure, weight, mood, etc., to enter their own day's diet, record the health diary using kindeditor.

**3.2.4 Teacher physical examination management function design**

The school hospital imports the physical examination results from the backend, and teachers can query the results of each physical examination through the platform, and use Highcharts to display the curves of some items of the physical examination over time, such as blood pressure, blood sugar, etc. Highcharts is a charting library written in pure JavaScript that makes it easy to add interactive charts to web sites or web applications and is free for personal study, personal websites and non-commercial use. Currently, Highcharts supports curve, area, bar, pie, scatter, and composite charts. In addition, Highcharts has good compatibility and can perfectly support most current browsers.

**3.2.5 Disease prediction and assessment function design**

Mainly for two diseases, diabetes and hypertension, according to the user's medical examination related items and the database of similar people in the last ten years, the analysis of the risk factors for the disease, the use of mathematical linear regression method, to establish a prediction model, the future medical examiners can input the relevant medical examination items to simulate and calculate the probability of teacher disease.

**3.2.6 Registration function design**

New users can register as prompted, each with a specific regular expression for form validation, to ensure that the information is valid and available.

**3.3 Performance Requirements**

3.3.1 Diversity: An important principle that must be followed in the design of a health management platform is that the user is the center of the system design. The development of the network has greatly reduced the cost of information consultation, and users are faced with more choices, so the health management platform must work on the appearance, and diversity of functions, in addition to reducing costs, in order to be recognized by users. Specifically, the application system should be able to achieve the specific needs of users, with a friendly and convenient operating interface and function settings that meet the habits and requirements of users.

3.3.2 Easy maintenance: The administrator can simply and easily manage the system, define new applications, and perform daily maintenance on the system; with the change or increase of health management platform information and the increase of data scale, the system upgrade and maintenance should be easy to operate; the design of each subsystem and functional module should have good flexibility to ensure that the realization of system functions at a later stage in the health management platform system construction process does not affect the functions of the system that has been put into use earlier.

3.3.3 Security: The system should fully consider the security of information, equipment, operation and management, and establish a perfect multi-level security system to ensure the data security of users' personal information and other materials.

3.3.4 Scalability: The users of the health management platform are engaged in various industries, and their needs vary. The system should be designed to take full account of the need for changing user needs and to reduce the complexity of the system when expanding new business functions.

**3.4 Logical Database Requirements**

**3.4.1 Detailed database design**

Data items and data structure are as follows:

1. Manager: administrator number, administrator name, administrator password.

2. Notice: news number, news title, news content, release time, picture path, publisher.

3. Food: food number, food name, food calorie, food type, food carbohydrate, fiber, fat, protein.

4. Exam: systolic blood pressure, diastolic blood pressure, body mass index, waist-to-hip ratio, urea, uric acid, creatinine, triglycerides, cholesterol, HDL cholesterol, LDL cholesterol, glucose, homocysteine, microalbumin, microalbumin/urine creatinine.

5. Teacher: teacher number, teacher nickname, teacher password, teacher age, teacher gender, teacher email, password retrieval question, password retrieval answer, teacher real name.

6. Dairy: log content, log time, sleep time, mood, smoking, weight, stress, high pressure, low pressure, heart rate.

7. Dairy-food: log number, food number.

**3.4.2 Database logical structure design**

Database logical structure design summary table, as shown in Table 3.1:

|  |  |
| --- | --- |
| Table name | Function Description |
| Manager | Manager information |
| Notice | News information |
| Food | Food information |
| Exam | Physical examination indicators |
| Teacher | Teacher information |
| Dairy | Dairy information |
| Dairy-food | Dairy-food information |

Table 3.1 Database logical structure design summary table

**3.4.3 Database Table Design**

There are seven tables involved in the database, which are Manager table, News table, Food table, Dairy table, Dairy-food table, Physical examination table and Teacher table.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field name | Field meaning | Data type | Length | NULL | Note |
| id | ID | Number | 10 | NO |  |
| name | Username | Varchar2 | 20 | NO |  |
| password | Password | String | 20 | NO |  |

Table 3.2 Manager table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field name | Field meaning | Data type | Length | NULL | Note |
| id | ID | Number | 10 | NO |  |
| title | Username | Varchar2 | 50 | NO |  |
| content | Password | Varchar2 | 2000 | NO |  |
| datetime | Publish time | Timestamp | 6 | NO |  |
| imagepath | Image path | Varchar2 | 30 | NO |  |
| manager\_id | Publisher | Number | 10 | YES |  |

Table 3.3 Notice table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field name | Field meaning | Data type | Length | NULL | Note |
| id | ID | Number | 10 | NO |  |
| name | Food name | Float | 20 | NO |  |
| heat | Calories | Float | 126 | YES |  |
| carbohydrate | Carbohydrate | Float | 126 | YES |  |
| fat | Fat | Float | 126 | YES |  |
| fiber | Cellulose | Float | 126 | YES |  |
| protein | Protein | Float | 126 | YES |  |
| type | Type | Number | 10 | NO |  |

Table 3.4 Food table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field name | Field meaning | Data type | Length | NULL | Note |
| datetime | ID | Varchar2 | 20 | NO | Note |
| text | dairy | Varchar2 | 2000 | YES |  |
| sleephour | sleep | Float | 126 | YES |  |
| mood | mood | Varchar2 | 255 | YES |  |
| cigrette | smoke | Number | 10 | YES |  |
| weight | weight | Float | 126 | YES |  |
| pressure | pressure | Varchar2 | 255 | YES |  |
| bphigh | High blood pressure | Float | 126 | YES |  |
| bplow | Low  Blood  pressure | Float | 126 | YES |  |
| bs | Heart rate | Float | 126 | YES |  |
| teacher\_id | Teacher id | Number | 10 | YES |  |

Table 3.5Dairy table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field name | Field meaning | Data type | Length | NULL | Note |
| dairy\_id | Dairy id | Varchar2 | 20 | NO |  |
| food\_id | Food id | Number | 10 | NO |  |

Table 3.6Dairy-food table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field name | Field Meaning | Data type | Length | NULL | Note |
| id | ID | Number | 10 | NO |  |
| datetime | Medical examination time | Varchar2 | 6 | NO |  |
| sbp | Systolic pressure | Float | 255 | YES |  |
| dbp | Diastolic blood pressure | Float | 255 | YES |  |
| bmi | Body mass index | Float | 255 | YES |  |
| whr | Waist-to-hip ratio | Float | 255 | YES |  |
| bun | Cholesterol | Float | 255 | YES |  |
| ua | Urea | Float | 255 | YES |  |
| crea | Uric acid | Float | 255 | YES |  |
| tg | Creatinine | Float | 255 | YES |  |
| chol | Triglycerides | Float | 255 | YES |  |
| glu | Glucose | Float | 255 | YES |  |
| hcy | Homocysteine | Float | 255 | YES |  |
| m\_alb | Microalbumin | Float | 255 | YES |  |
| m\_alb\_crea | Microalbumin/urinary creatinine | Float | 255 | YES |  |
| teacher\_id | Teacher id | Number | 10 | YES |  |

Table 3.7 Exam table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field name | Field meaning | Data type | Length | NULL | Note |
| id | ID | Number | 10 | NO |  |
| name | Teacher name | Varchar2 | 20 | NO |  |
| password | Password | Varchar2 | 20 | NO |  |
| age | Age | Number | 10 | NO |  |
| sex | Gender | Varchar2 | 10 | NO |  |
| email | email | Varchar2 | 30 | NO |  |
| question | Password retrieval problem | Varchar2 | 50 | NO |  |
| answer | Password retrieval problem  answer | Varchar2 | 50 | NO |  |
| phone | Phone number | Varchar2 | 11 | NO |  |
| realname | Real name | Varchar2 | 20 | NO |  |

Table 3.8 Teacher table

**3.5 Design Constraints**

The Design Constraints section actually specifies design constraints that can be imposed by other criteria, such as hardware limitations, environmental constraints, etc. Design constraints are required for a project to succeed. Design constraints can sometimes feel like a negative thing, but they help shape the project to fit the exact needs of the customer. System constraints describe the constraints that designers choose when building a product and how the product will operate under a set of conditions. Constraints in our software systems include user interface constraints, hardware constraints, software constraints, communication constraints, data management constraints, operational constraints, and site adaptation constraints.

**3.5.1 Constraints:**

**User Interface Constraints:**

1. Restrict different permissions for normal users and administrators in operations.
2. Improve the happenability of the interface, the information of the interface should contain necessary information and there should be no redundant information to reduce the system complexity.
3. The system should provide different personal preference settings for users.
4. The interface should be kept up-to-date at all times.
5. The system graphical user interface (GUI) should adhere to industry standards.

**Hardware Constraints:**

Hardware constraints refer to the requirements for the physical components of the computer, including the central processing unit (CPU), memory (RAM), auxiliary storage, and the pairing of the various components with their enmity. The platform we develop will be mainly based on JavaSSH development of the site, as our main development tool is JAVA, there is no great need for hardware, but it is recommended to use Windows system for development.

**Software constraint:**

Software constraint is assumed that specific software will be available and necessary for the operation of the product.

1. the system must run on Windows.

2. Java must be installed on the system.

3. Local files must be accessible to provide users with the ability to upload local information or images.

**Communications constraints:**

Description of communication devices or protocols with which the product must interact

1. System must be able to communicate with output devices, like Printer for printing health statements.
2. System must communicate through TCP/IP and standard network protocols.

**Data Management Constraints:**

Describe data flow constraints for data management software and devices that are outside the scope of the project. Which means that reports and information in the system (HMP) are able to be saved locally, or can be printed out directly.

**Operational Constraints：**

This section actually describes how the product will operate in its environment, and in our HMP project we list some of the major constraints.

1. The system can be used by up to 50 teachers at the same time

2. The system can store over 1000 physical examination records.

3. System uses up to 10% of memory, 10% of disk space and 20% of company internet bandwidth.

4. Minor Bugs (display error messages, other display failures) These will remain at a maximum of 10 errors/KLOC.

5. Severe Bugs (e.g., too much user access, not enough access) These will remain at a maximum of 10 errors/KLOC.

6. Major Bugs (e.g. system crash, data loss, unable to create output) these will remain at a maximum of 10 errors/KLOC.

**3.5.2 Standards Comliance**

This section specifies requirements derived from existing standards or regulations. There are many software attributes that can be used as requirements. The following list specifies the factors that must be established for an ePECS system to work properly. Existing standards and regulations impose the following restrictions on our systems.

1. The system should be designed using open source languages and/or free software.

2. The system interface is developed in Java and can run on multiple operating systems.

3. The system needs to use oracle or MSQL to store data.

4. The system is developed using windosw .

5. The system should be designed to be able to switch between the interface and/or back-end (database).

6. Web programming technology (front-end) using DIV+CSS, JAVASCRIPT, AJAX and other tools.

7. backend processing technology using, Struts2, Hibernate and Spring.

**3.6 Software System Attributes**

**3.6.1 Versatility**

HMP must be designed with the user at the center of the system design, and should be able to implement user-specific needs with a user-friendly and easy-to-use interface and feature set that meets user habits and requirements.

**3.6.2 Reliability**

Users should not cause the system to crash when using the HMP, and developers should not crash the system when making subsequent improvements.

**3.6.3 Portability (non-local)**

Users can access the platform from any computer with an Internet connection.

**3.6.4 Maintenability**

Administrators can simply manage the system, define new applications, and perform routine maintenance of the system, including adding, deleting, changing, and checking information; subsystems and functional modules should be designed with good flexibility to ensure that the system is compatible with each other at later stages and in previous work.

**3.6.5 Security**

The system should fully consider the security of information, equipment, operation and management, and establish a perfect security system, while ensuring the data security of users' personal information and other information.

**3.6.6 Scalability**

The users of the health management platform are engaged in various industries and their needs vary. When designing the system, the changing user needs should be fully considered, and the complexity of the system should be reduced when expanding new business functions.

\*: Both 3.7 and 3.8 are supplementary documents

**3.7 Organizing the Specific Requirements**

**3.7 below is not really a section, it is a suggestion on how we should organize the requirements written in Section 3.2.**

Since detailed requirements are often broad, this requires us to think carefully and organize these specific content in the most understandable way.

**3.7.1 System Mode**

The behavior of certain systems varies by mode of operation. When organized by mode, there are two possible outlines. The choice depends on the interface and whether the performance depends on the schema.

**3.7.2 User Class**

Some systems provide different sets of functions for different classes of users.

**3.7.3 Objects**

Objects are real-world entities that have counterparts in the system. Associated with each object is a set of properties and functions. These functions are also referred to as services, methods, or processes. Note that sets of objects may share properties and services. These are combined as classes.

**3.7.4 Feature**

Features are externally visible services provided by the system that may require a set of inputs to achieve the desired result. Each feature is typically described as a sequence of stimulus-response pairs.

**3.7.5 Stimulus**

Some systems can be best organized by describing their functions in terms of stimuli.

**3. 7.6 Response**

Some systems can be best organized by describing their features to support response generation.

**3.7.7 Functional Hierarchy**

When none of the above organizational schemes are helpful, the overall functionality can be organized into functional hierarchies organized by public inputs, public outputs, or public internal data access. Data flow diagrams and data dictionaries can be used to show the relationships between functions and data.

**3.8 Additional Comments**

When we are considering a new SRS, more than one organizational technique given from 3.7 is appropriate. It also means that we need to customize the specific requirements for multiple levels of structure based on the specific needs of the system under the specification. In the actual development process, we should not rigidly apply the organization techniques in the requirements document, but we need to follow sections 3.7 and 3.8 for analysis and design as closely as possible to fulfill the requirements in the requirements document according to the actual situation.

# 4. Change Management Process

The change of project plan can be caused by many factors, and it will influence the following change in documents, human resources, and it may affect product quality (Rudic & Sobajic, 2012). After the change of management process, one important issue is to take care of the documents, software, and database implementation referring to the new requirements. The control over the interdependency among each element should be considered, and the correction of the code should be careful, otherwise, it may cause a collapse of the system.

When meeting any issues in using the product and need some changes to the platform, please contact Ziang Ling by email at ziangl@kean.edu.

# 5. Document Approvals

Professor Pinata Winoto will take responsibility to approve the software requirements specification.

# Supporting Information

**Appendix I References**

Chi, J. H., & Lee, B. J. (2022). Risk factors for hypertension and diabetes comorbidity in a Korean population: A cross-sectional study.*PLoS One, 17*(1) doi:http://dx.doi.org/10.1371/journal.pone.0262757

Hua, E. C. C., Nen, V. Y., Tee, F. S., & Ann, O. C. (2018). Pigeon-chart: A customized HTML element for data visualization in data-driven web application using AngularJS, HighCharts, UnderscoreJS and PHP. Piscataway: The Institute of Electrical and Electronics Engineers, Inc. (IEEE). Retrieved from <https://kean.idm.oclc.org/login?url=https://www.proquest.com/conference-papers-proceedings/pigeon-chart-customized-html-element-data/docview/2184766633/se-2?accountid=11809>

Raigoza, J., & Thakkar, R. (2016). *Browser performance of JavaScript framework, SAPUI5 & jQuery*. Piscataway: The Institute of Electrical and Electronics Engineers, Inc. (IEEE). Retrieved from <https://kean.idm.oclc.org/login?url=https://www.proquest.com/conference-papers-proceedings/browser-performance-javascript-framework-sapui5/docview/1879756417/se-2?accountid=11809>

Rudic, T., & Sobajic, V. (2012). CHANGE MANAGEMENT PROCESS OF SOFTWARE PROJECTS.*Metalurgia International, 17*(11), 179-184. Retrieved from https://kean.idm.oclc.org/login?url=https://www.proquest.com/scholarly-journals/change-management-process-software-projects/docview/1114880461/se-2?accountid=11809

Shahid, B. (2014). *Highcharts essentials*. Packt Publishing, Limited.