VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CBS1005 Software Engineering Methodologies

Lab Assessment-2 (SRS)

Name: Omkar Kabadagi

Registration Number:19BBS0038

Slot: L59+L60

Date:29/08/2020

Software Requirements Specification

for

Health and Fitness Tracking Website

Version 2.0 approved

Prepared by 19BBS0031-Rithvik Ayithapu Sai

19BBS0038-Omkar Kabadagi

19BBS0039-Akhil Chaitanya Ghanta

SEM, SCOPE, VIT University, Vellore, 29/08/2020

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

To develop a web app that can track calorie intake, blood pressure, sugar levels and provide insights about the user's health. Internet is a resource that has seen widespread adoption and usage over the past few decades. This resource can be effectively utilized to track and maintain health records of the user which can be used to assess the quality of the user's health. This is the SRS for the first release of the website.

1.2 Document Conventions

Every requirement statement in this SRS document has its own priority.

1.3 Intended Audience and Reading Suggestions

This document is intended for development team, design team and the management team. Testers may refer this document to gain an understanding of the features of the software. This SRS contains a description of all the features that will be a part of the software.

1.4 Product Scope

Internet is a resource that has seen widespread adoption and usage over the past few decades. This resource can be effectively utilized to track and maintain health records of the user which can be used to assess the quality of the user's health. The objectives of this project are to develop a website that tracks the health and the fitness of the user and provide suggestions to the user based on the metrics provided.

1.5 References

Some references used for making this SRS are:

Cse.msu.edu. n.d. [online] Available at: <http://www.cse.msu.edu/~cse870/IEEEXplore-SRS-template.pdf> [Accessed 21 August 2020].

Allassignmentexperts.com. 2020. [online] Available at: <https://allassignmentexperts.com/sample_assignments/programming/Software%20Requirements%20Specification%20-%20Gym%20App.pdf> [Accessed 21 August 2020].

2020. *Software Requirements Specification GYM App*. [ebook] Available at: <https://allassignmentexperts.com/sample_assignments/programming/Software%20Requirements%20Specification%20-%20Gym%20App.pdf> [Accessed 21 August 2020].

Trap.ncirl.ie. 2020. [online] Available at: <http://trap.ncirl.ie/2706/1/markkirby.pdf> [Accessed 21 August 2020].

Teamfusion.zerograpes.com. n.d. [online] Available at: <http://teamfusion.zerograpes.com/ProjectPlan/SRS.pdf> [Accessed 21 August 2020].

Mendiola, Martin & Kalnicki, Miriam & Lindenauer, Sarah. (2015). Valuable Features in Mobile Health Apps for Patients and Consumers: Content Analysis of Apps and User Ratings. JMIR mHealth and uHealth. 3. e40. 10.2196/mhealth.4283.

Adewumi, Adewole & Olatunde, Godwin & Misra, Sanjay & Maskeliunas, Rytis & Damasevicius, Robertas. (2018). Developing a Calorie Counter Fitness App for Smartphones. 10.1007/978-3-319-74980-8_3.

2. Overall Description

2.1 Product Perspective

This product is a new, self-contained product. It is a project started with a clean slate with no influence from any previous products that have been made. This product is not a part of another larger system. The process model that has been chosen Incremental SDLC. This process model has been chosen since functions having maximum priority have been added to the product first, after which more and more features will be added. In this project the functions having maximum priority are login and signup, home page and the trackers. These will be built first. After this, the diets page and calorie tracker will be made. Lastly the overview page will be made completing the entire project. Just like an Incremental SDLC project, all the components of the product have been decided beforehand and the design has also been worked on, after which the implementation stage begins and functions with maximum priority are built first followed by functions with lower priority.

2.2 Product Functions

Some of the major functions of the product are:

- Tracking calories, blood sugar, blood pressure and other attributes related to the health of an individual.
- Based on the recorded data the product must analyze and provide suggestions to improve the individual's health.
- Tracked data of the user shall be stored on a database to show progress made by the user over time.
- A food catalog shall be provided for the user to pick dishes from to track his/her calories. The user may also add dishes which are not in the catalog.
- The dishes added into the catalog shall be regularly examined. If the data regarding the dish provided is valid, it will be given a verified status.

2.3 User Classes and Characteristics

The various user classes who will use the product are:

- Medical personnel – These users shall use the product more than once in a day.
- Health enthusiasts – These users shall use the product every week.
- Senior citizens – These users shall use the product on a biweekly or monthly basis.
- Athletes – These users shall use the product every day.

This classification is made based on technical expertise and experience. Since the user base is so broad the project must be simple and easy to use and understand.

2.4 Operating Environment

The product will run on any web browser. It is a website that will work on mobile phones, desktops and laptops. The recommended web browser versions are:

- Google Chrome 84
- Safari 13.1
- Edge 84
- Firefox 79
- Opera 69

2.5 Design and Implementation Constraints

The implementation constraints require us to use NodeJS and MongoDB. The implementation has to be completed within 15 days. The software has to be maintained by the development team.

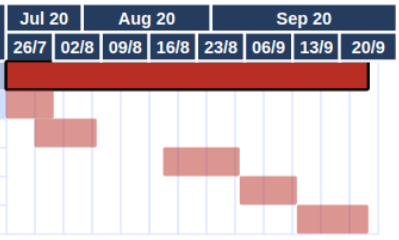
2.6 User Documentation

The user will be supplied with a “Help” page which can be referred to understand how to use the product.

2.7 Scheduling diagrams

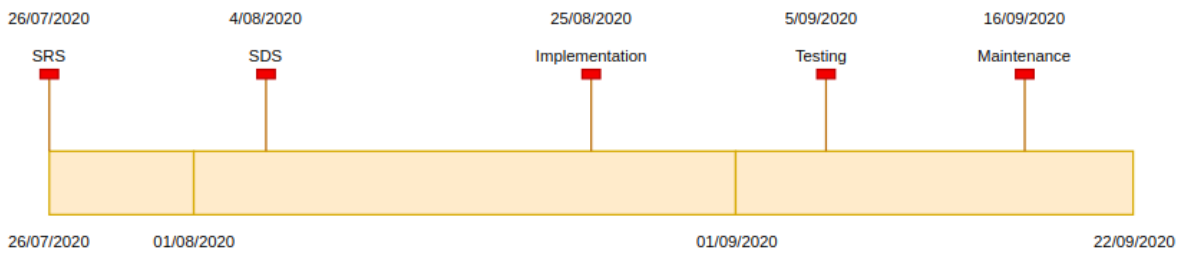
1. Gantt Chart

	Task Name	Duration	Start	ETA
	Complete project execution	62 days	26.07.20	22.09.20
1	SRS (Survey, Functions: Omkar, Rithvik)	7 days	26.07.20	02.08.20
2	SDS (Detailed model: Akhil)	8 days	30.07.20	07.08.20
3	Implementation (Front-end: Rithvik | Back-end: Omkar)	16 days	18.08.20	02.09.20
4	Testing (Test cases: Akhil)	6 days	03.09.20	09.09.20
5	Maintainance	13 days	09.09.20	22.09.20



Gantt chart along with work distribution

2. Timeline Graph

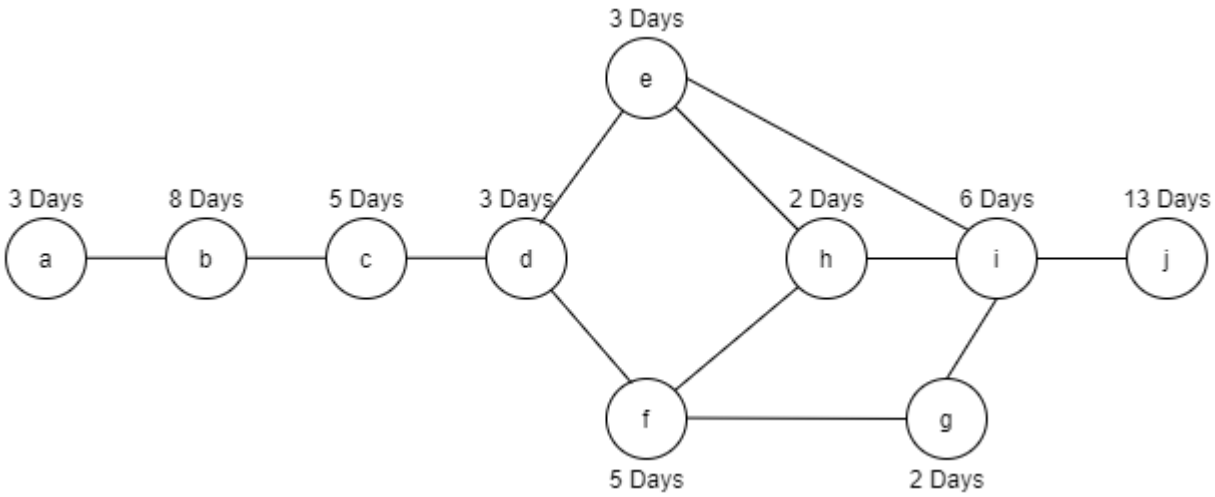


Project Timeline

3. PERT(Program Evaluation Review Technique) Chart

SR. NO.	SUBTASK	TIME	DEPENDENCY
a	Scheduling Diagrams	3 Days	
b	Deciding Features	8 Days	a
c	Website Design	5 Days	b
d	Login, Home Page	3 Days	c
e	Trackers-BP,Weight,Blood Sugar	3 Days	d
f	Diets Page	5 Days	d
g	Calorie Tracker	2 Days	f
h	Overview Page	3 Days	e,f
i	Testing	6 Days	g,h
j	Maintenance-Food Verification	13 Days	i

Subtask Division

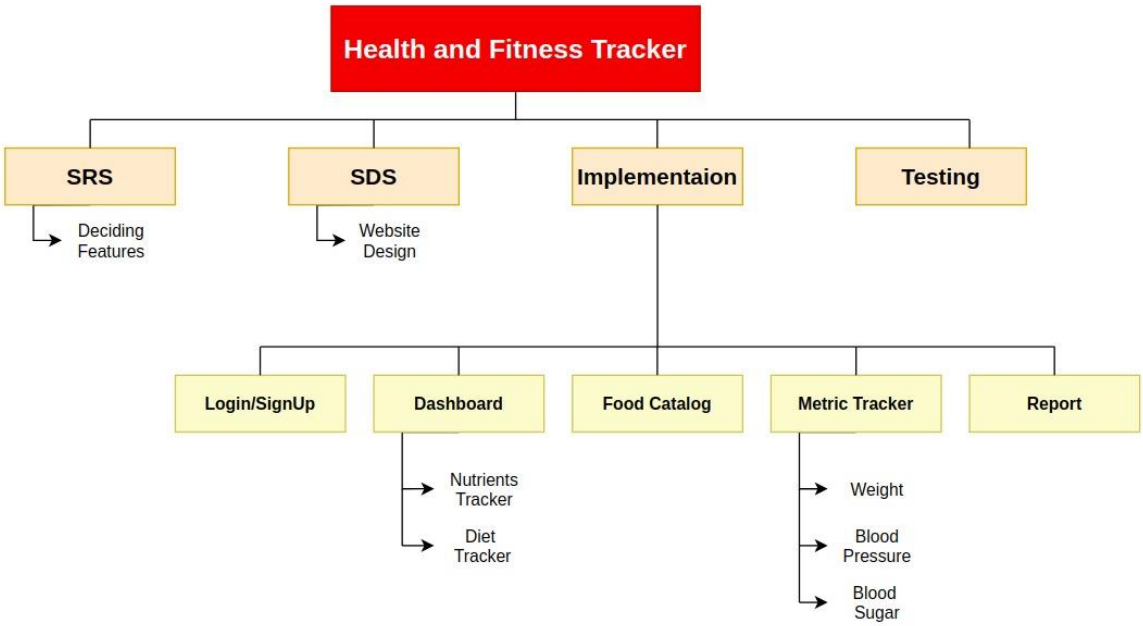


PERT Chart

SR. NO.	SUBTASK	START TIME	COMPLETION TIME	CRITICALPATH
a	Scheduling Diagrams	0	3	*
b	Deciding Features	3	11	*
c	Website Design	11	16	*
d	Login, Home Page	16	19	*
e	Trackers-BP,Weight,Blood Sugar	19-21	22-24	
f	Diets Page	19	24	*
g	Calorie Tracker	24-25	26-27	
h	Overview Page	24	27	*
i	Testing	27	33	*
j	Maintenance-Food Verification	33	46	*

PERT table

4. WBS(Work Breakdown Structure) Chart



WBS Chart

3. External Interface Requirements

3.1 User Interfaces

Shall be mentioned in SDS report.

3.2 Hardware Interfaces

The supported device types include mobile phones, tablets, laptops and desktops. The website will be made responsive for mobile devices. The data of the users will be stored in JSON format on a MongoDB database. A REST API will be used to interact with the database. The Representational State Transfer uses HTTP protocol as its communication protocol.

3.3 Software Interfaces

The connection between the product and other specific software components are:

- Operating System: Windows 10/8.1/8, MacOS 10.12 (Sierra) and above, Ubuntu 16.04 and above, Android 7.0 (Nougat) and above, iOS 9.3.5 and above
- Database: MongoDB
- Libraries and Frameworks: Bootstrap, NodeJS, Express, Mongoose.
- Browser: Google Chrome 84, Safari 13.1, Edge 84, Firefox 79, Opera 69 will be recommended.
- Data items will be coming into, and out of, the system in JSON format through a REST API which uses the HTTP protocol for communication. The product needs to use the GET, POST, UPDATE and DELETE request.

3.4 Communications Interfaces

For communicating with a MongoDB database, the product will use a REST API which uses the HTTP methodologies defined by the RFC 2616 protocol for communication. REST has been preferred over SOAP (Simple Object Access Protocol) since REST uses less bandwidth, allowing for efficient internet usage.

4. System Features

4.1 System Feature 1

4.1.1 Description and Priority – Stakeholders of the product

The stakeholders of this project are the end user, the database admin, nutritionists, catalog admin.

- This product is a medium sized project.

4.1.2 Stimulus/Response Sequences

- For the login page - The user will enter his/her login details which will allow them to access their account. The user data will be retrieved and can be seen by the user. The user can now go to all other web pages and can see the statistics relevant to their health.
- For the trackers page - These trackers take input from the user regarding their blood pressure, blood sugar and weight. Upon receiving input, a POST request is sent by the web page to the database to store the data that has been sent. When the user chooses to see the trends over time, a GET request is sent to the database to get the data required.
- For the diets page - The user will choose a dish which will be added to their diet. The calorie intake of the dish shall be recorded and shall be added with the other dishes the user chooses. In case there is no such dish that the user wants, then they can add a new dish which will be examined by the team whether the data given is valid or not.
- For the calorie tracker - The dishes chosen in the dishes page should be totaled up and stored in the database. Overtime this will allow the user to see the trends of their calorie intake. This will allow the users to find out how much calories are they taking and how much of it is used. They can also put in the number of calories they lost after an exercise. This can be used to find out how many calories are unused.
- For the overview page – All the data given by the user is used to make a concise report of the current health status of the user. Any suggestions, if required, shall be shown here. The user shall navigate to the overview page through navigation bar and access should get the report.

4.1.3 Functional Requirements

REQ-1: Validation of user input takes place. If the given username and password do not match, the user will be informed of the error. The validation is carried out by comparing the given username and password with their corresponding values stored on the database.

REQ-2: REST API must be used for GET and POST requests.

REQ-3: In case any process fails the user must receive an error message mentioning the error code and error message.

REQ-4: The page must show all dishes and add the calorie values of the dishes that the user chooses.

REQ-5: In case of the dish not being available, the user should be allowed to add the dish to the catalog. In case the dish cannot be added the user must get an error code and error message.
REQ-6: The submitted dish must be examined by the team and should be given a verified status if the data given is accurate.
REQ-7: The calories given in the dish are added up and the total calorie intake for the day must be given.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Database queries should be fast. For personal information and fitness logs it should be instantaneous. For national average reports it can take longer.

Reliability

The system should be extremely reliable and have an approximate up time of 99.999%.

In the event of a crash or any other error, the System should inform the user of any problems and gracefully terminate.

Speed

The application should open and be ready to use within 10 seconds of being selected.

Ease of Use

The application should be user friendly and intuitive to use. GUIs should make their functions clear and navigation around the application should be straight forward.

5.2 Safety Requirements

The Application Shall Be maintained by the developers in unison with System updates.
The Application shall be modified by the Developers if the application is found to have a flaw or bug.
If the Application is to be changed, The Application shall go through detailed testing to determine the Reliability and Security of the Application.

Improving the Database:
1. With evolution of the app and usage of the app by users data base will be re-mapped.
2. Users will give rating to the exercises and based on the ratings the remapping will be done. This helps in improving the performance of the app with the usage.

5.3 Security Requirements

Privacy

All data retained by the system will be stored in accordance with the Data Protection Act 1988 and the Data Protection (Amendment) Act 2003.

The System shall not disclose and personal information about the costumers.

The Application shall not grant Access to an unauthorized user.

The Application shall not communicate with any other devices or servers while in use by the user

Further, the application will give access based on secured authorization.

5.4 Software Quality Attributes

All personal information will be encrypted and saved so that it will be safe. Salting algorithms will be used to ensure safety on internet.

The product must be maintainable & responsive.

5.5 Business Rules

N/A

6. Other Requirements

- The application will be maintained by the developers.
- Database queries must be fast.
- Application should be user friendly and intuitive to use.
- No personal data of the users is supposed to be released
- The software used for making this application are NodeJS, HTML, Bootstrap CSS and MongoDB.

Appendix A: Glossary

GUI Graphical User Interface

SDLC System Development Life Cycle

ISD Information System Development

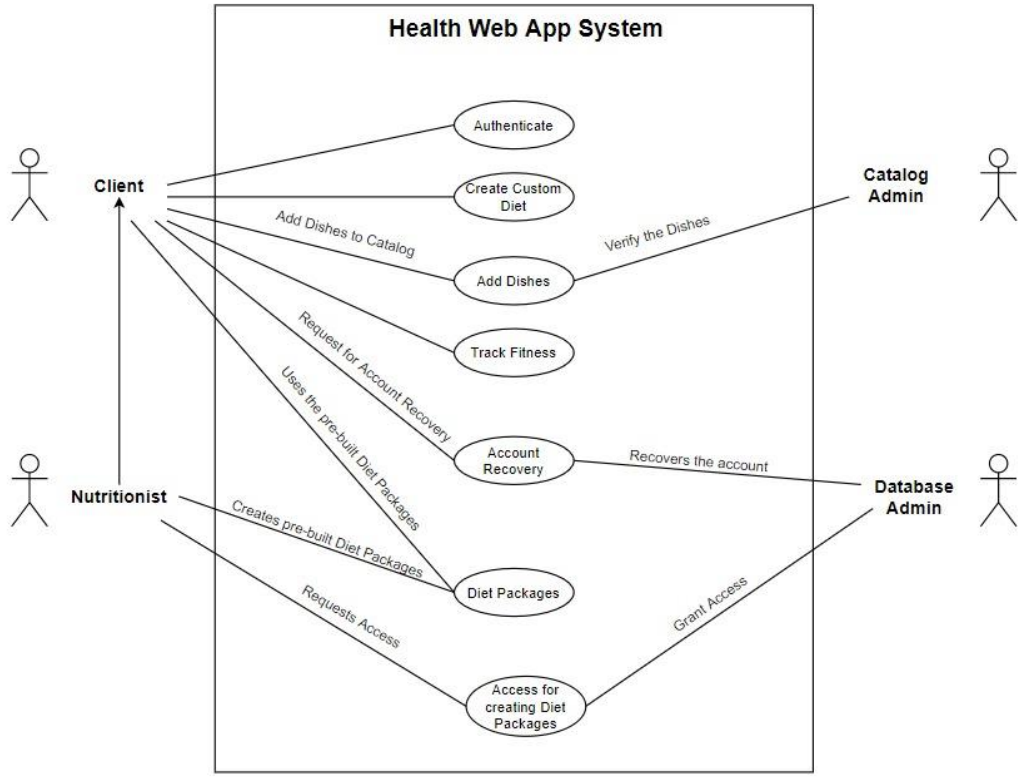
UI User Interface

PERT Program Evaluation Review Technique

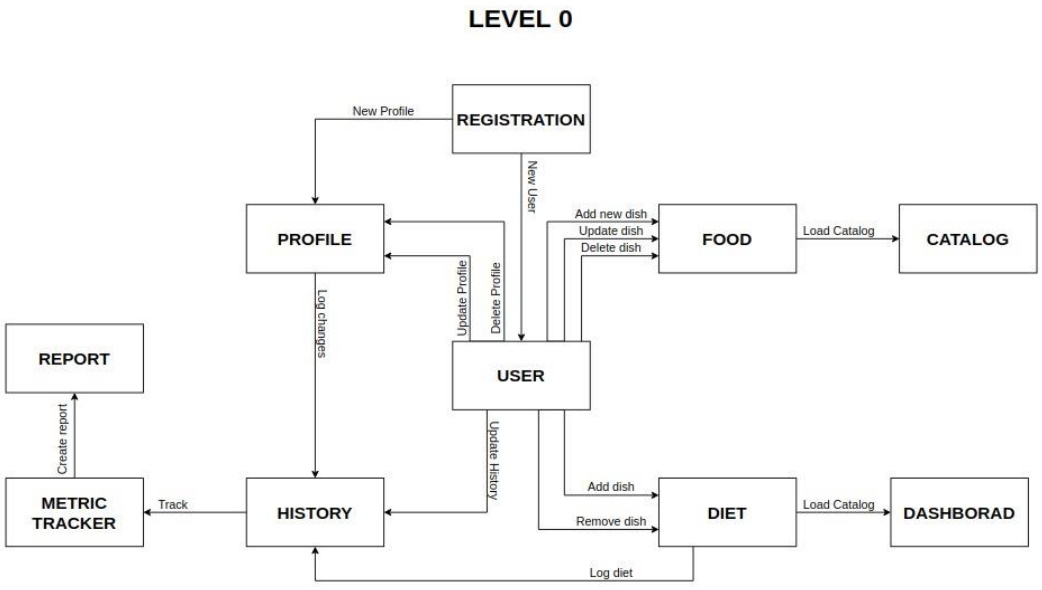
SOAP Simple Object Access Protocol

REST Representational State Transfer

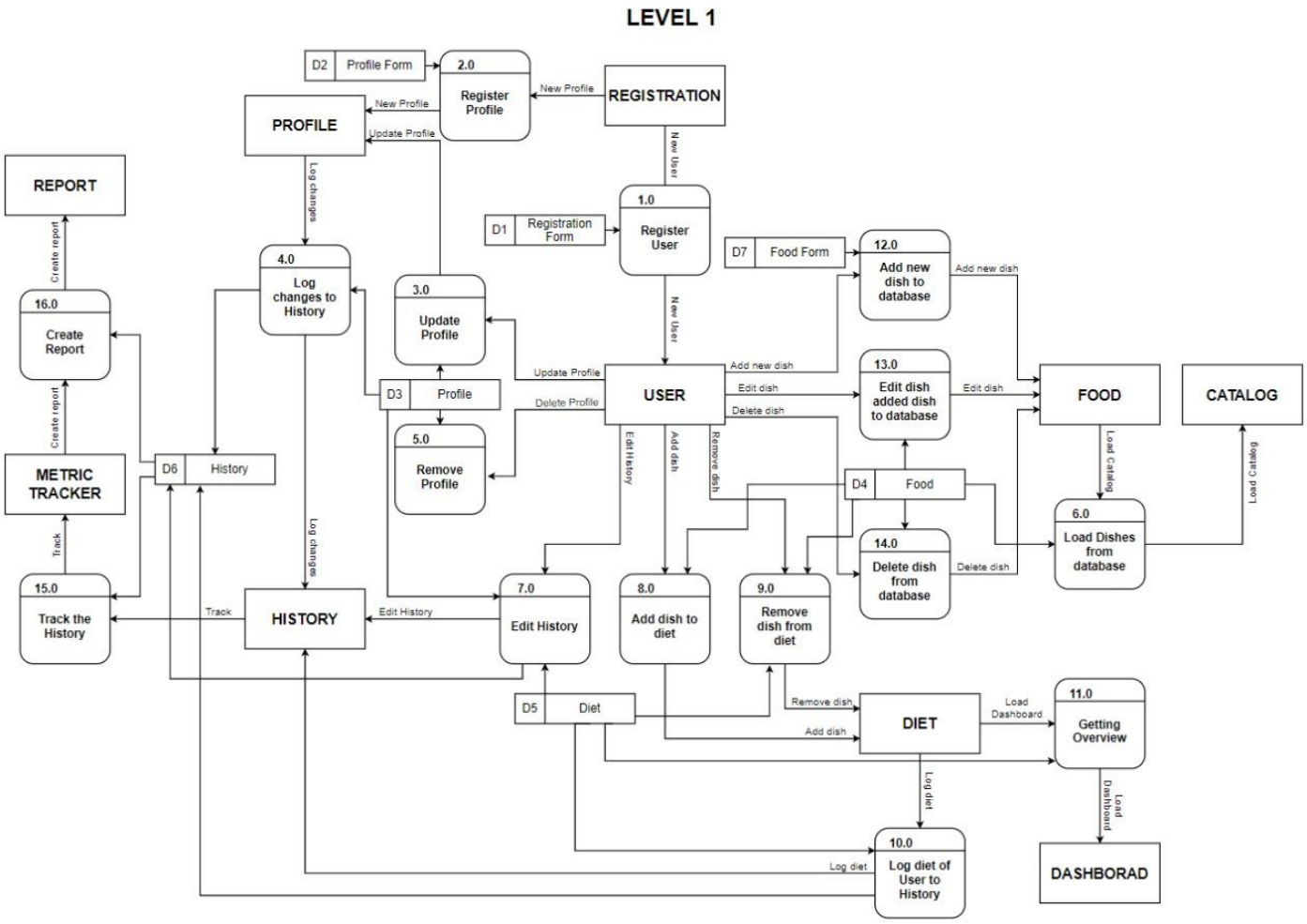
Appendix B: Analysis Models



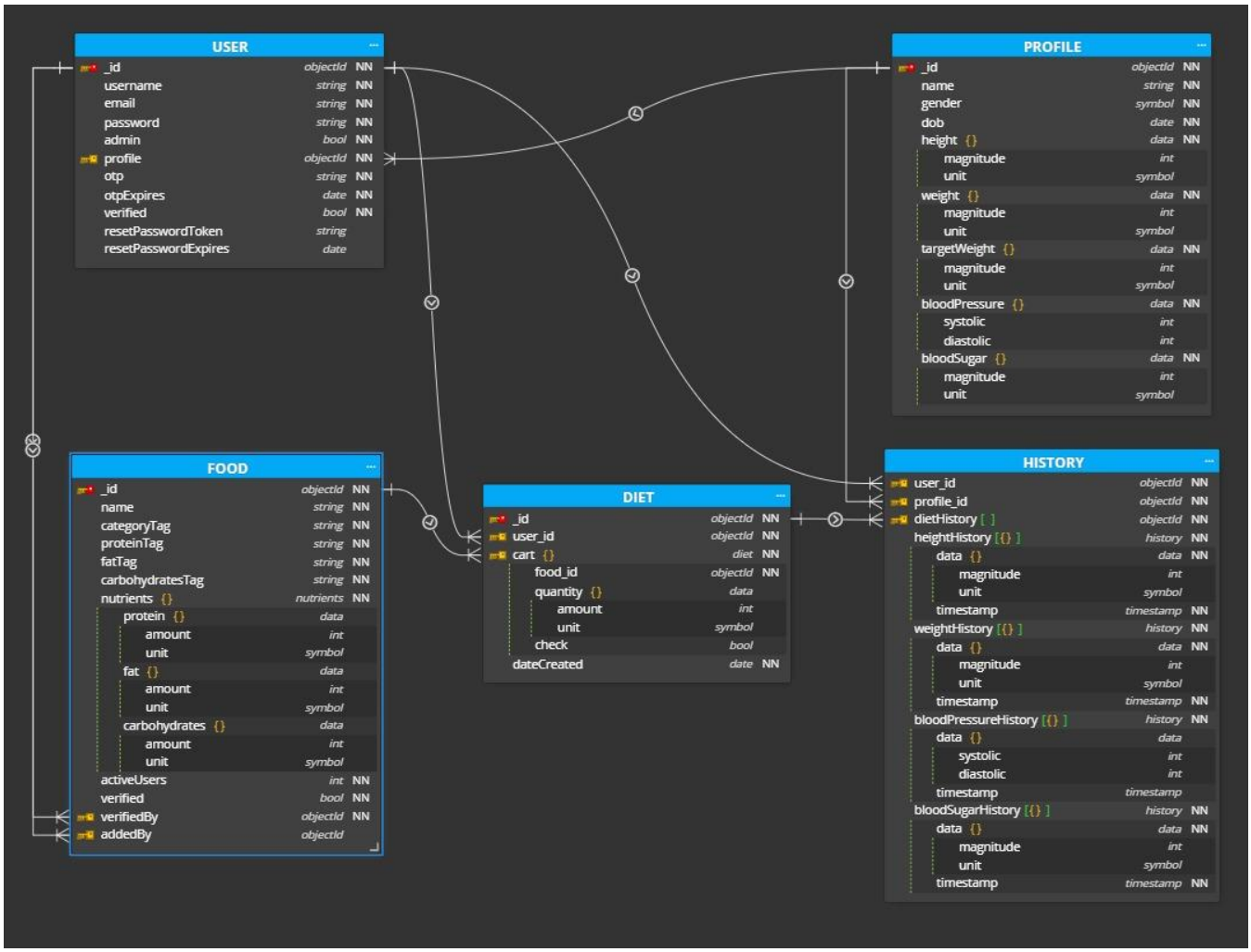
Use case diagram



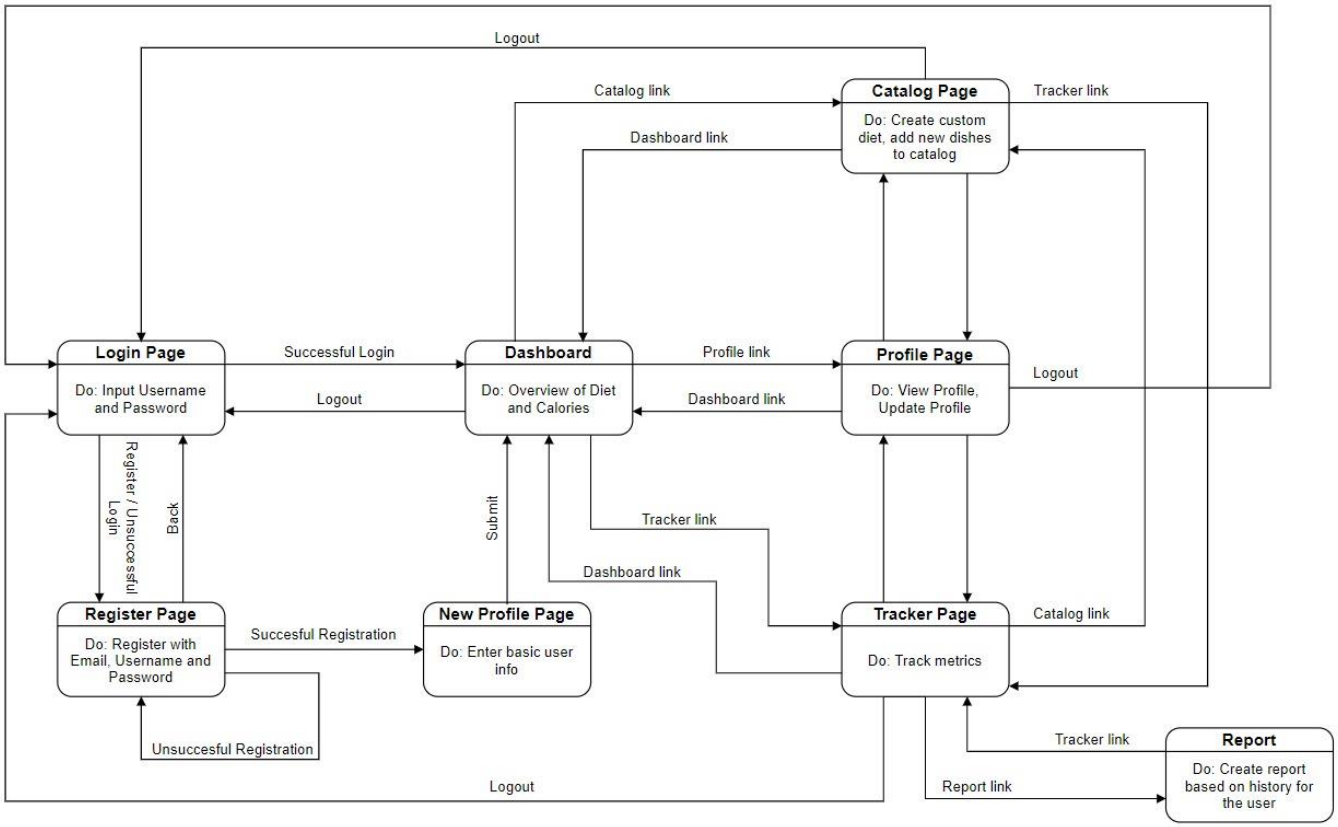
Data Flow Diagram-Level 0



Data Flow Diagram-Level 1



Entity Relationship Diagram



State Transition Diagram

Appendix C: To Be Determined List

N/A