
Software Requirements Specification

for

LifeLens: Data Visualisation for Chronic Illness Prevention

Prepared by Michael Regan 22112111

25/11/2023

Table of Contents

Table of Contents	ii
Revision History	ii
1. Introduction	1
1.1 Purpose	1
1.2 Product Scope	1
1.3 Definitions, Acronyms, and Abbreviations	1
1.5 References	1
2. Overall Description	2
2.1 Product Perspective	2
2.2 Product Functions	2
2.3 User Classes and Characteristics	4
2.4 Design and Implementation Constraints	4
2.7 Assumptions and Dependencies	4
3. Functional Requirements	5
3.1 External Actor Descriptions	5
3.2 Use Cases	5
3.2.1.1 Use Case 1.1 Create Account	5
3.2.1.2 Use Case 1.2 Log in	6
3.2.2 Use Case 2 Upload lifelog data	7
3.2.3 Use Case 3 Visualise lifelog data	7
3.2.4 Use Case 4 Corelate behaviours to chronic illness	8
3.2.5 Use Case 5 Calculate current risk of chronic illness	9
3.2.6 Use Case 6 Get current risk of chronic illness	9
3.2.7 Use Case 7 Get future activity suggestions	10
3.2.8 Use Case 8 Prompt text generation API	11
4. System Architecture Diagram	12
5. High-Level Design - Data flow diagram	13
6. Preliminary Schedule	15

1. Introduction

1.1 Purpose

This document contains the LifeLens system requirements and its analysis. Its purpose is to serve as a reference for the system design. The intended audience includes system designers, project supervisors, module coordinators and the CA326 module demonstration panel.

1.2 Product Scope

The goal of development is to produce a web app called LifeLens

LifeLens is a web application that visualises and extracts meaningful insights from users' lifelog data. It is intended that users will collect their own lifelog data and upload it to the website under their own personal account. This lifelog data will include their daily activities and metadata collected from biometric sensors.

The system's primary goal is to make users' data clear and comprehensible to understand their health patterns. This goal is essential because a link exists between people's daily activities and chronic illnesses [1].

The secondary goal of the system is to analyse the user's lifelog data and assess their current risk of chronic illnesses. The feature can provide users with a deeper understanding of their lifelog data. The system will suggest personalising future activities, such as exercise, to prevent the chronic illnesses the user is at considerable risk of.

The tertiary goal of the system is to incorporate a text generator API so that users can further discuss health patterns they noticed from the data visualisation.

1.3 Definitions, Acronyms, and Abbreviations

BRFSS Stands for “ Behavioral Risk Factor Surveillance Survey” and is a survey conducted by the CDC.

CDC Stands for “Centre for Disease Control and Prevention” and is a United States government public agency.

Chronic Illness: A health condition or disease that is persistent or otherwise long-lasting in its effects or a disease that comes with time.

Lifelog: A lifelog is a personal record of one's daily life in varying detail.

1.4 References

[1] Willett, W. C. Balancing Life-Style and Genomics Research for Disease Prevention. Science Vol. 296, 5568 (Apr. 2002), <https://www.science.org/doi/10.1126/science.1071055>

[2] Centres for Disease Control and Prevention (CDC), Behavioral Risk Factor Surveillance Survey, 2014-edition, last updated Nov. 21, 2018, <https://data.world/healthdatany/ttzx-73qb> (accessed Nov 26, 2023)

[3] S. Chung, ETRI lifelog Dataset, 2020 edition, last updated Mar. 17, 2022, https://nanum.etri.re.kr/share/list?lang=En_us (accessed Nov. 26, 2023)

2. Overall Description

2.1 Product Perspective

Lifelog is a self-contained product with two purposes:

- 1: Turn the user's lifelog data from raw data into a clear, comprehensible form through data visualisation.
- 2: Prevent chronic illness by identifying the user's current risk and suggesting behavioural changes to reduce that risk.

2.2 Product Functions

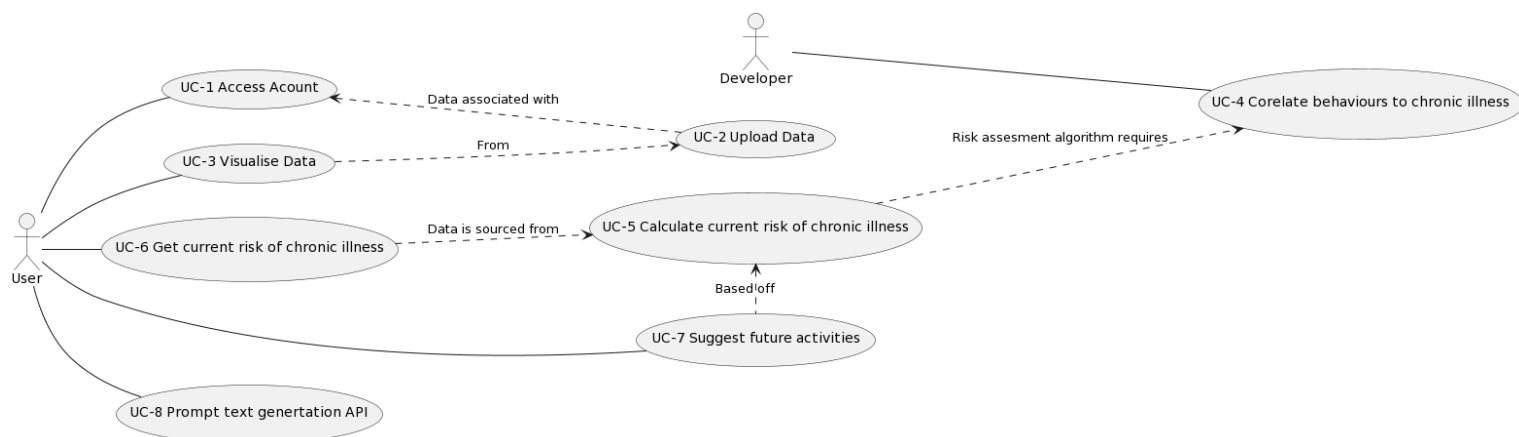


Figure 1 Context level use cases

A user wants to create an account on the LifeLens web application to visualise their lifelog data.

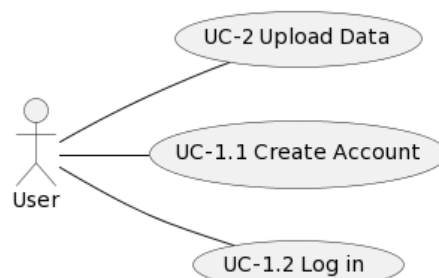


Figure 2 User account actions

Once users have logged into their account and lifelog data have been uploaded to the system; it can be visualised with various graphs and charts highlighting the frequency of daily activities and health patterns.

A Developer wants to Correlate behaviours to Chronic illness. Behaviours and chronic illnesses that are being correlated are as follows:

Chronic Illness	Behaviour
<ul style="list-style-type: none"> • Depression • Asthma • Cancer • Chronic obstructive pulmonary disease, emphysema or chronic bronchitis • Kidney Disease, • Alzheimer's • Heart attack • Stroke • Diabetes • Arthritis 	<ul style="list-style-type: none"> • Alcohol consumption • Smoking • Exercise • Sleep • Diet • Stress • Recreational activities • Job type

Now that the developer has correlated behaviour to chronic illnesses The system can provide the user with their current risk of chronic illness based on their behaviours seen in their lifelog data. Users are also offered suggested activities to reduce their risk of chronic illness.

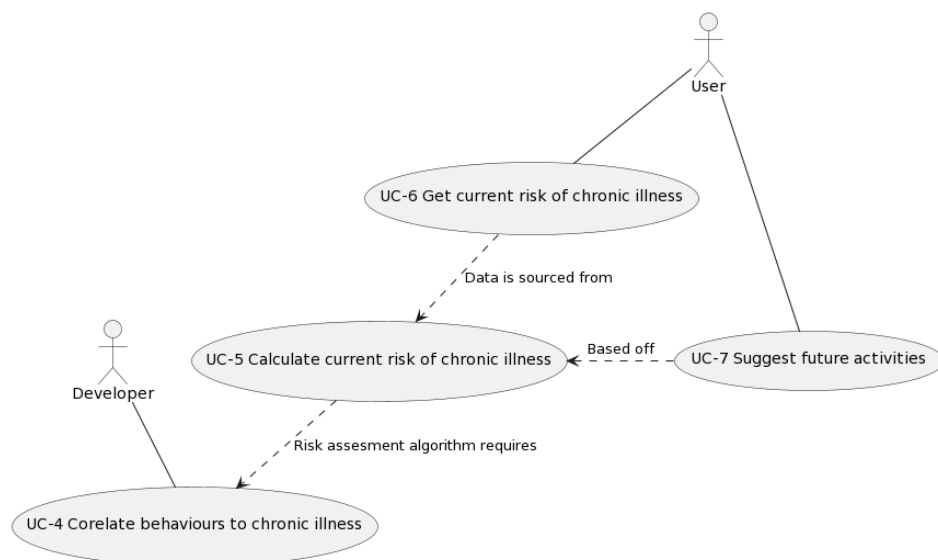


Figure 3 Chronic illness prevention use cases

For users who want to ask questions or gain further insights about their data, a chatbot will be incorporated that queries a text generator API.



Figure 4 Prompt text generation API use case

2.3 User Classes and Characteristics

The intended users of this are

1. **Health-conscious individuals.** They are frequent users who would use the full range of features on the app. Their technical experience would be low, and they are an essential user to consider when developing the app.
2. **Individuals Managing Chronic Conditions.** Frequent users who would use the data visualisation features to track health patterns. Their technical experience would be low, and they are influential users to consider during development.
3. **Researchers and Data Analysts.** The frequency of use and features used would vary based on research. Their technical experience would be high. As a user, they are unimportant to the project's development.
4. **Healthcare Professionals.** Frequency depends on interest and patients' willingness to collect lifelog data. They would use the visualisation features. They have high technical experience and a deep understanding of health patterns. Their importance when developing the app is low.

2.4 Design and Implementation Constraints

UI constraints:

Requires a frontend UI library such as React.js or Node.js.

Backend constraints:

Requires a web framework such as Django or Flask.

Visualisation Constraints:

Requires a visualisation library such as D3.js or Chart.js.

Criticality of the application

The mark of CA326 is bound to the development of this application.

Safety and security Considerations

Due to the personal nature of lifelog data, security must be a priority.

2.5 Assumptions and Dependencies

Lifelens's chronic illness risk assessment depends on the Behavioral Risk Factor Surveillance Survey data. BRFSS monitors modifiable risk behaviours alongside other factors contributing to the leading causes of morbidity and mortality. LifeLens will use this data to correlate behaviour to chronic illness for risk assessment on lifelog data [2].

Testing and demonstration are dependent on ETRI's lifelog dataset 2020 edition. This dataset consists of 30 people and has data collected from a range of biometric sensors and tracked 92 daily activities including but not limited to sleeping, medical services, walking, studying and smoking [3].

3. Functional Requirements

3.1 External Human Actor Descriptions

ACT-1	User
Description	A person who wants to access the app's functionality.
Notes:	Users are either inactive or active in use cases.

ACT-2	Developer
Description	Is an admin user that uploads human behavioral-medical data for correlating behaviour to chronic illness.
Notes:	The developer does not interface with a UI.

3.2 Use Cases

3.2.1.1 Use Case 1.1 Create Account

UC-1.1	Create Account	
Precondition	1	A novice user wants to set up an account on LifeLens.
Description	1	System indicates to the user to enter their email username and password.
	2	User enters their email, username and password.
	3a	The system throws an error if an account with the entered email is already registered.
	3b	If an account with the same username exists, the system throws an error.
	3c	If the password is unsafe, the system throws an error.
	3d	Password is encrypted, and email, username and password are saved to the database.
	4	The system sends an email confirming account creation.
	5	The user is logged in.

	6.	The home page is loaded.
Postcondition	1	A user account is created.
Expectations	1	The user will choose the appropriate username.
	2	The user will create an account with their email.
Notes	If any condition for steps 3a - 3c are met, then the system breaks from the process; else, 3d is executed	

3.2.1.2 Use Case 1.2 Login in

UC-1.2	Log-in	
Precondition	1	User wants to log into the website.
	2	User account has been created, and the user is automatically signed in.
Description	1a	The user is prompted to provide their username, email, and password.
	1b	User enters their username, email, and password.
	2	System receives username, email, and password.
	3	Database is queried.
	4a	The system throws an error if the email or username is not in the database.
	4b	If the given password does not match the one from the database, the system throws an error.
	5	The user gets access to their account.
	6	The homepage is loaded.
Postcondition	1	The user is logged in and can upload their lifelog data or access features that incorporate their already uploaded data.
Expectations	1	Users will remember their username or email and password.
Notes:	If the system logs in after an account creation, skip steps 1a-1b. If conditions from 4a-4b are met, the system breaks	

	from the process.
--	-------------------

3.2.2 Use Case 2 Upload Lifelog Data

UC-2	Upload Lifelog Data	
Precondition	1	A user wants to upload their recorded Lifelog data to the website for data visualisation and analysis.
Description	1	The System asks the user to pick a file to upload.
	2	The User chooses a file containing their lifelog data.
	3	The system checks if the file is .csv or .sas7bat.
	4	A Post request is made from the frontend to the backend with the data.
	5	The system cleans the data, handles incomplete data entries and normalises it.
	6	The system accepts the data and stores it in its database.
Postcondition	1	The User's lifelog data is stored in the system's database and can be used for data visualisation and analysis.
Expectations	1	User will collect their lifelog data.
	2	User will upload the correct file type.
	3	The user's dataset will contain the same attributes as ETRI's lifelog 2020 edition [2] .
Notes		

3.2.3 Use Case 3 Visualise lifelog data

UC-3	Visualise lifelog data	
Precondition	1a	A user logged into their account
	1b	A user has uploaded lifelog data.
	1c	A user selects a page on the website for data

		visualisation.
Description	1	A Fetch request is made from the frontend to a REST API for necessary lifelog data
	2	The backend processes the request.
	3	The request is approved.
	4	The data is sent as JSON.
	5	The JSON data is inserted into graphs and charts.
Postcondition	1	Lifelog data is visualised on the front end for users to comprehend their health patterns easily.
Expectations	1	The user has uploaded lifelog data.
Notes:	The type of graph or chart visualised will depend on the page requesting the data.	

3.2.4 Use Case 4 Corelate behaviours to chronic illness

UC-4	Corelate behaviours to chronic illness	
Precondition	1	Developer uploads behaviour-medical dataset to the system.
Description	1	Relevant data, such as lifestyle behaviours and chronic health conditions, is gathered.
	2	Data is cleaned to handle missing values, outliers, and anomalies.
	3	The data is normalised and stored in the database.
	4	Data is categorised based on different behaviours and chronic conditions.
	5	Static methods analyse the relationship between specific behaviours and chronic illnesses .
	6	Strong correlations that indicate a higher risk of certain illnesses are identified.
	7	Sensitive data is deleted from the system.
Postcondition	1	The system is aware of behaviours that increase the risk of chronic illness.

Expectations	1	The correlation will have a high degree of accuracy.
Notes:	The correlation algorithm will be simple, and we cannot account for nonce causes of chronic disease.	

3.2.5 Use Case 5 Calculate current risk of chronic illness

UC-5	Calculate current risk of chronic illness	
Precondition	1	User Successfully uploads their lifelog data to the system.
Description	1	Data is fetched from the database.
	2	Aspects of lifelog data relevant to chronic illness are identified.
	3	Risk assessment algorithm analyses the lifelog data to quantify risk.
	4	Current risk level is identified.
	5	Current risk level is stored in the database.
Postcondition	1	The system is aware of the user's current risk level of chronic illness based on their lifelog data.
Expectations	1	The user's lifelog data will be large enough for accurate risk assessment.
Notes	Activities in the data will be categorised into groups, such as physical activity.	

3.2.6 Use Case 6 Get current risk of chronic illness

UC-6	Get current risk of chronic illness	
Precondition	1a	A user logged into their account.
	1b	A user has uploaded lifelog data.
	1c	A developer has uploaded behaviour-medical data.
	1d	A user loads the web page that displays the user's current risk of chronic illness.

Description	1	A Fetch request is made from the front end to the REST API for risk assessment data
	2	The backend processes the request.
	3	The request is approved.
	4	The data is sent as JSON.
	5	The data is displayed on the UI in order of highest risk.
Postcondition	1	The user can view their current risk of chronic illness.
Expectations	1	The current risk of chronic illness is accurately calculated.
Notes		

3.2.7 Use Case 7 Get future activity suggestions

UC-7	Get future activity suggestions	
Precondition	1a	User is logged in.
	1b	User had uploaded lifelog data.
	1c	The user's current risk of chronic illness has been calculated.
	1d	User has requested to view requesting to view the page that suggests behavioural changes.
Description	1	A Fetch request is made from the front-end to a REST API for future daily activities.
	2	The request is validated by the backend.
	3	Future activities are suggested that will reduce the risk of chronic illnesses the user is most at risk of.
	4	Future activities are sent to the frontend as JSON data.
	5	The data is displayed on the UI.
Postcondition	1	The user can view activities that will reduce their risk of chronic illness.

Expectations	1	Activity suggestions should be relevant to the person's health risk.
	2	Suggestions should be applicable and feasible within the user's daily routine.
	3	Suggestions should contribute positively to the user's physical, mental and emotional health.
Notes		

3.2.8 *Prompt text generation API*

UC-8	Prompt text generation API	
Precondition	1	User is logged in.
Description	1	The user is indicated to enter a prompt to discuss any health trend they see for their data visualisation.
	2	The user enters a prompt.
	3	The frontend calls a text generator API.
	4	The response from the API is sent back and displayed on the UI.
Postcondition	1	The user gains further insight into their data.
Expectations	1	The text generator's response to the prompt will be insightful to the user.
Notes		

4. System Architecture Diagram

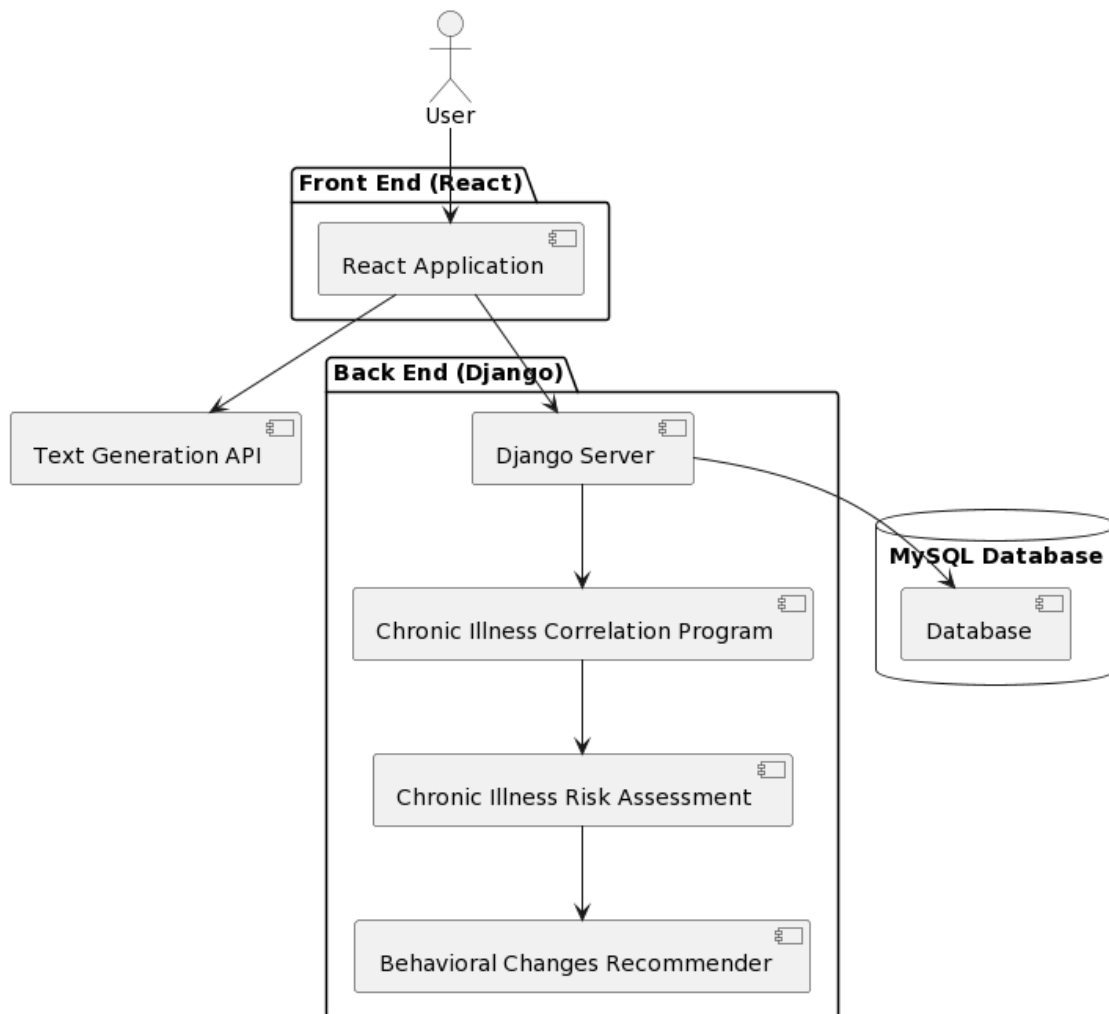


Figure 5 System Architecture Diagram

5. High-Level Design - Data flow diagram

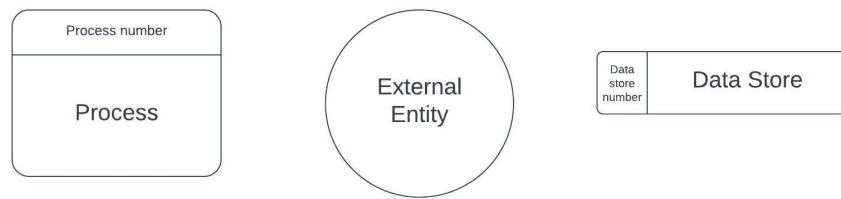


Figure 5 Data flow glossary

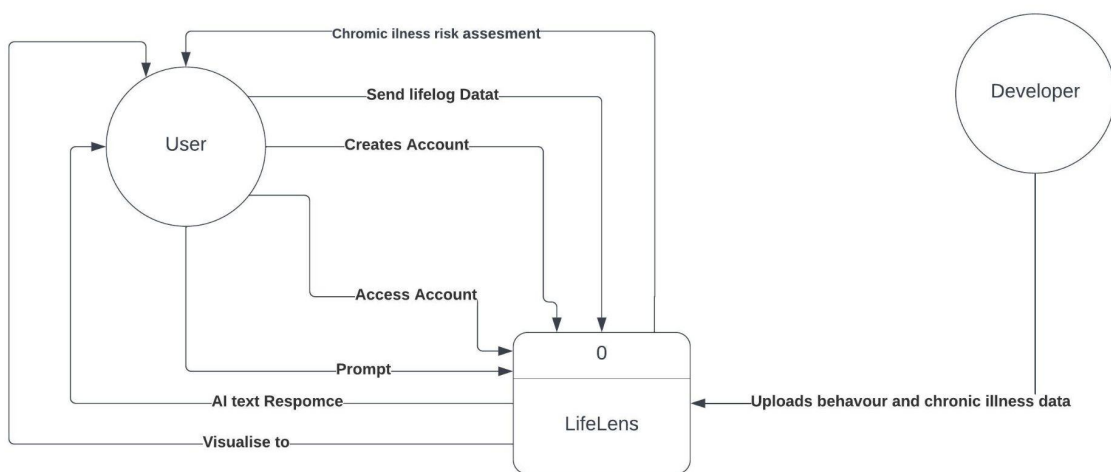


Figure 6 Data flow diagram, level 0

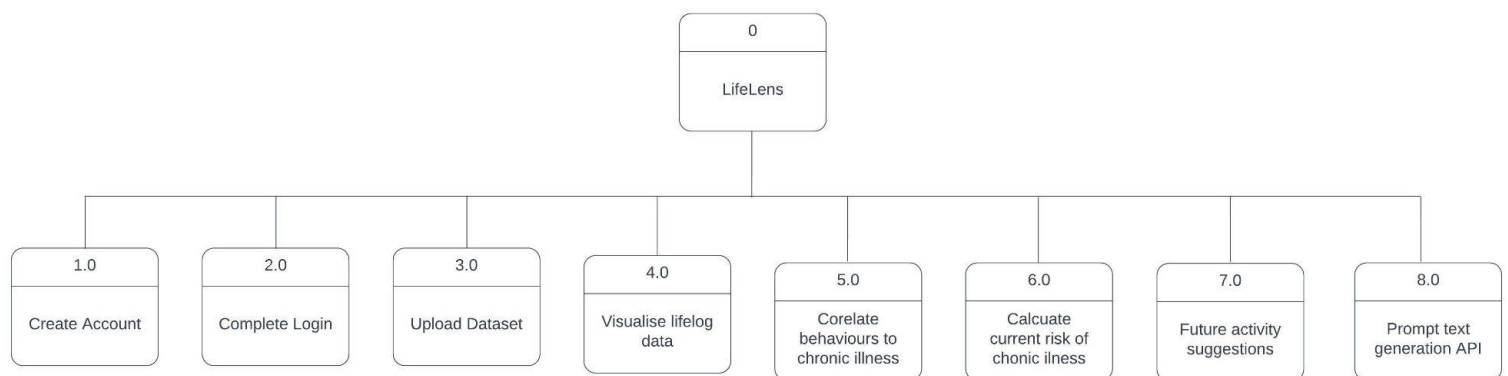


Figure 7 Data flow diagram, level 1

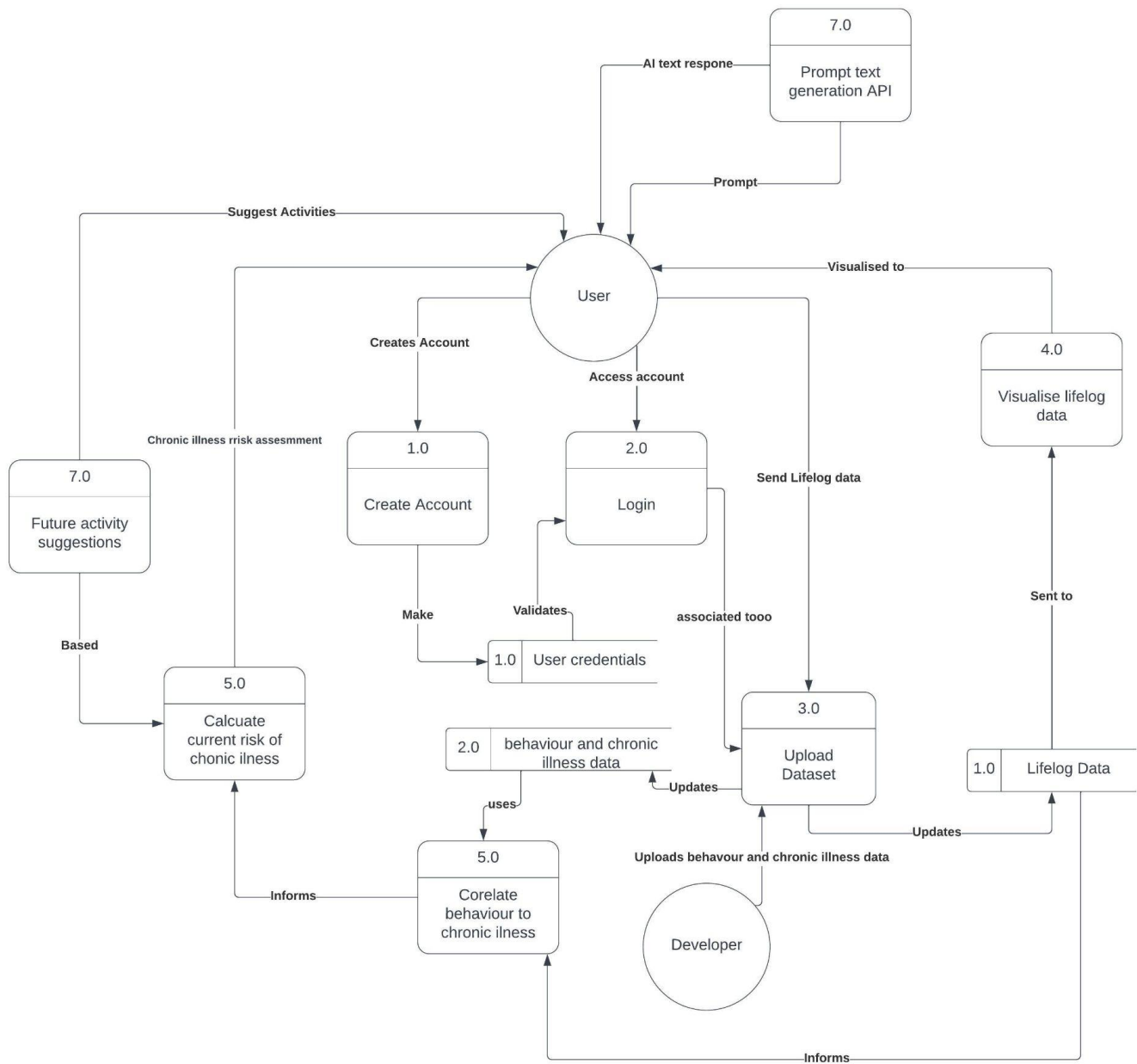


Figure 8 Data flow diagram, level 2

6. Preliminary Schedule

<i>Goal</i>	<i>Date of completion</i>
Development begins	1/1/23
Front-end + backend working, Account creation and login validation functioning	7/1/23
User's lifelog data can be normalised and stored in the database	14/1//23
Lifelog data visualisation features working	21/1/23
Correlation algorithm working	28/1/23
Risk assessment based on users lifelog data working	4/2/24
Future activity suggestion working	11/2/23
Text generator API integrated + initial UI design	17/2/23
Finished UI - design	23/2/23