(CM10251)

FitByte

Group 1

07/05/2020

Authors

Group member	Username
Varun Alvakonda	va367
Philip Bardy	pmb44
Lucas Bebell	lb2104
Robert Chambers	rcc41
Solomon Arnell-Larcombe	sal71
Tiago Assuncao Reis	tjar20
Toby Britton	tb875
Jakob Aylott	jma66

Github: https://github.com/PBardy/FitByte

FitByte

Varun Alvakonda, Philip Bardy, Lucas Bebell, Robert Chambers, Solomon Arnell-Larcombe, Tiago Assuncao Reis, Toby Britton, Jakob Aylott

1 - Abstract (Toby)

In this project we have attempted to solve the problem of fitness among university students, which is known to be a key issue. We have tried to do this through the development of a personal informatics application to track and inform the user on their current calorie intake as opposed to their recommended calorie intake. Through our agile development sprints we have researched and incorporated many gamification techniques to entice the user into continual use of the application such as leaderboards and goals. We have also ensured that the system is customisable to allow for different user preferences through the implementation of multiple different BMR calculation techniques as well as allowing for users to input in different units. Through this experiment we determined that personnel shortfall is a key risk to consider as it had a moderate impact on the project in the later stages, however we were able to manage it through scaling back the amount of features we intended to implement. Furthermore we have also considered the future of our application through the implementation of an administrative user who has access to commands that will be used to debug the system to figure out the nature of issues found, which then we may be able to potentially fix.

2 - Table of Contents

3 - Introduction

3.1 - The Problem Area

The issue that we plan to address with our software is that of student wellbeing, largely the diet aspect. It is common knowledge that university students tend to have a higher alcohol intake and consumption of fatty/processed foods, while also eating less fruit and vegetables which contain vital nutritional values that are not found in other foods. Another common issue is the eating pattern amongst university students, in a study of 80 undergraduates (Spanos, D. and Hankey, C.R. 2010) found that a quarter of students at a university in Scotland skipped breakfast on a daily basis, it has been theorized that this has lead to a higher than usual consumption of fatty/energy dense snacks later on in the day to make up for it. Results also showed that three quarters of these students used vending machines on a regular basis, eating chocolate snacks and crisps often.

These disordered eating habits (different to eating disorders) can lead to many health issues if continued later into their lives, this is supported by an abundance of research found online. Fatty and processed foods, as well as increased alcohol intake are often associated with weight gain/obesity, which is known to make you vulnerable to a multitude of health risks, this includes heart disease, strokes, diabetes, cancer, gallbladder disease, osteoarthritis, gout and breathing problems (Robert Bargar, MD, 2017). Adults who are 40% more overweight than recommended are twice as likely to face a premature death when compared to those who are at a recommended weight. Since these statistics are based on adults, students may face less risks of any major diseases. However, it is still relevant as if you consider the possibility that a student having a poor dietary lifestyle could continue this further into his adulthood as it has now become a part of their daily routine, which could mean that they could face higher risks further into their future.

Other than dietary issues a lot of students can also suffer from poor mental health caused by increased levels of stress. After a study was done on american college (university) students (Weinstein, Lawrence Laverghetta, Antonio), it was concluded that 'These results suggest that general life satisfaction, which is a factor in subjective well-being, in college students is inversely related to stress and anxiety' by a significant amount. This increased levels of stress leading to mental wellbeing problems can supposedly lead to an increased likelihood of the student dropping out of uni, which can have a significant impact on the individual's life. 'In fact there is data that show that many students who leave our university do so because the stress level is high (Weinstein, 2008)'.

3.2 - Our Solution

To tackle this problem area we have decided to develop a personal informatics system. Since the advent of modern-day computers, technology has played a major part in enabling people to track their wellbeing. Personal informatics is a field of research and development that aims to assist people collect and reflect on data about themselves (Li, Dey & Forlizzi, 2010). Collecting data allows users to analyse their daily lives and potentially implement a new regime to achieve a goal they have. For this reason, the field of personal informatics has gained popularity and attention over the last few years.

The field of personal informatics is broad, as well-being encompasses many factors in a person's life. Naci & Ioannidis (2015) stated that wellness: "refers to diverse and interconnected dimensions of physical, mental, and social well-being that extend beyond the traditional definition of health." For this reason, we have seen an abundance of products and services intended to help a person self-improve, each focusing on a particular aspect and implementing their own unique approach.

In the realm of physical wellbeing, the personal information that is collected can relate to a person's lifestyle choices such as diet, sleep, and exercise. Wearable fitness trackers such as the Fitbit Charge 3 have sensors that can sense physical events and automatically record these events into data. These physical events can be heart rate, footsteps, distance travelled, waking up, flights of stairs climbed, etc (Song, 2018). The information collected can help a person monitor their activity and implement a lifestyle change to help them achieve a health-related goal (e.g. lose weight, sleep longer).

Mental wellbeing is described as a state in which the individual realizes his or her own abilities, can cope with the normal stresses of life, can work productively, and is able to make a contribution to their community (WHO, 2004). The data collected by the software systems which are designed to aid a person improve their mental health can be a quantitative measure of a person's mood. Data can be collected by having a user complete a periodic survey based on their emotions. The Warwick-Edinburgh Mental Wellbeing Scales enable the measuring of mental wellbeing in the general population. The scale consists of five response categories, summed to provide a single score. In a software system, this score can be used to track a person's mood bidaily.

In regard to social wellbeing, personal informatic systems typically aim to encourage positive social behaviour. One major focus is in reducing screen time. Excessive time spent looking at screens has been known to harm children and young people's psychosocial health (Stiglic & Viner, 2019). The screen time feature present in iOS devices is a personal informatic system intended on offering the user insights into their screen time and offers settings that enable the user to take action (Fitzpatrick, 2018).

Users can encounter barriers when using a personal informatic system. It is thus important as software engineers that we understand these barriers. Most users got through five stages using a personal software system. These five stages, as proposed by Li, Dey & Forlizzi (2010) are: preparation, collection, integration, reflection, and action. The first two stages are of particular interest. The first stage involves the user learning about what information is collected by the software system and how it will be used. A major barrier is not knowing what is being tracked and potentially not having information needs met. It is thus vital that we inform the user at the very start what kind of information we will be collecting and how we will help them reflect on it. The second stage, collecting data from the user, can potentially be disrupted if the manner of collecting data is inconvenient to the user. To mitigate this barrier, we must conduct requirements gathering such as a survey to conclude what way of collecting data our target users would prefer.

When designing a personal informatics system, it is crucial that a person's information is visualised in a way that suits them. While evaluating user requirements, Rapp et al. (2018, p.343) noted that there were two reasons why some users became less engaged in exploring their data. Some said that the way data was visualised was too difficult to understand, while others complained that the data collected lacked detail. This finding stresses the importance in making a personal informatics system that is flexible and accommodates the diverse needs of its user base. Rapp et al. (2018, p.343) suggests that to meet the needs of different users, at least two different modes of visualisation should be offered, "one more direct and intuitive, the other one more detailed and comprehensive." We will consider these findings when establishing the requirements for our system.

Our end goal with this system is to provide users with a tool which can be used to better understand their lifestyles in a couple of ways. This will then hopefully inspire them to make healthy alterations to eating and exercise patterns and other potential self improvement methods, reducing stress/improving health and improving overall physical and mental wellbeing.

4 - AGILE Process Planning & Management

4.1 - Sprint Planning

During our initial group meetings, as evidenced by the group meeting minutes (See 10.2), we planned that, after the initial design phase, we would divide our project into three sprints Sprint 1 would last 3 weeks and would attempt to implement the high priority features from the backlog, focussing on database, account and profile creation as well as basic data input to allow a more fleshed out version in Sprint 2. Sprint 2 would last 4 weeks and implement all other features of medium or low priority that did not constitute 'gamification'. Sprint 3 would last 1-2 weeks and implement additional 'gamification' features derived from our research (See 10.3.2). To prioritise, we used our features backlog, looking at the requirements which most other requirements depended upon and prioritised these. As part of our discussion of risk management we also agreed that the final sprint, as it was not implementing essential features, could be scrapped if our developers did not have enough time before the deadline.

4.2 - Sprint Tracking

After developing our software system requirements specification we created a product feature backlog which contained the description of features and their associated priority which needed to be implemented over the course of our three sprints. These were copied into Github's Kanban project board. We then used this board to track the completion and state of tasks during the sprints, Each sprint had a dedicated column for its backlog and three further state's; tasks chosen by developers to implement, tasks which were partially completed and tasks which were fully completed/shippable. Evidence of this can be seen in Appendix 10.5.

In addition to the project board, we tracked progress routinely with meetings with our scrum master. Scrum meetings occurred mid-sprint to evaluate the progress made on current features. Members of the development team were also queried about their estimates for the completion of certain tasks. As sprints were short and we had relatively few developers we did not see fit to produce a sprint burndown chart. At the end of each sprint we also conducted a sprint review with our scrum master, which entailed a product demo which exemplified the working features of the product.

4.3 - Developing Requirements (Toby)

Initially, we collected a large amount of both functional and non-functional requirements, which can be seen in 5.5, these were created through research done on the domain as well as data results from surveying our stakeholders, see 10.3. We then developed these into a sprint backlog which we used to create certain tasks, depending on the focus of the sprint and priority of the task, and added them to the list of sprint tasks for the sprint. Over the course of the sprints, if we found a necessity for further requirements then they were developed into sprint tasks and added to the sprint backlog to be implemented upon during further sprints.

4.4 - Addressing Project Risks

We used Boehm's list of the 10 most likely sources of risk in managing our project, in particular we found major sources of risk in personnel shortfalls, developing wrong user interfaces and gold plating, which are explained below.

Personnel shortfalls were assessed to be a high risk when we created our initial risk assessment, and throughout the duration of this project this has proved to be the case. This was evidenced by the lack of participation in meetings in the later weeks of the project. To address this we discussed our mitigation plan with our scrum master (over the course of several scrum meetings), which involved scaling back the amount of features we intended to implement, and providing a slightly longer timescale than originally planned. We reallocated roles to the active members to ensure that the range of tasks could still be carried out

Developing the “wrong” user interface came about as programming commenced. We found that instead of requiring the user to enter letters which would change each time, the use of numbers corresponding to a printed list was the easiest to use. This came about in sprint 2 as the number of menus grew, here agile methodology was beneficial, the requirements changed as we identified improvements and these improvements were implemented.

Gold plating was another problem encountered. We identified several cases in scrum meetings, especially in sprint 2. Initially the menus provided options beyond the scope of the project including salt intake and fibre intake. Although these are components of a balanced and healthy diet, they were beyond the scope of our project domain, which was primarily calorie intake, weight tracking and exercise tracking. These were identified as gold plating and were removed, simplifying the ongoing project implementation.

4.5 - The Evolution of Our System

Our research constituted some primary data collection in the form of a survey (See 10.3.1) which interviewed our potential user base - university students, and from academic papers on personal informatics systems and health. From the survey responses we gained a better understanding of how the users would like to add data and goals into the system; mainly that the data should be inputted and outputted in units that they're most familiar with, and that they'd be more motivated to set goals if they were gamified (e.g. through streaks). A study that was done into features which improve user retention of personal informatics systems (See 10.3.2) supported the idea of gamification, from which we derived our requirements for a leaderboard and badges (for achieving certain numbers of goals).

During our initial brainstorming we considered adding features which could track exercise data, and features which could track health data. Research we conducted into exercise recovery times (See 10.3.3), however, concluded that a feature based around this may be more difficult to implement and we decided, though our group meetings, that attempting to implement a feature we might have not had the expertise to do was too risky. Furthermore, we had conducted a considerably larger amount of research into health data and adopted the need for calorie tracking into our requirements. In addition to this we included the requirement to display BMR from an online calculator (See 5.1) which we discovered during the research of calorie measuring equations.

5 - Software Requirements Specification

5.1 - Methods of Establishing System Requirements

Building upon the initial requirements set out by the specification, we set out to gather more requirements that were specific to the system we were creating. To gather our requirements we had decided to review academic literature based on our domain and also to conduct a survey to get an idea of any specific features.

One of the aims of our system is to give each user a calorie intake goal that is specific to them. In order to express this as a set of requirements we had to conduct research on the existing methods of calculating daily calorie intake. Upon reviewing an article in the Journal of Science & Medicine in Sport by Hasson (2011), we learnt that the basal metabolic rate (BMR) represents the energy needed to sustain the human body. We also learnt that amongst all the equations present the Mifflin–St. Jeor equation is more likely to accurately predict a person's BMR. Calculating a person's BMR requires knowing a person's weight, height, sex, and age, thus resulting in us adding the collection of this data to our requirements~~.~~

Further reading of academic literature led us to learning about the impact and importance of gamification. Rapp (2018) described gamification as 'the use of "game design elements in non-game contexts"'. The primary reason to incorporate game elements in personal informatic systems is that instruments used to measure physiological data 'require a considerable effort from their users, in terms of time, compliance, and long-term engagement', which is difficult to obtain when the user is not fully dedicated to the 'quantified-self' lifestyle. For this reason, we have included some form of gamification as a non-functional requirement.

The responses from our survey also helped us gather requirements and consider how to implement them. Our survey consisted of 11 questions and received 27 responses. Requirements we have decided adopt from our survey include: offering a variety of units to input data as, a weekly progress report, a daily score indicating how close a user is to their target, a daily streak, a negative streak, being able to amend data from prior days, making our system available on multiple platforms (Windows, MacOS, Linux). One feature that proved to be controversial from our survey was the negative streak, 11 respondents indicated they would not like a negative streak. Therefore, we have made it a requirement that a negative streak is optional.

5.2 - Our System Domain

For our personal informatics system, we focused on the application of personal informatics within the domains of fitness and nutrition. Fitness and nutrition have been the focus of many personal informatics systems in the past. Examples include fitness trackers (e.g. Fitbit Charge 3) and nutrition-tracking applications (e.g. Cronometer).

As mentioned before the goal of any personal informatics system is to help people collect and reflect on personal data (Li, Dey & Forlizzi, 2010). Within the domain of fitness and nutrition, personal data can include physical activity (e.g. number of steps, distance ran, calories burnt), physiological data (e.g. heart rate, blood pressure), and nutritional data (e.g. caloric intake, protein consumed).

Systems within this domain use different methods to collect personal data. The Fitbit Charge 3 employs several sensors capable of tracking the user's movements and heart rate. The data is collected automatically with the user only having to manually input some data at the beginning of use such as: height, weight, etc. Other systems depend entirely on the user manually inputting data, such as Cronometer. Cronometer allows users to input food items and exercise manually and provides a summary of a person's caloric intake and daily caloric limits based on the user's BMI (Fadhil, 2019). When comparing the two different methods of data collection, the fitness tracker has the advantage of not requiring as much user engagement as Cronometer does.

5.3 - User Profiles (Toby)

Our system is designed to be used by two different types of users, a normal user and an admin/software technician. The normal user is the user profile of a person who will use the software regularly to input and track their daily intake and will have the following use cases: log in to profile, create new profile, edit current profile, view data, input data, calculate data values, view leaderboard, display trends, create goal, and check goal progress.

As the normal user will be a human we will need to take ethics into account, so, for this reason, we will not share or manipulate the user's data without their consent. We will also

need to make sure their data is secure so we will use a password system to only allow the user of a profile to access the data stored in that profile.
The admin, however, will be a special type of user who will have access to a special use case of ‘debug’ which will be used to maintain the upkeep of the system and check that everything is working correctly. We will use passwords for this also to prevent not admins from accessing the debug feature.

5.4 - Use Cases and Scenarios

Actor	**Use Case**	**Initial Assumptions**	**Normal Flow**	**Alternative Flow**	**On Completion**
User	Log in to account	One or more accounts exist in database User has input method (assume for all) User hasn’t been blocked from previous failed attempts	1. User prompted for username 2. User inputs username 3. Account with username exists and the user is prompted for the password 4. User inputs password 5. Password is correct for username 6. User is logged in to account	3. No account exists with username, error displayed and user prompted for a different username 5.a. Password does not match the hashed password for username, the user is given 3 more attempts 5.a. User inputs correct password and normal flow continues 5.b. User fails password check 3 times and is blocked from logging in	User is logged into account
User	Create new profile (edit current profile very similar)	Space available in the database for a new profile	1. User requests to create new profile 2. User prompted for username 3. User inputs username which fits format requirements 4. User is prompted for new password 5. User inputs password that is valid for the hash function 6. User is prompted for personal information such as weight, height, gender, DOB	3. Username does not fit format requirements, user prompted to try again and requirements displayed 5. Password is not valid for hash function so cannot be stored, user prompted to try again and requirements displayed 7.a. User inputs all information but with invalid formats, user prompted which fields to re-input and requirements displayed 7.b. User does not fill all fields, the user is prompted to fill in all information and fields	A new profile is in the database with basic user data has been created

			7.User inputs all information with valid formats 8. Profile created and data stored in the database	highlighted in some way.	
User	Input Data	User is logged into account	1. User selects add data option 2. User selects data type from list 3. User inputs data in correct format 4. Data is added to the database	3.User inputs data in incorrect format e.g. a future data. Return an error message explaining the correct format.	New data is added in the database which can later be viewed or used to calculate values User can select new option
User	Calculate data values	User is logged into an account	1. User requests to calculate data value 2. Database queried for data 3. Necessary data is returned 4. Data values are calculated using the relevant algorithm 5. Data value displayed to the user	1. User is not in the correct menu to select this option, the error message displayed saying invalid input 3. Not all data required exists in the database, the user is informed that there is insufficient data for the command 4. Data values cannot be calculated with provided data, the user is informed that data has been wrongly input, terminate execution	Data has been displayed to the user in a clear way User can now select other options
User	View data	User is logged into account	1. User requests to see data that they have previously input 2. Software fetches entries from the past 3. Previous data from current account is displayed to the user	2. There are no entries to display to the user. Return error telling user data must first be input to view.	User can clearly see any data they have previously input User can select new option
User	View leaderboard	User is logged into account	1. User requests to view leaderboard 2. Software fetches data on goals from other	3. There are no other users with data for leaderboard. Empty leaderboard will be displayed with only user	User can clearly see the leaderboard and their

			accounts(no personal data) 3. Data is displayed and the user is inserted into leaderboard to view position	in it.	position in comparison to other accounts User can select new option
User	Display trends	User is logged in to an account Previously input values are validated format	1. User requests to display recent trends 2. Database queried for data 3. Database returns necessary data 4. Software generates graph using data 5. Graph and data is displayed on the screen to the user	3. Insufficient data to return (user hasn't input anything yet). Return invalid or generate empty graph, user infers reasoning	User can clearly see trends of recent activity User can select new option
User	Create goal	User is logged into account	1. User requests to create a goal 2. Software checks for space in database 3. User creates goal with specified end 4. Software adds goal to database	2. No space available in the database.Inform user that maximum set goals have been reached.	New goal is added to the database to be tracked against new input data User can select new option
User	View goal progress	User is logged into account Account has at least one goal tracking	1. User requests to view goals 2. Database queried for current goals and data added since goal start 3. Software calculates goal progress 4. Goal progress displayed to user	2. No goal progress has been made to calculate, display no progress	User can clearly see how much of their goal they have completed. User can select new option
Admin	Debug	User is identified as administrator with password	1. Admin requests software to debug 2. Software runs as it would with ordinary user 3. Software returns any		Admin can see any issues with code or unexpected results from software

			unexpected results, such as alternate pathways from previous scenarios or other runtime program errors		running.

5.5 - Requirements (Toby, Robert and Philip)

5.5.1 - Non-Functional Requirements

1	**Requirement name: Editing inputs**	**Author: Toby Britton**
	Description: the user will be able to change anything they input, in case they make a mistake.	Priority: High Dependencies: None
2	**Requirement name: Performance**	**Author: Toby Britton**
	Description: the application's response time must never exceed 1 second.	Priority: Medium Dependencies: None
3	**Requirement name: Portability**	**Author: Toby Britton**
	Description: the application must run on Microsoft Windows, MacOS and Linux.	Priority: Medium Dependencies: None
4	**Requirement name: Efficiency**	**Author: Toby Britton**
	Description: the user should be able to access all parts of the application within a small number of inputs	Priority: Medium Dependencies: None
5	**Requirement name: Security**	**Author: Robert Chambers**
	Description: secure data should not be extractable from the application.	Priority: High Dependencies: None Sources: ACM Code of Conduct 2.9
6	**Requirement name: Privacy**	**Author: Robert Chambers**
	Description: potentially sensitive data should not be made visible between users	Priority: High Dependencies: FR5: Security Sources: ACM Code of Conduct 1.6 and 2.9
7	**Requirement name: Validation of Input**	**Author: Robert Chambers**
	Description: the system should check user's input and request re-entry if it does not make sense	Priority: High Dependencies: None
8	**Requirement name: Informatics feedback**	**Author: Robert Chambers**
	Description: the system should provide some form of feedback on how the user should positively change their behaviour	Priority: High Dependencies: None

5.5.2 - Functional Requirements

1	**Requirement group name: Accounts**	**Author: Robert Chambers**
	Description: The program must allow creation of an account, which has a username and password associated	Priority: High
1.1	**Requirement name: Enter account username**	**Author: Robert Chambers**
	Description: upon the initiation of account creation the program must display a prompt requesting the a username, take text input and store this input locally	Priority: High Dependencies: None
1.2	**Requirement name: Enter account password**	**Author: Robert Chambers**
	Description: after a user enters a username, the program must present a prompt requesting a password, hash the entered text and store it in the database with the username. The password must be validated to be over 7 characters long	Priority: High Sources: FR5: Security, FR6: Validation of Inputs
1.3	**Requirement name: Database store**	**Author: Robert Chambers**
	Description: After local storage of username and password, the system stores these in the database	Priority: High Dependencies: 7.1
1.4	**Requirement name: Login**	**Author: Robert Chambers**
	Description: the system must prompt for a username and password, the password is then hashed and compared to the value in the database, if match then the system logs the user in, if not then repeat prompt	Priority: High Dependencies: 1.1, 1.2, 1.3, 1.4 Sources: FR5: Security
2	**Requirement group name: Profile creation**	**Author: Robert Chambers**
	Description: the program must allow the user to create a profile for their account	Priority: high
2.1	**Requirement name: Enter profile DOB**	**Author: Robert Chambers**
	Description: after a password is entered from account creation, the program must display a series of prompts asking for day, month, year, and take text input for each and store the result. Each value must be validated: 1<= x <= 31, 1 <= y <= 12, current year - 120 <= x <= current year respectively	Priority: High Dependencies: None Sources: FR7: Validation
2.2	**Requirement name: Enter profile weight**	**Author: Robert Chambers**
	Description: after date of birth is stored, the program must present a prompt asking first for units (kilograms or pounds) then for a weight value, the program then stores this value. Value must be validated to be a positive value.	Priority: High Dependencies: None Sources: FR7: Validation
2.3	**Requirement name: Enter profile height**	**Author: Robert Chambers**
	Description: after weight is stored, the program must display a prompt asking for units then a height value, and take text input and store the result. Value must be validated to be a positive value.	Priority: High Dependencies: None Sources: FR7: Validation

2.4	**Requirement name: Enter profile gender**	**Author: Robert Chambers**
	Description: after height is stored, the program must display a prompt asking the user for their gender “M or F” and take text input and store the result. Input must be validated to be either “M” of “F”	Priority: High Dependencies: None Sources: FR7: Validation
2.5	**Requirement name: Enter current activity level**	**Author: Robert Chambers**
	Description: after storing gender the program must display a prompt asking the user to select their activity level: “1. no activity 2. little activity 3.moderate activity, 4.high activity, 5.extreme activity”, take text input and store the result. Input must be validated 1<=x<=5	Priority: High Dependencies: None Sources: FR7: Validation
2.6	**Requirement name: Multiple profiles**	**Author: Toby Britton**
	Description: the application will allow the user to create and switch between multiple profiles with different data. The program must initiate new account and profile creation from the menu.	Priority: Low Dependencies: 2.1, 2.2, 2.3, 2.4, 2.5, 2.6
2.7	**Requirement name: Editing profile**	**Author: Toby Britton**
	Description: the user will be able to change all the information associated with their profile.	Priority: Medium Dependencies: 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 7.1 Sources: FR 1: Editing Input
2.8	**Requirement name: Database store**	**Author: Robert Chambers**
	Description: upon local store of all profile information, the information is stored in the database	Priority: High Dependencies: 7.1
3	**Requirement group name: Data Analytics**	**Author: Robert Chambers**
	Description: from the collected data the application should be able to calculate various health values.	Priority: High
3.1	**Requirement name: calculation of BMR**	**Author: Robert Chambers**
	Description: the application must be able to calculate a person's BMR using the Mifflin-St Jeor Equation. The application gets gender, weight, activity level from database, applies the equation and prints the result	Priority: High Dependencies: 2.8, 5, 7.1 Sources: FR8: Informatics Feedback
3.2	**Requirement name: Data comparison**	**Author: Toby Britton**
	Description: the application must compare the user’s BMR to their calorie intake to see if they met their goal, printing the result as text	Priority: Medium Dependencies: 3.1, 5.1, 7.1 Sources: FR8: Informatics Feedback
3.3	**Requirement name: Data access**	**Author: Philip Bardy**
	Description: the user must be able to access all the raw data used in analytics. This data should be presented as a table when this option is selected from the menu	Priority: Medium Dependencies: 5.1, 5.2, 5.3, 7.1
3.3.1	**Requirement name: Data sorting**	**Author: Philip Bardy**
	Description: the program should provide a prompt allowing the user to sort	Priority: Low

	their data in table for: alphabetical, chronological, ascending order (by value), descending order (by value)	Dependencies: 3.3, 7.1
3.4	**Requirement name: Trend identification**	**Author: Philip Bardy**
	Description: the program must print entered data: weight, calories, activity, in graph form	Priority: Medium Dependencies: 3.3, 7.1
4	**Requirement group name: Gamification**	**Author: Robert Chambers**
	Description: the application may contain gamification elements which are designed to encourage repeated, if not daily, use of the application and which motivate the user to reach their self set goals.	Priority: Medium
4.1	**Requirement name: Positive streaks**	**Author: Robert Chambers**
	Description: days for which the user meets their goal adds to a streak counter variable, stored in the database	Priority: Medium Dependencies: None Sources: FR8: Informatics Feedback
4.2	**Requirement name: Leaderboard**	**Author: Robert Chambers**
	Description: the program should allow users to compare their goal completion against other users through a leaderboard. The values of all users' goal completions are fetched from the database and displayed to the user. Note that this is for privacy, the weights etc are not fetched as users are unlikely to want others seeing this	Priority: Low Dependencies: 7.1 Sources: FR8: Informatics Feedback, FR6: Privacy
4.2.1.	**Requirement name: Leaderboard ranking**	**Author: Robert Chambers**
	Description: the leaderboard should contain a column which indicates the rank of data entry it displays, the lowest values (best) being the users with the highest number of completed goals	Priority: Low Dependencies: 7.1 Sources: FR8: Informatics Feedback
4.2.2	**Requirement name: Leaderboard row**	**Author: Robert Chambers**
	Description: the leaderboard should display entries of user data (for a given metric) in rows. Rows should contain: the user's name and the value of a metric being displayed.	Priority: Low Dependencies: 7.1 Sources: FR8: Informatics Feedback
5.	**Requirement Group Name: Inputting data**	**Author: Philip Bardy**
	Description: the system must be able to take self-tracked health data as input from the data input menu	Priority: High
5.1	**Requirement name: Inputting energy intake**	**Author: Robert Chambers**
	Description: The system displays a prompt asking for units ("1. KJ 2. KCal) and an integer value and takes a text input. The text input should be stored as the date and corresponding energy intake in the database.	Priority: High Dependencies: 7.1
5.1.1	**Requirement name: Different units**	**Author: Robert Chambers**
	Description: the program prompts for: "1. KJ, 2. KCal" If "2"is input by the user, then the program converts the following value to kilojoules to store in the database and stores this in the database. The input must be validated so the 1<=x<=2	Priority: Low Dependencies: 5.1.1 Sources: FR7: Validation

5.1.2	**Requirement name: Editing calorie intake**	**Author: Robert Chambers**
	Description: the user can change their previously inputted calorie intake for any date. The program takes text input, a date which is searched for in the database, and a value which is substituted into the database. The input should be validated to test if it is a positive integer	Priority: Medium Dependencies: 5.1.1, 5.1.2, 7.1 Sources: FR7: Validation
5.2, 5.3...	**Requirement Name: Inputting weight and activity data**	**Author: Robert Chambers**
	Description: as with 4.1, the program should display prompt for “1. kilograms or 2. pounds” for weight and take a value, and for activity data similar, and allow editing of these values. These can not be written out due to space constraints of the report.	Priority: Medium Dependencies: 7.1
6	**Requirement Group Name: Goals**	**Author: Philip Bardy**
	Description: the user can set goals which correspond to one one of their tracked metrics.	Priority: Medium
6.1	**Requirement Name: Setting Goals**	**Author: Philip Bardy**
	Description: the user can choose a metric to track, and can specify the duration for which they wish to track the metric. Tracking the metric involves determining if any informatics data entered (for the metric being tracked) contains a value which is below a target value, which is also determined by the user.	Priority: Medium Dependencies: 7.1 Sources: Meeting minutes
6.2	**Requirement Name: Editing Goals**	**Author: Philip Bardy**
	Description: the user can view their currently set goals, and edit the goal information; changing either the metric, the target value, or the span of time for which the goal is set.	Priority: Medium Dependencies: 7.1 Sources: FR1: Editing Inputs
6.3	**Requirement Name: Deleting Goals**	**Author: Robert Chambers**
	Description: the user can view their currently set goals, and choose a goal to stop tracking.	Priority: Medium Dependencies: 7.1
7	**Requirement Group Name: Database**	**Author: Robert Chambers**
	A database is needed to store data for multiple users	Priority: High
7.1	**Requirement Name: Database connection**	**Author: Robert Chambers**
	On startup of the program, a MySQL database named “fitbyte” must be present at localhost for the program to write data to	Priority: High Dependencies: None Sources: Meeting minutes

5.6 - Management of Conflicts Between Requirements (Toby)

There might be a scenario in which a conflict between requirements may arise, be it due to time constraints or manpower, we will need to have a consistent way to decide what requirements we will pursue so that the overall impact on the project is minimised. So due to this reason we will focus on the requirements with the higher priorities and that are dependent on other requirements that will be focussed on in the near future. We will focus on higher priority requirements as it will allow us to create the most complete solution possible given the conflicts. Focussing on requirements that are depended on is also important as it will reduce the chance of conflicting requirements having a knock on effect, creating more conflicts, further into the project.

6 - Design

6.1 - High Level Architecture (HLA)

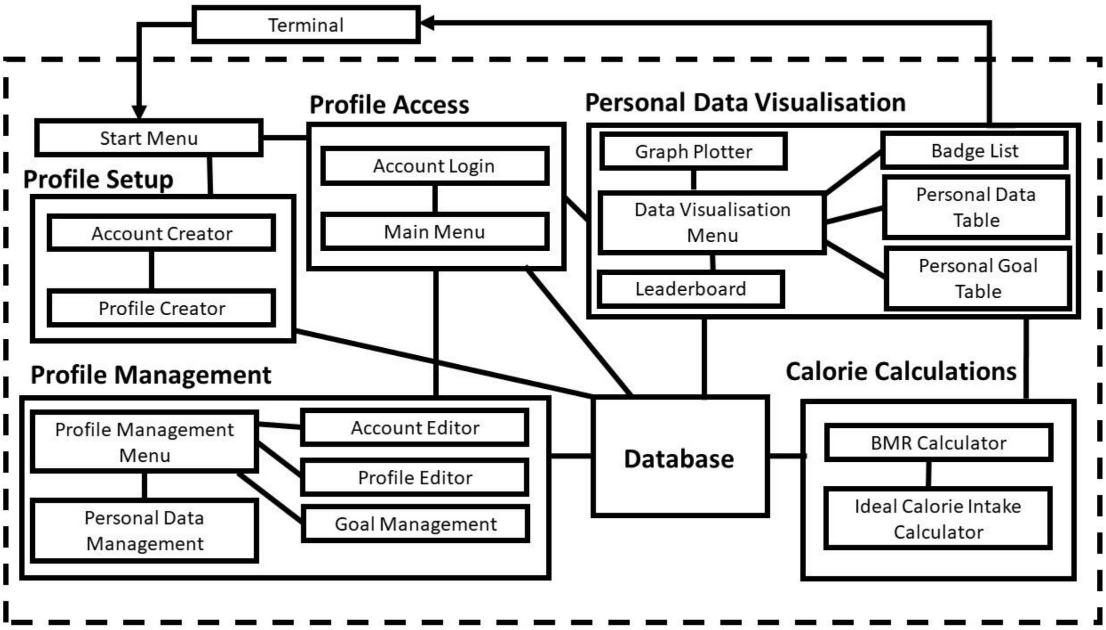


Figure 1

The high-level architecture diagram for our system consists of twenty-one components, with five subsystems.

The terminal exists outside of the system and enables our system to receive input from the user and display output. The Start Menu asks the user to either create a new account or log into an existing account. If the user decides to create an account the Account Creator will prompt the user to input a username and password. Then the Account Creator will check the database to see if the username has already been taken. If the username is available and both the username and password are of an appropriate length, they are stored along with an account ID in the database. The Profile Creator prompts the user to input information that will make up that user's profile, this includes date of birth, current weight, height, sex, and activity rating (out of five). The profile is then saved in the database, being associated with its respective account ID.

Once a profile is set up, a user can begin to access their profile from the start menu. Account Login will prompt the user to log in with their username and password. If the username exists and the password is correct the user is authenticated and proceeds to the Main Menu. Main Menu is responsible for directing the user to either Profile Management Menu or Data

Visualisation Menu. The Profile Management Menu presents the user with some options to manage their profile. If the user chooses to change their username and password, the Account Editor will prompt them for a new username and password and save the new username and password with the associated account ID. If the user chooses to change profile details, Profile Editor prompts for the same five pieces of information (date of birth, current weight, height, sex, and activity rating), the updated profile is stored in the database. Goal Management allows the user to create, edit and delete goals related to diet and weight. A user's goals are stored in the database. Personal Data Management allows the user to add entries related to goals and stores them in the database.

Data Visualisation Menu allows a user to access visualisations of their data. Graph Plotter uses the user's entries to generate a graph illustrating the progress they have made on a particular goal. Badge List s all the badges a user has earned. Personal Goal Table tabulates all entries for goals associated with a user and the progress the user has made. Personal Data Table tabulates all entries a user has made and their relevant dates. Leaderboard generates a list ranking all users by the number of goals they completed.

BMR Calculator uses profile data stored in the database to calculate their BMR. Ideal Calorie Intake Calculator uses the BMR value and the user's activity level to calculate the user's ideal caloric intake. The values for the user's BMR and ideal caloric intake are used by components within the Personal Data Visualisation subsystem.

The database stores a user's account ID, badges, goals, and entries related to goals, when components require these values, they retrieve them from the database.

6.2 - UML Modelling

6.2.1 - Use Case Diagrams (Toby and)

The use case diagram below shows the use cases described in 5.4 displayed in a visual way, showing how each actor interacts with the system in different ways. It is good as it serves as reference material for the overall design and function of the system.

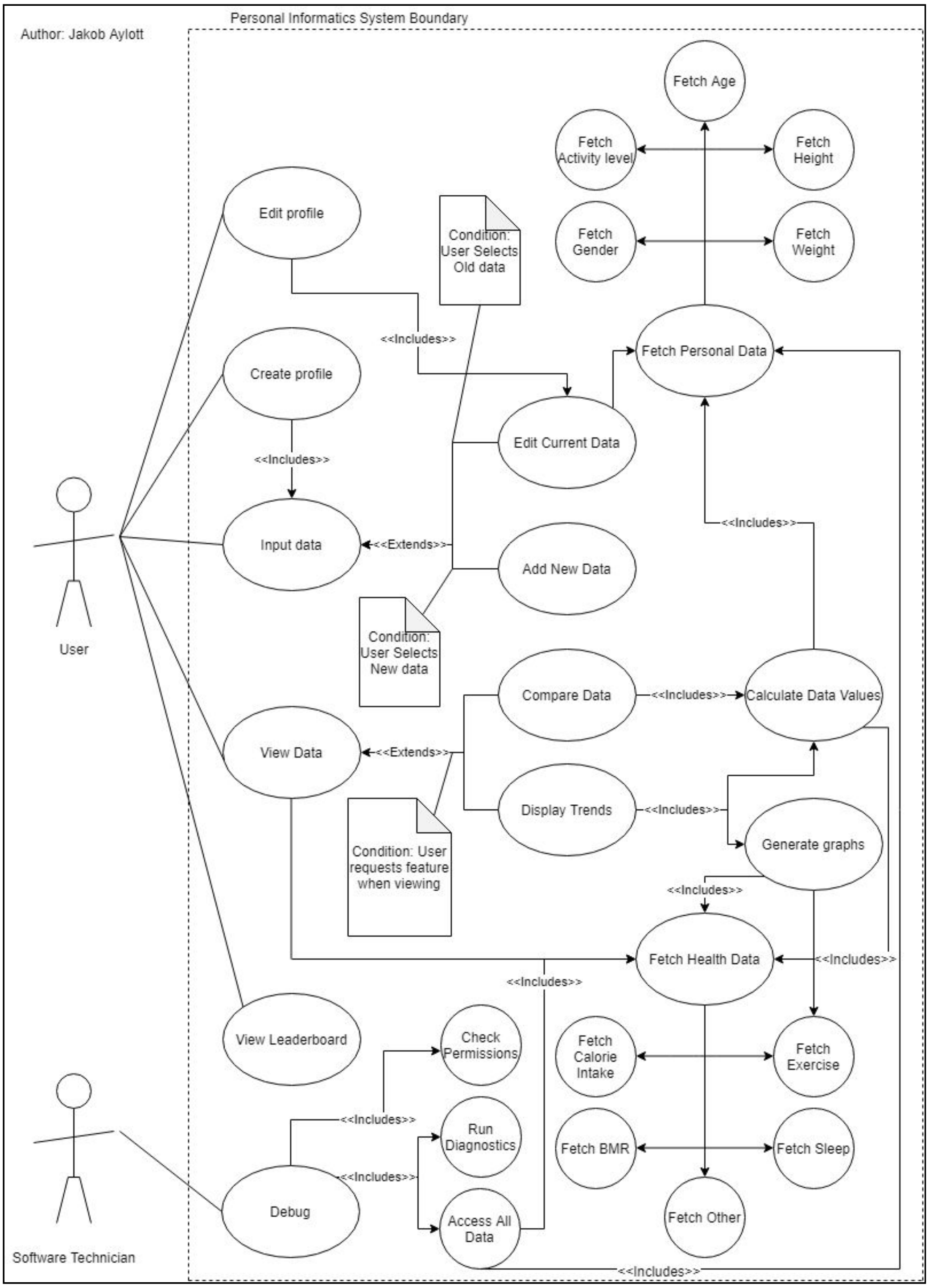


Figure 2

6.2.2 - Class Diagrams (Toby and)

We have created the class diagram below that shows the relationships between different classes in our system, we will then use this as reference material when implementing the application. This will ensure that different parts implemented by different programmers work together correctly.

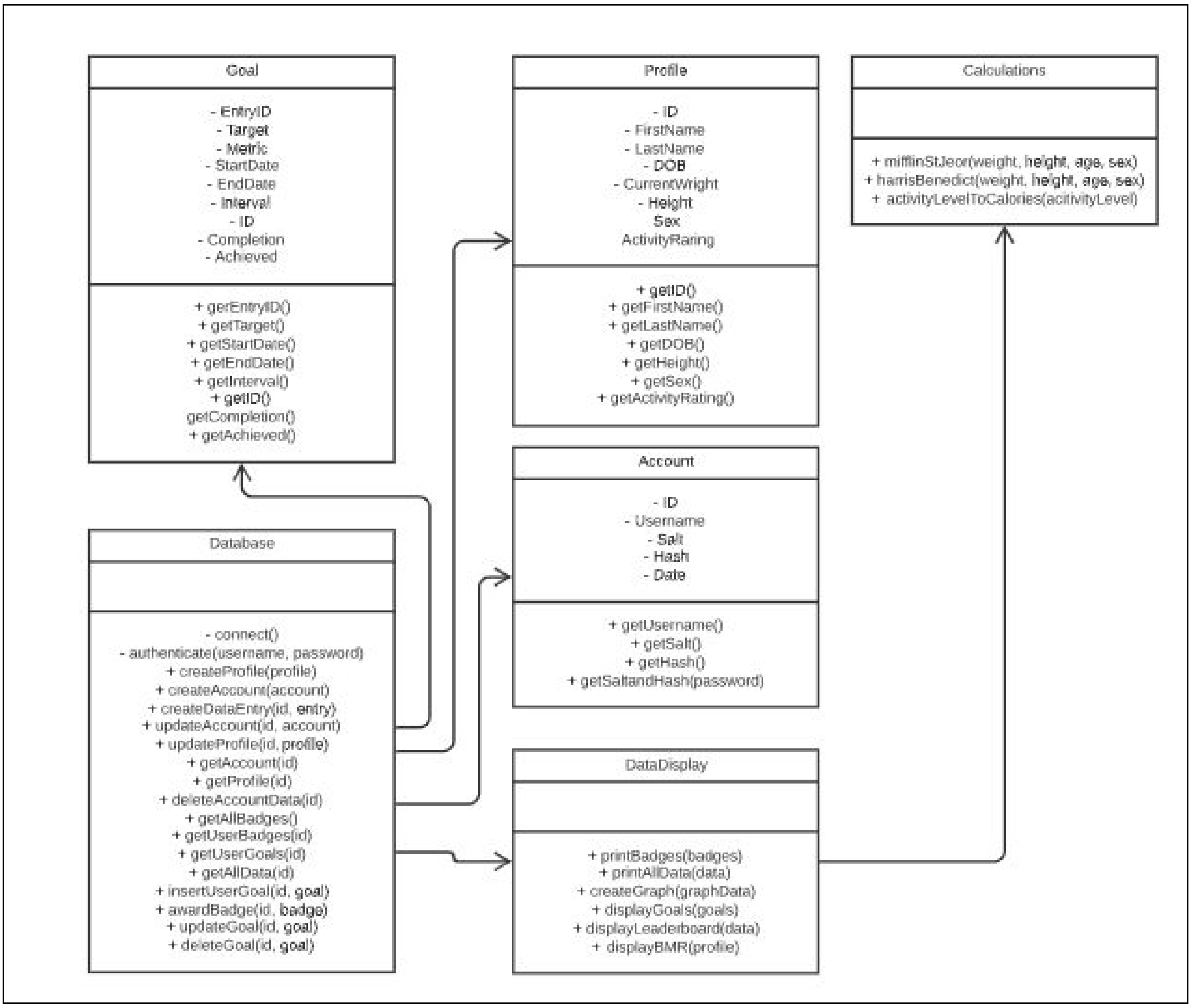

Goal
- EntryID
- Target
- Metric
- StartDate
- EndDate
- Interval
- ID
- Completion
- Achieved
+ gerEntryID()
+ getTarget()
+ getStartDate()
+ getEndDate()
+ getInterval()
+ getID()
getCompletion()
+ getAchieved()
Profile
- ID
- FirstName
- LastName
- DOB
- CurrentWright
- Height
Sex
ActivityRaring
+ getID()
+ getFirstName()
+ getLastName()
+ getDOB()
+ getHeight()
+ getSex()
+ getActivityRating()
Calculations
+ mifflinStJeor(weight, height, age, sex)
+ harrisBenedict(weight, height, age, sex)
+ activityLevelToCalories(acitivityLevel)
Account
- ID
- Username
- Salt
- Hash
- Date
+ getUsername()
+ getSalt()
+ getHash()
+ getSaltandHash(password)
Database
- connect()
- authenticate(username, password)
+ createProfile(profile)
+ createAccount(account)
+ createDataEntry(id, entry)
+ updateAccount(id, account)
+ updateProfile(id, profile)
+ getAccount(id)
+ getProfile(id)
+ deleteAccountData(id)
+ getAllBadges()
+ getUserBadges(id)
+ getUserGoals(id)
+ getAllData(id)
+ insertUserGoal(id, goal)
+ awardBadge(id, badge)
+ updateGoal(id, goal)
+ deleteGoal(id, goal)
DataDisplay
+ printBadges(badges)
+ printAllData(data)
+ createGraph(graphData)
+ displayGoals(goals)
+ displayLeaderboard(data)
+ displayBMR(profile)

Figure 3

6.2.3 - Sequence Diagrams (Toby and)

Below are the sequence diagrams we have created to outline the computation involved in our processes at run time. This will also be used when implementing and serves as a graphical representation of the processes that will be easier to follow than a text description.

6.2.3.1 - Calculate BMR

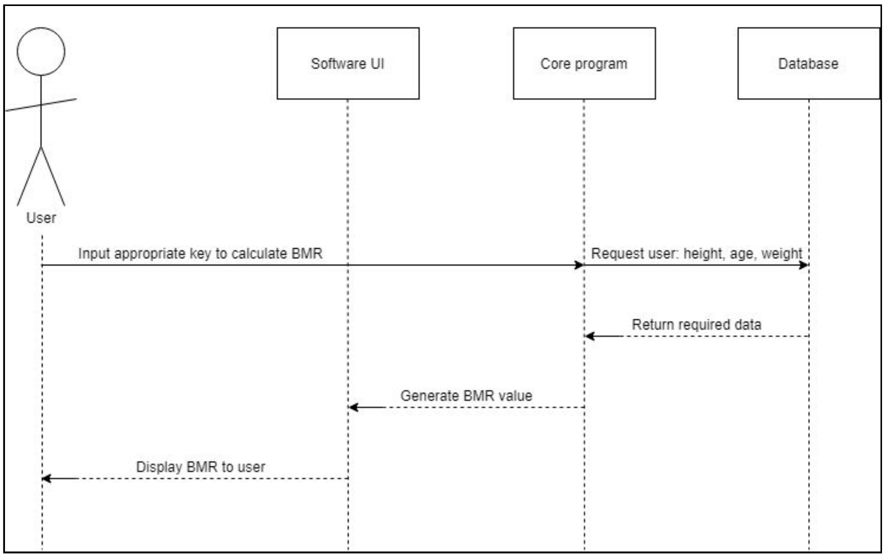


Figure 4

6.2.3.2 - Log In

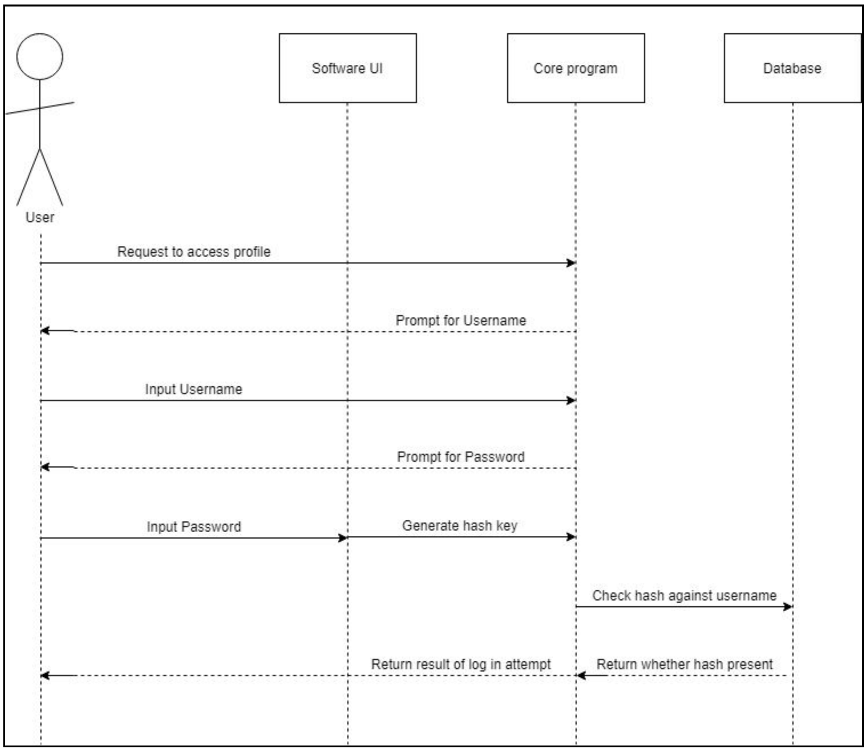


Figure 5

6.2.4 - State Machine Diagrams (Toby)

State machine diagrams are used to design and analyse real-time systems in terms of events occuring and their outcomes. We will not be creating a system that uses real time elements so it is unnecessary for us to design one of the diagrams.

6.3 - Evaluation of Our Design (Toby)

In this section, we have created both an HLA and diagrams representing the system we are building and how each of the different components act and interact with one another in different ways. The visual way of representing the system, as stated in previous sections, provides a much clearer way of how the system functions as opposed to explaining it with text. This clarity will ensure that, when implementing the system, all the different programmers correctly know how the system functions so the risk of different implemented parts not working together is reduced. However, our design could be improved through the addition of further sequence diagrams for more processes in the system as the current design leaves ambiguity of how some processes exactly work, this will likely not impact the system however as due to the other parts in the design it is clear what the end goal of each process is.

7 - Software Testing (Verification)

7.1 - Testing Plan

Test Case	Req	Precons.	Steps	Test Data	Expected Output	Actual	Pass/ Fail
Account creation: When prompted to enter a username enter a value 0 characters long.					[No errors]	[No errors]	Pass
Account creation: When prompted to enter a username enter a value 32 characters long				ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEF	[No errors]	[No errors]	Pass
Account creation: When prompted to enter a username enter a value greater than 32 characters long.				ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFG	“Input must be between 0 and 32 characters”	“Input must be less than 32 characters”	Pass
Account creation: When prompted to ensure a				testuser	[No errors]	[No errors]	Pass

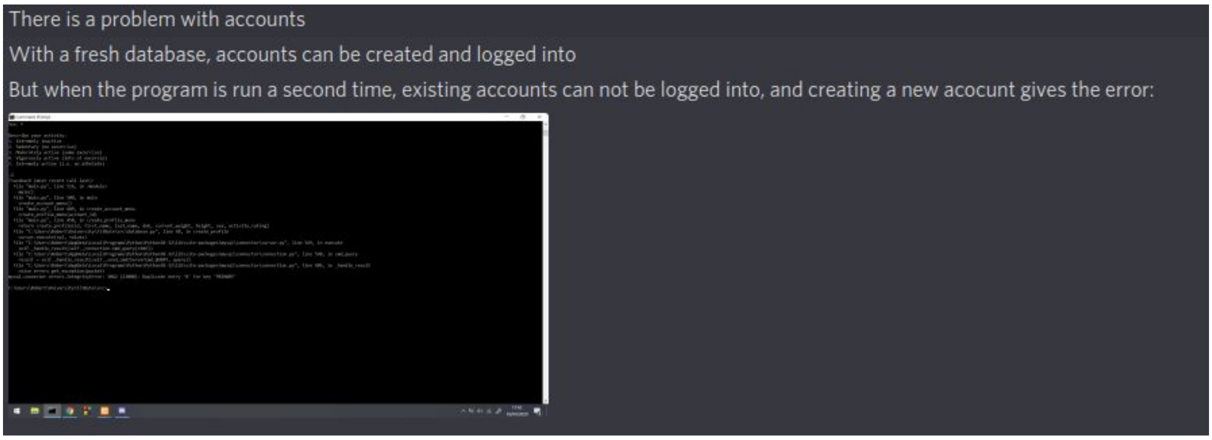
username, enter a value between 0 and 32 characters long.							
Account creation: When prompted to enter a username, enter a username that belongs to another user.				testuser	“The username ‘testuser’ is already taken”		Fail
Account creation: When prompted to enter a password enter a value less than 6 characters long.				pass	“Input must be between 6 and 72 characters”	“Input must be more than 6 characters”	Pass
Account creation: When prompted to enter a password, enter a value 6 letters long.				passwd	[No errors]		Pass
Account creation: When prompted to enter a password enter a value 72 characters long.				THIS_IS_A_PASSWORD_THAT_IS_WAY_WAY_WAY_WAY_WAY_LONGER_THAN_NECESSARY	[No errors]	[No errors]	Pass
Account creation: When prompted to repeat the password you just entered, enter a different value.				Different password	“Passwords do not match”	[Option not available]	Fail
Account creation: When prompted to repeat the password you just entered, enter the same value.				passwd	[No errors]	[Option not available]	Fail
Profile creation: When prompted to enter a first or last name, enter a value 0 characters long.	1.1	The user has created an account.			[No errors]	[No errors]	Pass
Profile creation: When prompted to enter a first or last name, enter a value 35 characters long.	1.1	The user has created an account.		ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHI	[No errors]	[No errors]	Pass
Profile creation: When prompted to enter a first or last name, enter a value 36	1.1.	The user has created an		ABCDEFGHIJKLMNOPQRSTUV	“Names cannot be longer than 35 characters”	“Input must be less than 35 characters”	Pass

characters long.		account.		WXYZABC DEFGHIJ			
Profile creation: When prompted to enter a first or last name, enter a value between 0 and 35 characters long.	1.1	The user has created an account.		firstname	[No errors]	[No errors]	Pass
Profile creation: When prompted to enter a date of birth enter a date in the future.	1.2	The user has created an account.		01/01/9999	“Incorrect date format”	“Dates cannot be in the future”	Pass
Profile creation: When prompted to enter a date, enter a date 150 years ago.	1.2	The user has created an account.		01/01/1800	“Incorrect date format”		Fail
Profile creation: When prompted to enter a date of birth, enter a date less than 150 years ago.	1.2	The user has created an account.		01/01/2000	[No errors]		Pass
Profile creation: When prompted to enter your current weight enter a string.	1.3	The user has created an account.		64kg	“Input must be decimal number”	“Input must be a decimal number”	Pass
Profile creation: When prompted to enter your current weight, enter a float.	1.3	The user has created an account.		64	[No errors]		Pass
Profile creation: When prompted to enter your current weight enter a value in stone and pounds and check its converted into kg before being stored in the database.	1.3	The user has created an account.		12 12	81.6466266		Pass
Profile creation: When prompted to enter your current weight in stone and pounds, enter an invalid value.	1.3	The user has created an account.		12 22	“Input must be between 0 and 14”	“Input must be less than %d 14”	Pass
Profile creation; When prompted to enter your height enter a string.	1.4	The user has created an account.		“10metres”	“Input must a decimal”	“Input must be a decimal number”	Pass
Profile creation: When prompted to enter your height in feet and inches ensure it is converted into metres before	1.4	The user has created an account.		6 1	185.42	[No errors]	Pass

being stored in the database.							
Authentication: a user, when prompted to enter a username and password is logged in, and shown the application’s main menu.		The user has an account.		testuser passwd	“Login successful”	“Login successful."	Pass
Authentication: a user is prompted to re-enter their login details if either their username or password was incorrect.		The user has an account.		fakeuser fakepasswd	“Authentication failed” “Your username or password was incorrect”	“Authentication failed” “Your username or password was incorrect”	Pass
Data processing: selecting the option from the menu to ‘view my BMR’ correctly outputs the user’s BMR.	.2.1	The user is logged in.	Login. Enter 3 Enter 2		“Your BMR is x”	“Your BMR” x x	Pass
Data processing: selecting the option from the menu to ‘view my BMR’ also outputs a recommended calorie intake value.	2.2	The user is logged in.	Login. Enter 3 Enter 2		“Your recommended calorie intake is x”	“Based on your activity level you need x calories per day”	Pass
Data processing: the user can view the raw personal informatics data they have inputted as a series of tables.	2.3	The user is logged in.	Login. Enter 3 Enter 1		[Tables with dates and values of informatics data entries]	[Tables with dates and values of informatics data entries]	Pass
Data visualisation: the user can view the raw information as a graph, showing trends in their data over time.	2.5	The user is logged in.	Login. Enter 3 Enter 3 Enter 1		[A graph plotting informatics data against time]	[A graph plotting informatics data against time]	Pass
Data input: the user can add personal informatics data about one of the system’s targeted metrics.	4	The user is logged in.	Login. Enter 3 Enter 4				Pass
Goal: the user can set a goal for a given metric.	5.1		Login. Enter 4 Enter 5				Pass
Goal: the user can edit a goal that has previously been set.	5.2		Login. Enter 4 Enter 4				Pass
Goal: the user can delete a goal that has previously been set.	5.3		Login. Enter 4 Enter 6				Pass

7.2 - Evidence of Testing and Debugging

Black box testing was conducted regularly. In one case Robert asked a family member with no knowledge of the code to perform inputs without any information, except what is displayed by the program. This case is described below with screenshots of the discord messages

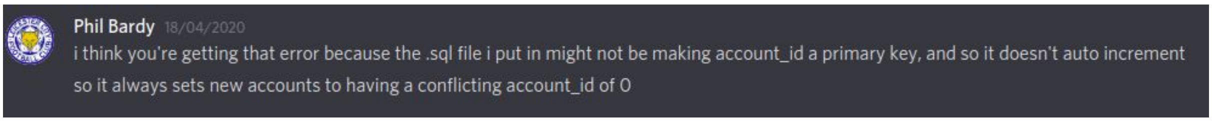


There is a problem with accounts

With a fresh database, accounts can be created and logged into

But when the program is run a second time, existing accounts can not be logged into, and creating a new acocunt gives the error:

Above: after the black box tester showed Robert the issue, Robert messaged Phil, who has the most knowledge of databases, so would likely know best how to modify the code to resolve the issue.



Phil Bardy 18/04/2020
i think you're getting that error because the .sql file i put in might not be making account_id a primary key, and so it doesn't auto increment so it always sets new accounts to having a conflicting account_id of 0

Above: Phil identified the issue and modified the code. Robert pulled from the project's repository and allowed the black box tester to continue their testing. They were satisfied that accounts could now be created and logged into. This occurred during sprint 2.

8 - Conclusion

In conclusion, our system was successful in implementing the most important features that. Although we did not add all gamification features as intended, the program still functioned as a personal informatics system. In sprint 3 goals were added, giving some element of gamification, however leaderboards were not due to time constraints. The agile nature of the scrum methodology allowed development to move in the direction most needed, with sprint 1 successfully creating a basis for the personal informatics application, sprint 2 adding the important user data input and viewing features and sprint 3 included adding goals and adding an element of gamification to the system.

To realise our portability functional requirement, we chose python as our language. This allowed our application to run successfully on both Windows and Linux, and while not tested on MacOS, should run on this as well. Initially, our software system was intended to be written in Java. However, after the first Scrum meeting, members of the group expressed the desire to switch to Python. They believed the expressiveness of the language was better suited for our software system and could lead to faster development. This is an example as to why the Scrum methodology is very useful, it allowed us to change our approach before getting too far into the development cycle. This contrasts with how we worked within the first semester, in which our 'waterfall' approach would not have permitted us to make such frequent changes to our system.

Python is not a particularly fast language, however as the user interface will be composed of text and the calculation involved in informatics feedback is quite simple, this is not too much of a concern. Furthermore, it has an easy to use graphing library: matplotlib, that one of our members had some previous experience with, which facilitated cross training in our project. As this was a prototype, speed may be of more concern if a future version were to be created with a GUI, here a compiled language may be used, such as C++, which would require cross compilation to run on different platforms.

Our preferred method of communication, through a Discord server, had both positives and negatives. The main benefit was that everything that was said was recorded, so if a member was later unsure of what was agreed, they could return to what was said and clarify. On the other hand, not meeting in person led to decreased personal responsibility, with only three members regularly communicating. Our weekly scrum meetings and our sprint reviews were conducted through Microsoft Teams due to the recent pandemic. This was not foreseen in risk planning, but was mitigated by use of virtual meetings, rather than face to face meetings. Group organisation as a whole was poorer than in semester 1, this was likely due to the format of our project requiring greater personal responsibility, this combined with the fact that meetings in person were no longer possible, lead to a disconnect with the project for some members.

One lesson learned from this is that people can not always be relied upon to attend or contribute unless there is sufficient incentive. In the workplace, this would be wages, at university this is marks, however year one marks do not contribute to degree classification, maybe this is why some members did not think it worth their time. The pandemic was a potential source of disruption, but not one that would cause complete absence.

Reflecting on the ACM Code of Ethics and Professional Conduct (Acm.org., 2018), we believe not everyone in our group adhered to the second principle in Section 2 of the code which encourages computing professionals to “maintain high standards of professional competence, conduct, and ethical practice”. In particular, the guidelines state professionals require “skill in communication”. The lack of communication from some members in the later weeks of this project represents a shortcoming in this responsibility. The guidelines for the first principle in Section 2 of the code state that professionals be aware of “any serious negative consequences affecting any stakeholder that may result from poor quality work.” Colleagues are stakeholders. The lack of contribution from some members of the group resulted in an inflated load of work for others, imposing unwarranted stress on them. This also violates the responsibility to respect the dignity of colleagues, mentioned earlier within the guidelines for the first principle in Section 2.

Another lesson is that Boehm’s list is an accurate predictor of risk sources. We did not initially identify gold plating as a realistic source of risk, yet this came up multiple times, most notably in what was being tracked by the system, which branched out to include items such as salt at one point, but was corrected due to regular testing and discussion by project members.

Aspect	Semester 1	Semester 2	Discussion of outcomes

Responding to changing requirements	None, requirement were set early on the project, HLA's and other documents were made to match	Good responsiveness to change, as priorities changed through the project, code and documentation was modified accordingly	Semester 2 was superior in this respect, due to the agile nature of scrum against the more rigid waterfall method
Professional code of conduct	Adhered to the ACM Code of Ethics and Professional Conduct. The principle to only "perform work in areas of competence" did not have to be strictly followed as we did not have to actually implement our software system. Therefore we were able to design a software system outside of the scope of our technical ability.	Used the ACM Code of Ethics and Professional Conduct as a guide on how to act as a team. Semester 2 also required cross training as programming was involved. Not all members adhered to the principles	Semester 2 demonstrated cross training, however overall did not go as well as semester 1 overall as the code of conduct was not upheld by as many within the group
Remote working	Occasional discussion through a discord server, with regular face to face meetings	Initially, frequent discussion on a discord server, with weekly face to face meetings and a sprint review every few weeks. This shifted to all meetings and discussion being held remotely.	Semester 1, face to face improved responsibility

9 - References

Spanos, D. and Hankey, C.R., 2010. The Habitual Meal and Snacking Patterns of University Students in Two Countries and Their Use of Vending Machines, Journal of Human Nutrition and Dietetics [Online], 23(1). Available from: https://onlinelibrary-wiley-com.ezproxy1.bath.ac.uk/doi/abs/10.1111/j.1365-277X.2009.01005.x# [Accessed 06/05/2020]

Robert Bargar, MD, 2017. Obesity: Health risks associated with obesity [Online]. WebMD. Available from: [https://www.onhealth.com/content/1/health_risks_associated_with_obesity [Accessed 06/05/2020]

Weinstein, L. and Laverghetta, A., 200. College Student Stress and Satisfaction with Life. College Student Journal [Online] Part A, Vol. 43 Issue 4. Available from https://web-a-ebscohost-com.ezproxy1.bath.ac.uk/ehost/detail/detail?vid=1&sid=6ead4fab-bdf0-448f-a139-dbe7cced0a8d%40sessionmgr4006&bdata=JnNpdGU9ZWhvc3QtbGl2ZQ%3d%3d#AN=55492493&db=eue [Accessed 06/05/202]

Weinstein. 2008, Why students quit university. Unpublished raw data

Stiglic, N. and Viner, R.M., 2019. Effects of screentime on the health and well-being of children and adolescents: a systematic review of reviews. BMJ Open [Online], Volume 9(1). https://bmjopen.bmj.com/content/bmjopen/9/1/e023191.full.pdf [Accessed 06/05/2020]

Naci, H. and Ioannidis, J.P.A., 2015. Evaluation of Wellness Determinants and Interventions by Citizen Scientists. JAMA [Online], Volume 314(2), p.121. Available from https://jamanetwork-com.ezproxy1.bath.ac.uk/journals/jama/fullarticle/2330497 [Accessed 06/05/2020]

Song, V, 2018. Fitbit Charge 3: Among the Top Dedicated Fitness Trackers. PC Magazine, pp.36–41. [Online] Available from: https://uk.pcmag.com/fitness-trackers/117903/fitbit-charge-3 [Accessed 06/05/2020]

Li, I., Dey, A. and Forlizzi, J., 2010. A stage-based model of personal informatics systems. Proceedings of the SIGCHI Conference on human factors in computing systems, pp.557–566. [Online] Available from: https://dl-acm-org.ezproxy1.bath.ac.uk/doi/10.1145/1753326.1753409 [Accessed 06/05/2020]

WHO. Strengthening mental health promotion. Geneva, World Health Organization, 2001: Fact sheet, No. 220. [Online] Available from: https://www.who.int/news-room/fact-sheets/detail/mental-health-strengthening-our-response [Accessed 06/05/2020]

World Health Organization. Promoting Mental Health: Concepts, Emerging Evidence, Practice (Summary Report). Geneva: World Health Organization; 2004. [Online] Available from: https://www.who.int/mental_health/evidence/en/promoting_mhh.pdf [Accessed 06/05/2020]

Tennant, R., Hiller, L., Fishwick, R., Platt, P. et al. The Warwick-Edinburgh Mental Well-being Scale (WEMWBS): development and UK validation. Health and Quality of Life Outcome 2007;5(63) [Online]. Available from: https://hqlo.biomedcentral.com/articles/10.1186/1477-7525-5-63 [Accessed 06/05/2020]

Fitzpatrick, A., 2018. The 6 Biggest New iPhone Features Apple Just Announced. Time.com, pp.1–2. Available from: https://time.com/5300514/iphone-apple-new-features-ios-12/ [Accessed 06/05/2020]

Rapp, A. et al., 2018. Designing a personal informatics system for users without experience in self-tracking: a case study. Behaviour and Information Technology, Volume 37(4), pp.335–366. Available from: https://www.tandfonline.com/doi/abs/10.1080/0144929X.2018.1436592?journalCode=tbit20 [Accessed 06/05/2020]

Fadhil, A., 2019. Comparison of Self-monitoring Feedback Data from Electronic Food and Nutrition Tracking Tools. Available from: https://www.researchgate.net/publication/332494219_Comparison_of_Self-monitoring_Feedback_Data_from_Electronic_Food_and_Nutrition_Tracking_Tools [Accessed 06/05/2020]

Hasson, R. E., Howe, C. A., Jones, B. L., Freedson, P. S., 2011. Accuracy of four resting metabolic rate prediction equations: Effects of sex, body mass index, age, and race/ethnicity. Journal of Science & Medicine in Sport, 14(4), pp.344–352. [Online] Available from: https://www.ncbi.nlm.nih.gov/pubmed/21458373 [Accessed 06/05/2020]

Rapp, A., 2018. Gamification for Self-Tracking: From World of Warcraft to the Design of Personal Informatics Systems. Proceedings of the 2018 CHI Conference on human factors in computing systems, pp.1–15. [Online] Available from: https://dl.acm.org/doi/10.1145/3173574.3173654 [Accessed 06/05/2020]

Hersh, W. R.. (2020). Improving Health Care Through Information. [online] jamanetwork. Available at: https://jamanetwork.com/journals/jama/article-abstract/195427 [Accessed 21/02/2020].

Personalinformatics.org. (2020). Personal Informatics. [online] Available at: http://www.personalinformatics.org/ [Accessed 21/02/2020].

Cooke, R. and Papadaki, A. (2020). Nutrition label use mediates the positive relationship between nutrition knowledge and attitudes towards healthy eating with dietary quality among university students in the UK. [online] https://www.sciencedirect.com/. Available at: https://www.sciencedirect.com/science/article/pii/S0195666314004462 [Accessed 21/02/2020].

Waldhäusl, S., Aceijas, C., Lambert, N. and Bello-Corassa, R. (2020). Determinants of nutritional imbalance among UK university students: a cross sectional study: Sabrina Waldhäusl. [online]Available at: https://academic.oup.com/eurpub/article/26/suppl_1/ckw174.058/2449235 [Accessed 21/02/2020].

Haddadi, H., Ofli, F., Mejova, Y., Weber, I. and Srivastava, J. (2020). 360-degree Quantified Self - IEEE Conference Publication. [online] Ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/abstract/document/7349770 [Accessed 23/02/ 2020].

Rapp, A. (2020). Gamification for Self-Tracking | Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. [online] Dl.acm.org. Available at: https://dl.acm.org/doi/abs/10.1145/3173574.3173654 [Accessed 23/02/2020]

Dalleck, L., The Science of Post-Exercise Recovery. [online] acewebcontent.azureedge.net. Available at: https://acewebcontent.azureedge.net/SAP-Reports/Post-Exercise_Recovery_SAP_Reports.pdf [Accessed 24/02/2020].

Wilson, D. (2019). How Many Rest Days Do My Muscles Really Need to Heal? [online] greatist.com. Available at: https://greatist.com/fitness/do-my-muscles-need-two-days-recover [Accessed 24/02/2020].

Acm.org. (2018). *ACM Code of Ethics and Professional Conduct* [online] Available at: https://www.acm.org/code-of-ethics. [Accessed 06/05/2020]

10 - Appendix

10.1 - Group Contribution Form

Member	Score	Signature	Signature
Phil Bardy	10		SPhilip
Tiago Reis	10		Tiago Reis
Robert Chambers	10		Robert Chambers
Toby Britton	10		Britton
Varun Alvakonda	6		Varun
Yakob Aylott	8		
Lucas Bebell	0		None (no attendance)
Solomon Arnell-Larcombe	0		None (no attendance)

<u>Justification</u>

Varun - Contribution lower. Contributed regularly in the initial few weeks, unable to contribute much during lockdown due to poor internet connection that was only fixed recently.

Lucas, Solomon - Didn’t contribute.

10.2 - Group Meeting Minutes

20/02/2020

The purpose of this meeting was to narrow down a domain for our personal informatics system.

Our initial ideas were:

- Sleep tracking
- Health tracking
    - Dietary tracking
    - Exercise tracking
    - Heart rate
- Study tracking
- Web surfing / computer use

We decided that the domain we should go for was health tracking.

We created a github repository and started a project so we could track the progress we make.

Tasks to do for next meeting:

- Research calorie intake (and calories burnt during exercise) - Rob, Lucas
- Research diet (calorie intake) - Toby, Phil
- Research timing of exercise (understand when is best to eat) - Solomon
- Research exercise recovery times - Jake, Solomon
- Produce a survey of questions to ask students about health and exercise habits - Tiago

Communication will take place through a dedicated Discord server

27/02/2020

We discussed what some of the higher priority features of our application should contain, as well as specific features we wished to implement.

We began fleshing out the requirements our system should have. These were based on discussions over current personal informatics systems we had researched prior to the meeting. Members suggested:

- The need to create accounts (so a leaderboard of users could be created)
- The need to input the number of calories consumed per day
- The need for gamification to motivate users to reach their goals
- The need for users to set their own goals
- The need to analyse the inputted data
- The need for some form of visual representation of the data inputted by the user

05/03/2020

We discussed what things we needed to do for sprint 1:

- Write the section for the initial problem area (Tiago)
- Write the section for what out solution to the problem in this problem area are (Jakob)
- Database design (Philip)
- Feature backlog + additional requirements (Toby and Rob)
- Methods of establish system requirements
- Finish managing project risks
- Sprint planning
- Sprint tracking

We established our first sprint would be over 2 weeks from now.

12/03/2020
Mid sprint meeting

Jakob: Working on problem area solution, about halfway. Estimated completion: 15/03/2020

Rob: Working on initial feature backlog, nearly finished. Estimated completion:12/03/2020 evening. Once this is complete I will create a separate document with a sprint one backlog.

Phillip: Working on database design, sql file uploaded to github and produced a complete database structure. Plans on writing some queries for data insertion by 19/03/2020

Tiago: Working on introduction to the problem area, mostly complete, needs citations and needs to be put in google drive. Estimated completion time: 12/03/2020 evening.

Toby: Contributed to the initial requirements list

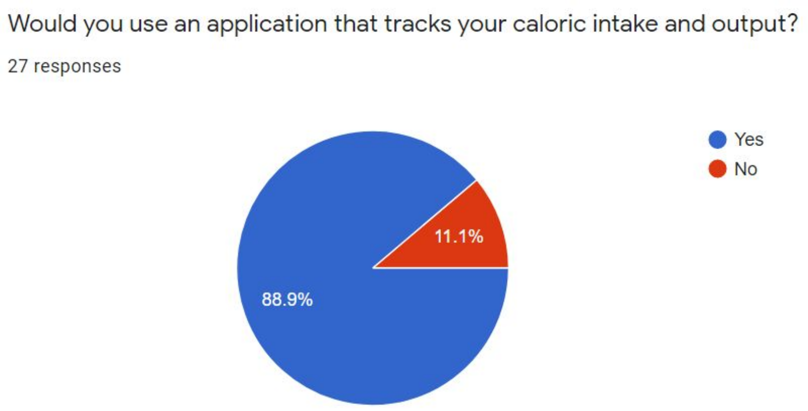
Agreed with scrum master that first sprint review would take place 19/03/2020

26/03/2020
Absences: Varun unable to join due to internet connection problems
Phil showed progress on the Python program, which has so far implemented account creation and some goals functionality

10.3 - Primary Research

10.3.1 - Survey

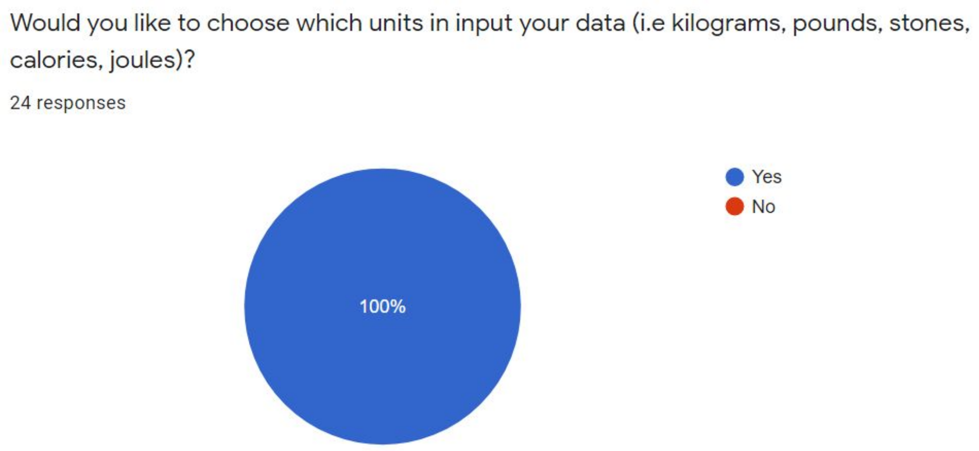
One of the ways we performed research was to collect first party data from stakeholders in the domain, in this case it was University students, we did this through a survey as it is quicker and easier to perform than other data collection techniques. We initially wanted to perform a stratified sample based on activity level to get a proportional representation of the population of University students, however due to time constraints we opted for a random sample instead. The data we collected can be seen below:



Would you use an application that tracks your caloric intake and output?

27 responses

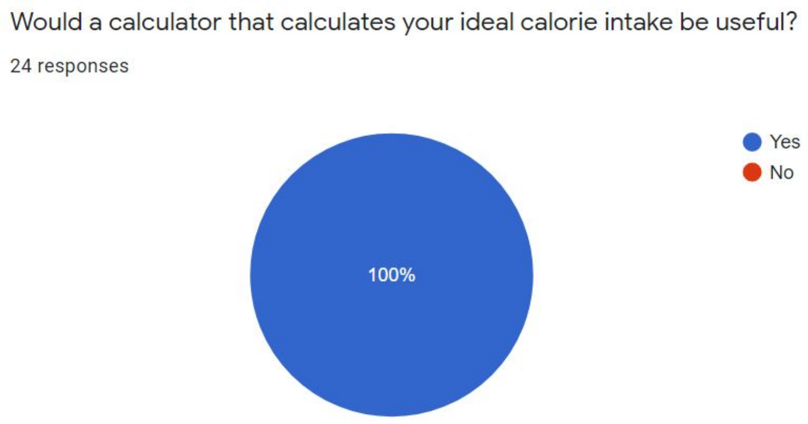
Figure 6



Would you like to choose which units in input your data (i.e kilograms, pounds, stones, calories, joules)?

24 responses

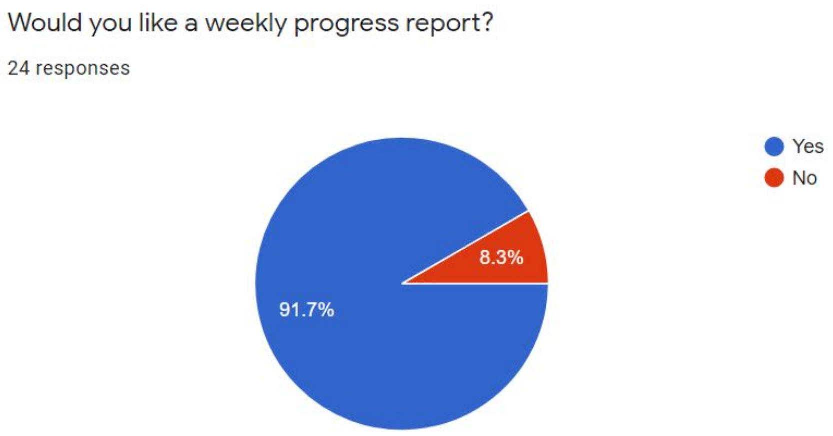
Figure 7



Would a calculator that calculates your ideal calorie intake be useful?

24 responses

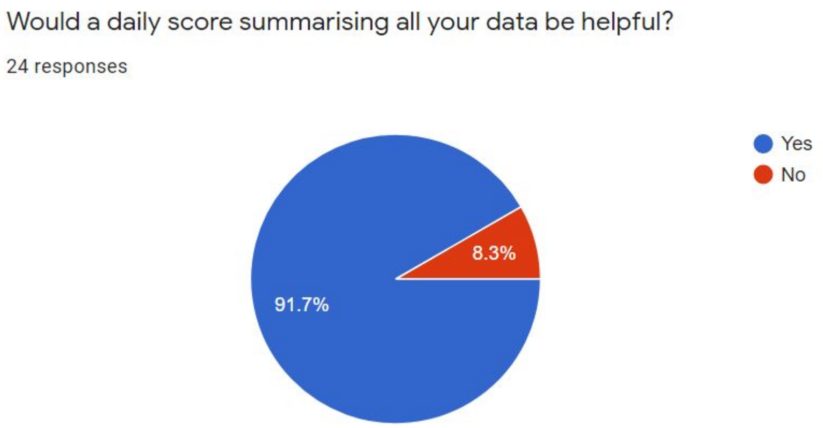
Figure 8



Would you like a weekly progress report?

24 responses

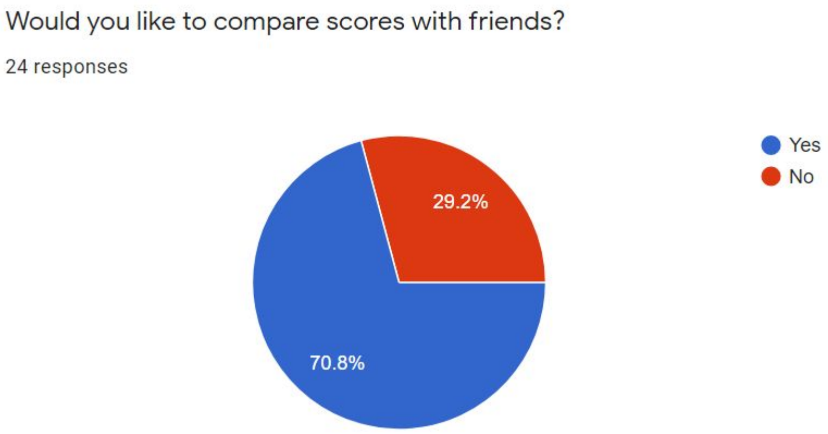
Figure 9



Would a daily score summarising all your data be helpful?

24 responses

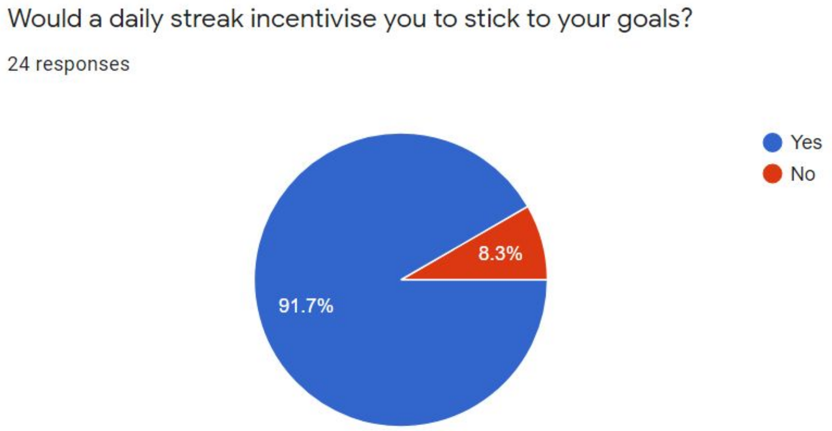
Figure 10



Would you like to compare scores with friends?

24 responses

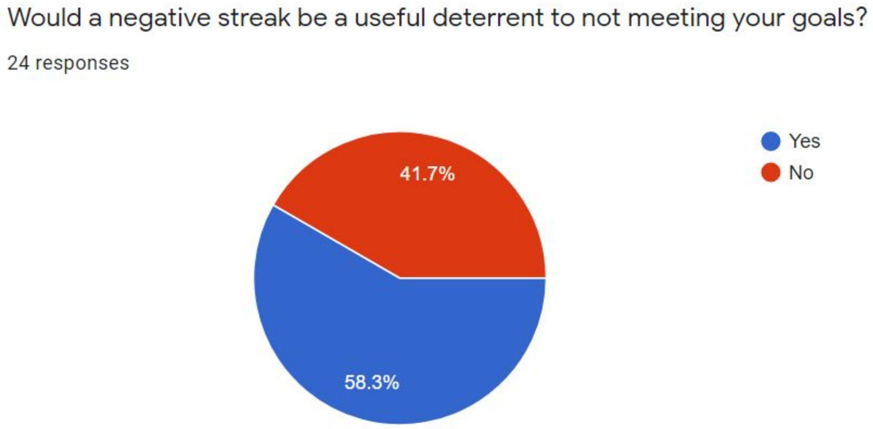
Figure 11



Would a daily streak incentivise you to stick to your goals?

24 responses

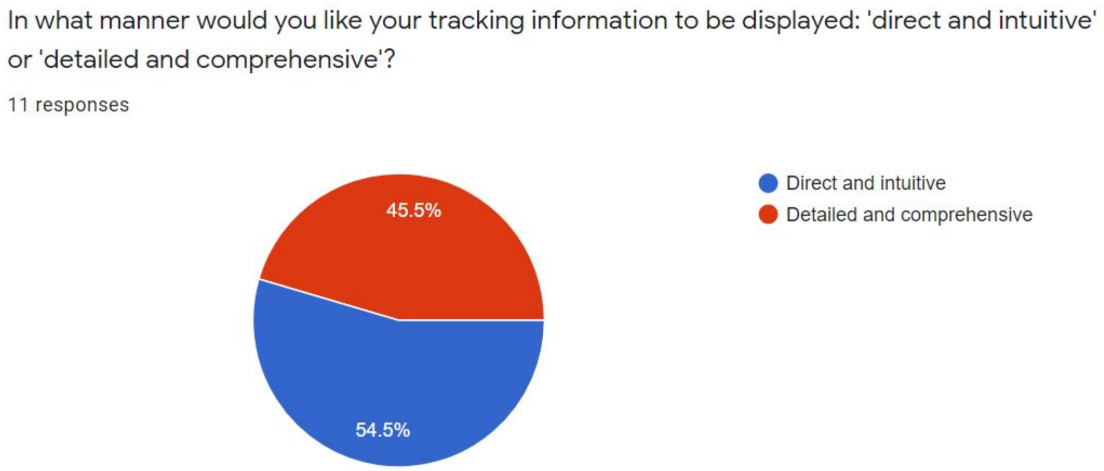
Figure 12



Would a negative streak be a useful deterrent to not meeting your goals?

24 responses

Figure 13



In what manner would you like your tracking information to be displayed: 'direct and intuitive' or 'detailed and comprehensive'?

11 responses

Figure 14

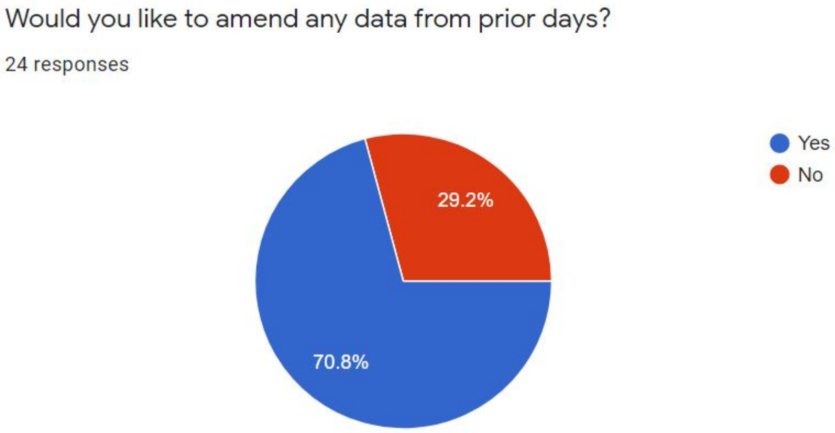


Figure 15

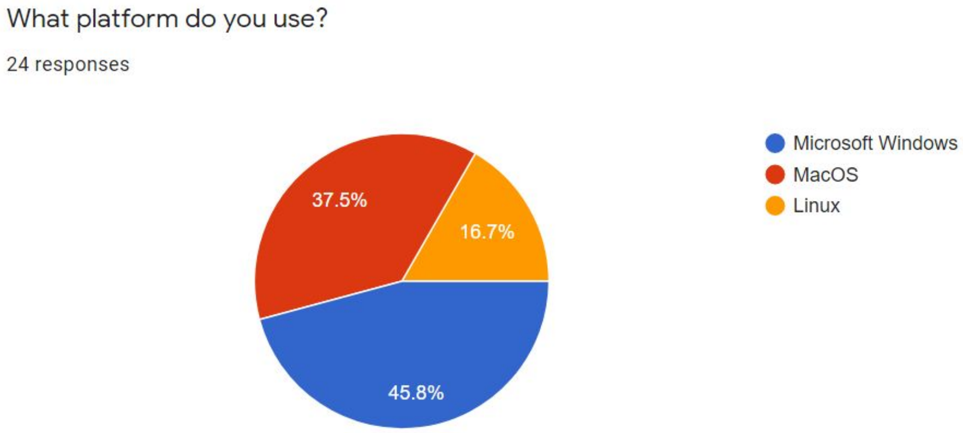


Figure 16

10.3.2 - Dietary Personal Informatics Research

Personal Informatics (PI)

Personal informatics (PI) is closely linked to the concept of quantified self (QS) which refers to the 'engagement in the self-tracking of', 'biological, physical, behavioural, or environmental information' to promote self-awareness and to improve certain metrics associated with the activities being tracked, e.g. reducing the number calories consumed in a given day when tracking a diet (R, W. and Hersh, M. (2020)). PI refers to the 'class of tools that help people collect personally relevant information' which enables an automated approach to the concept of QS, leveraging ubiquitous computing and Internet of Things (IoT) technologies (Personalinformatics.org. (2020)).

Dietary Problems With UK University Students

A study conducted in 2013 on 500 students across 37 UK universities found that 'dietary recommendations for fat, added sugar and fast food intake' were met but, 'recommendations for calcium, fibre, fruit and vegetable and dairy product intake' did not (Cooke, R. and Papadaki, A. (2020)). Another study of 468 students similarly found that '46% of respondents had unbalanced diets' (Waldhäusl, S., Aceijas, C., Lambert, N. and Bello-Corassa, R. (2020)). 'Poor dietary habits are associated with numerous preventable chronic diseases and premature deaths. Higher education students are considerably susceptible to adopting unhealthy dietary patterns, leading to an increased prevalence of overweight and obesity' (Waldhäusl, S., Aceijas, C., Lambert, N. and Bello-Corassa, R. (2020)).

'In the United Kingdom (UK), health promoting dietary and lifestyle changes at the population level could significantly reduce the overall costs of diet-related disease, which were found to be £5.8billion in 2006–2007' (Cooke, R. and Papadaki, A. (2020)). Despite the clear economic arguments there currently are no effective national health programs promoting healthy dietary practices for UK university students.

Analysis of Dietary PI Systems

Public Sharing

Early dietary and health PI systems focussed more on data collection and data processes than other equally valued properties such as security and interoperability with other applications. More recent systems, however, favour a health data sharing approach through social media platforms like Facebook and Twitter. The merits of public health sharing are debatable but the incentive of public accountability or wide scale support for meeting dietary targets has driven an increase in the adoption of this methodology amongst PI systems (Haddadi, H., Ofli, F., Mejova, Y., Weber, I. and Srivastava, J. (2020)).

Gamification

Another methodology used by personal informatics systems to motivate and engage users in keeping to their targets is gamification, which is 'is precisely the use of "game design elements in non-game contexts"' (Rapp, A. (2020)). The primary reason for the incorporation of game elements in PI systems is that instruments used to measure physiological data 'require a considerable effort from their users, in terms of time, compliance, and long-term engagement', which is difficult to obtain when the user is not fully dedicated to the 'quantified-self' lifestyle (Rapp, A. (2020)). Without the automation of data entry, we see from applications such as Fitbit, the use of self-trackers can become burdensome which leads to low retention of new users. Often users who are unfamiliar with the metrics these systems are measuring will see a plethora of meaningless numbers which they cannot process and cannot make actionable decisions with – hence resulting in the quick abandonment of the technology.

Adding scoreboards, badges, and other competitive elements has become a new trend in PI systems which supposedly increases user engagement. These appeal to the competitive nature of the user, but Rapp argues these are simple systems which do not consider the user's current situation. Rapp suggests systems could instead invite users 'to confront their current data with those of individuals with similar features and experiences' (Rapp, A. (2020)). Such a system would provide the user with more reasonable targets and provide a more useful way of processing data; if a similar user improves their dietary score by taking action X, then perhaps you should also take action X. By simulating the effects of taking actions within a PI system its users are encouraged to experiment with how choices can affect certain parameters. For example, users create a virtual ‘player’ based off their current diet and see immediately how eating a meal could affect their weight, or calories consumed, or any other metric the system is tracking. These insights could be calculated by the system itself, or by analysing the actions of similar users which enacting these decisions in real life.

10.3.3 - Exercise Recovery Time Research

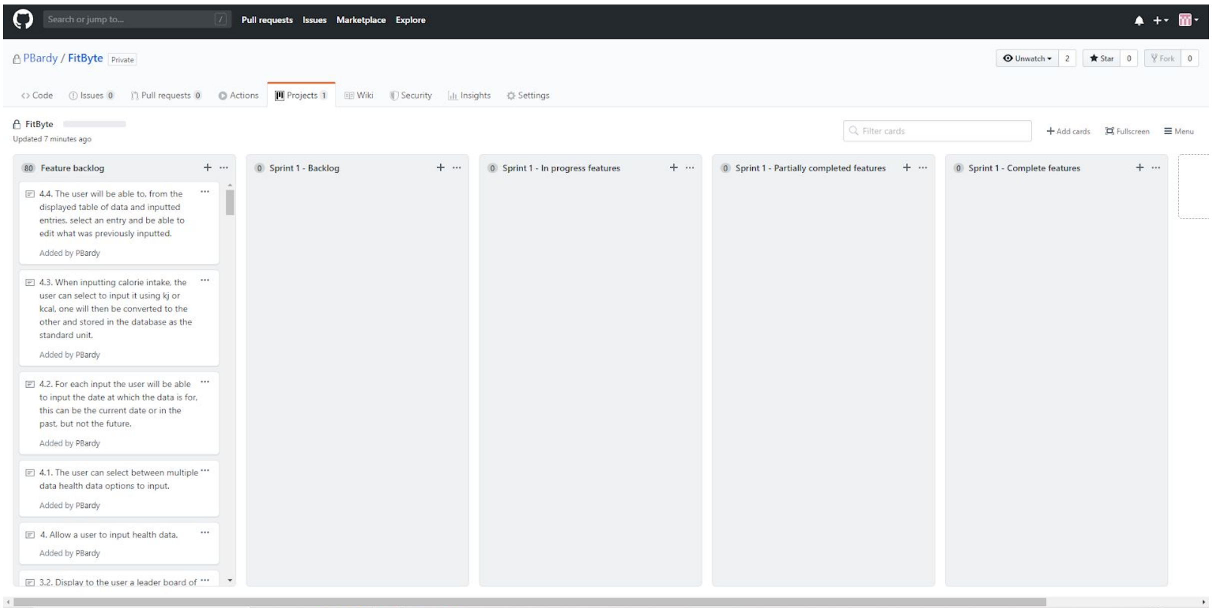
Research shows there is a large variation in required recovery time/type, heavily dependent on the user’s personal info: lifestyle, health etc. Along with intensity of exercise, frequency of it and experience in it. This is discussed in a paper on post-exercise recovery times by the American Council on exercise scientific advisory panel (Dalleck, L.), which is referenced by this article which summarises some of the main points (Wilson, D. (2019)). After finding this information during research I believe implementing a feature that will recommend the rest/recovery time after an exercise will create a high level and likely risk as it would have a large cost in terms of time to effectively implement the feature in our system. For it to be useful to the user there would have to be a high level of adaptability based on personal information and more calculations, this would be a large investment which could be much better spent on more integral features that are core to the functionality of the application.

10.4 - Testing

We did not perform any unit tests or additional tests.

10.5 - Evidence of Our AGILE Approach

10.5.1 - Project Board, Start of the Project



FitByte
Updated 7 minutes ago

80 Feature backlog

4.4. The user will be able to, from the displayed table of data and inputted entries, select an entry and be able to edit what was previously inputted.
Added by PBardy

4.3. When inputting calorie intake, the user can select to input it using kj or kcal, one will then be converted to the other and stored in the database as the standard unit.
Added by PBardy

4.2. For each input the user will be able to input the date at which the data is for, this can be the current date or in the past, but not the future.
Added by PBardy

4.1. The user can select between multiple data health data options to input.
Added by PBardy

4. Allow a user to input health data.
Added by PBardy

3.2. Display to the user a leader board of

0 Sprint 1 - Backlog

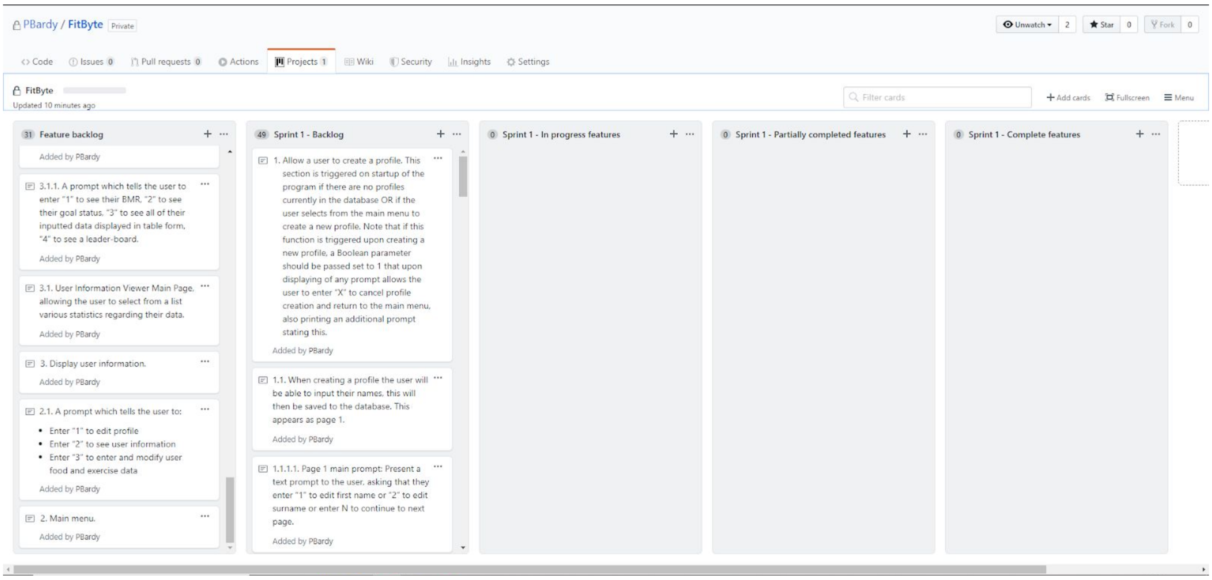
0 Sprint 1 - In progress features

0 Sprint 1 - Partially completed features

0 Sprint 1 - Complete features

(15/03/2020)

10.5.2 - Start of Sprint 1



31 Feature backlog

Added by PBardy

3.1.1. A prompt which tells the user to enter "1" to see their BMR, "2" to see their goal status, "3" to see all of their inputted data displayed in table form, "4" to see a leader-board.

Added by PBardy

3.1. User Information Viewer Main Page, allowing the user to select from a list various statistics regarding their data.

Added by PBardy

3. Display user information.

Added by PBardy

2.1. A prompt which tells the user to:

- Enter "1" to edit profile
- Enter "2" to see user information
- Enter "3" to enter and modify user food and exercise data

Added by PBardy

2. Main menu.

Added by PBardy

49 Sprint 1 - Backlog

1. Allow a user to create a profile. This section is triggered on startup of the program if there are no profiles currently in the database OR if the user selects from the main menu to create a new profile. Note that if this function is triggered upon creating a new profile, a Boolean parameter should be passed set to 1 that upon displaying of any prompt allows the user to enter "X" to cancel profile creation and return to the main menu, also printing an additional prompt stating this.

Added by PBardy

1.1. When creating a profile the user will be able to input their names, this will then be saved to the database. This appears as page 1.

Added by PBardy

1.1.1.1. Page 1 main prompt: Present a text prompt to the user, asking that they enter "1" to edit first name or "2" to edit surname or enter N to continue to next page.

Added by PBardy

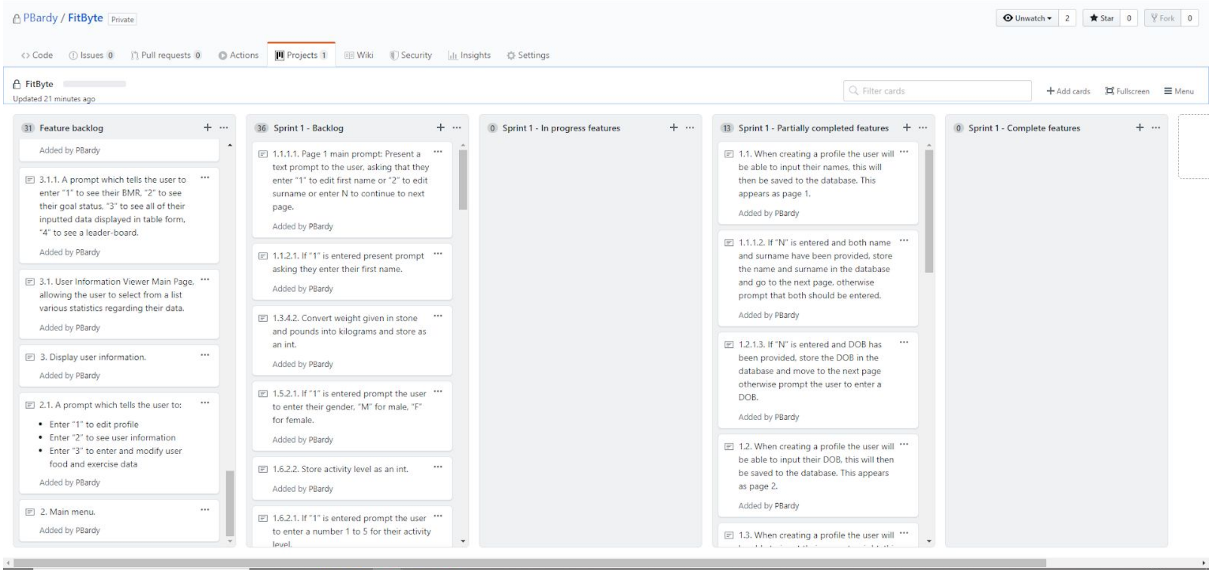
0 Sprint 1 - In progress features

0 Sprint 1 - Partially completed features

0 Sprint 1 - Complete features

(16/03/2020)

10.5.3 - Project Board, Sprint 1, Scrum meeting 1



31 Feature backlog

Added by PBardy

3.1.1. A prompt which tells the user to enter "1" to see their BMR, "2" to see their goal status, "3" to see all of their inputted data displayed in table form, "4" to see a leader-board.

Added by PBardy

3.1. User Information Viewer Main Page, allowing the user to select from a list various statistics regarding their data.

Added by PBardy

3. Display user information.

Added by PBardy

2.1. A prompt which tells the user to:

- Enter "1" to edit profile
- Enter "2" to see user information
- Enter "3" to enter and modify user food and exercise data

Added by PBardy

2. Main menu.

Added by PBardy

36 Sprint 1 - Backlog

1.1.1.1. Page 1 main prompt: Present a text prompt to the user, asking that they enter "1" to edit first name or "2" to edit surname or enter N to continue to next page.

Added by PBardy

1.1.2.1. If "1" is entered present prompt asking they enter their first name.

Added by PBardy

1.3.4.2. Convert weight given in stone and pounds into kilograms and store as an int.

Added by PBardy

1.5.2.1. If "1" is entered prompt the user to enter their gender, "M" for male, "F" for female.

Added by PBardy

1.6.2.2. Store activity level as an int.

Added by PBardy

1.6.2.1. If "1" is entered prompt the user to enter a number 1 to 5 for their activity level.

0 Sprint 1 - In progress features

13 Sprint 1 - Partially completed features

1.1. When creating a profile the user will be able to input their names, this will then be saved to the database. This appears as page 1.

Added by PBardy

1.1.1.2. If "N" is entered and both name and surname have been provided, store the name and surname in the database and go to the next page, otherwise prompt that both should be entered.

Added by PBardy

1.2.1.3. If "N" is entered and DOB has been provided, store the DOB in the database and move to the next page otherwise prompt the user to enter a DOB.

Added by PBardy

1.2. When creating a profile the user will be able to input their DOB, this will then be saved to the database. This appears as page 2.

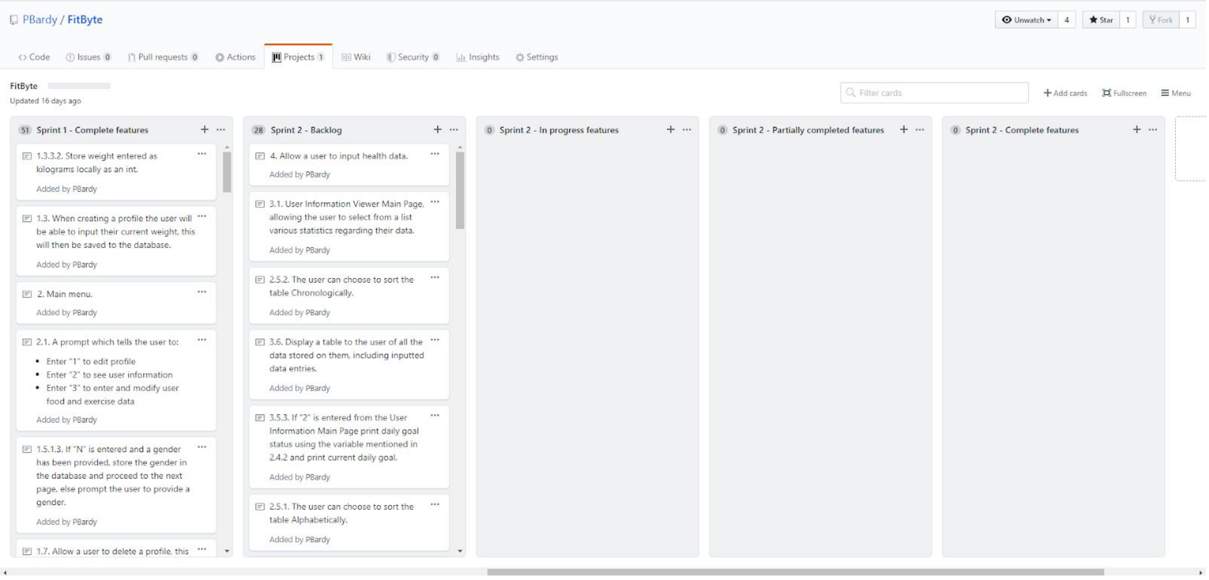
Added by PBardy

1.3. When creating a profile the user will

0 Sprint 1 - Complete features

(26/03/2020)

10.5.4 - Project Board, Sprint 1, Scrum Meeting 2



FitByte

Updated 16 days ago

51 Sprint 1 - Complete features

1.3.3.2. Store weight entered as kilograms locally as an int.

Added by PBardy

1.3. When creating a profile the user will be able to input their current weight, this will then be saved to the database.

Added by PBardy

2. Main menu.

Added by PBardy

2.1. A prompt which tells the user to:

- Enter "1" to edit profile
- Enter "2" to see user information
- Enter "3" to enter and modify user food and exercise data

Added by PBardy

1.5.1.3. If "N" is entered and a gender has been provided, store the gender in the database and proceed to the next page, else prompt the user to provide a gender.

Added by PBardy

1.7. Allow a user to delete a profile, this

28 Sprint 2 - Backlog

4. Allow a user to input health data.

Added by PBardy

3.1. User Information Viewer Main Page, allowing the user to select from a list various statistics regarding their data.

Added by PBardy

2.5.2. The user can choose to sort the table Chronologically.

Added by PBardy

3.6. Display a table to the user of all the data stored on them, including inputted data entries.

Added by PBardy

3.5.3. If "2" is entered from the User Information Main Page print daily goal status using the variable mentioned in 2.4.2 and print current daily goal.

Added by PBardy

2.5.1. The user can choose to sort the table Alphabetically.

Added by PBardy

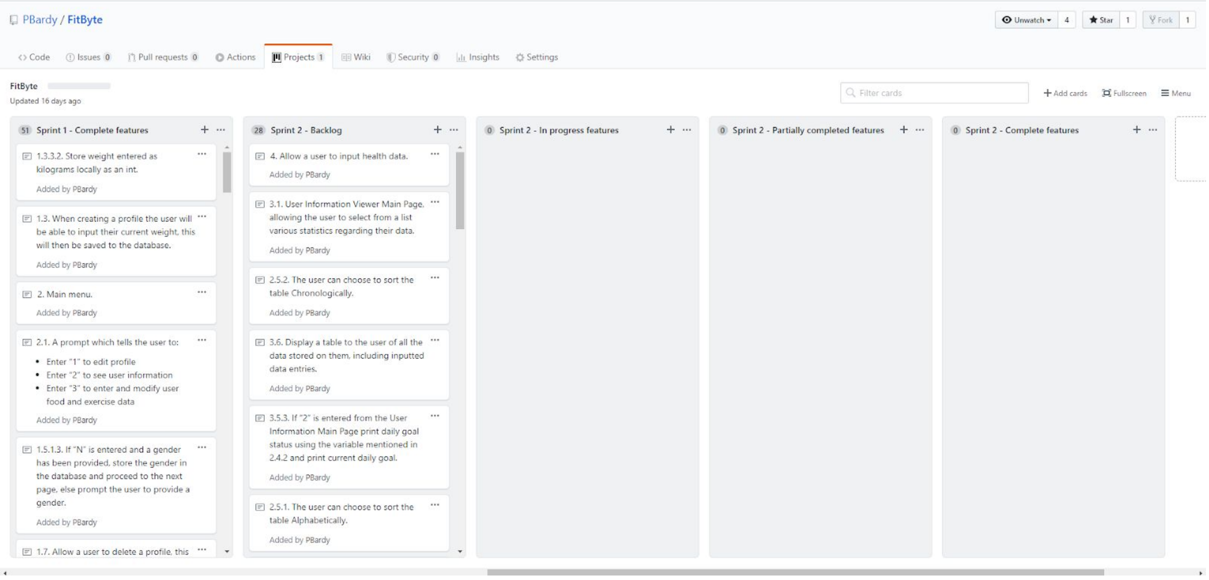
0 Sprint 2 - In progress features

0 Sprint 2 - Partially completed features

0 Sprint 2 - Complete features

(02/04/2020)

10.5.5 - Project Board, Sprint 2, Start



FitByte

Updated 16 days ago

51 Sprint 1 - Complete features

1.3.3.2. Store weight entered as kilograms locally as an int.

Added by PBardy

1.3. When creating a profile the user will be able to input their current weight, this will then be saved to the database.

Added by PBardy

2. Main menu.

Added by PBardy

2.1. A prompt which tells the user to:

- Enter "1" to edit profile
- Enter "2" to see user information
- Enter "3" to enter and modify user food and exercise data

Added by PBardy

1.5.1.3. If "N" is entered and a gender has been provided, store the gender in the database and proceed to the next page, else prompt the user to provide a gender.

Added by PBardy

1.7. Allow a user to delete a profile, this

28 Sprint 2 - Backlog

4. Allow a user to input health data.

Added by PBardy

3.1. User Information Viewer Main Page, allowing the user to select from a list various statistics regarding their data.

Added by PBardy

2.5.2. The user can choose to sort the table Chronologically.

Added by PBardy

3.6. Display a table to the user of all the data stored on them, including inputted data entries.

Added by PBardy

3.5.3. If "2" is entered from the User Information Main Page print daily goal status using the variable mentioned in 2.4.2 and print current daily goal.

Added by PBardy

2.5.1. The user can choose to sort the table Alphabetically.

Added by PBardy

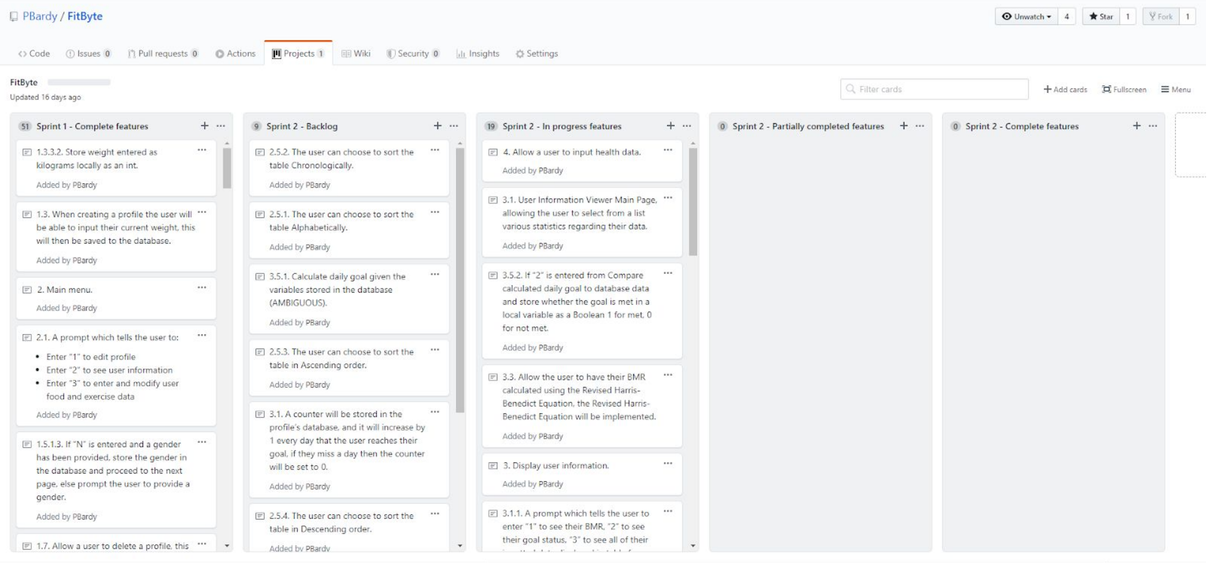
0 Sprint 2 - In progress features

0 Sprint 2 - Partially completed features

0 Sprint 2 - Complete features

(09/04/2020)

10.5.6 - Project Board, Sprint 2, Scrum Meeting 1



FitByte

Updated 16 days ago

51 Sprint 1 - Complete features

- 1.3.3.2. Store weight entered as kilograms locally as an int.
  Added by PBardy
- 1.3. When creating a profile the user will be able to input their current weight, this will then be saved to the database.
  Added by PBardy
- 2. Main menu.
  Added by PBardy
- 2.1. A prompt which tells the user to:
  - Enter "1" to edit profile
  - Enter "2" to see user information
  - Enter "3" to enter and modify user food and exercise data

  Added by PBardy
- 1.5.1.3. If "N" is entered and a gender has been provided, store the gender in the database and proceed to the next page, else prompt the user to provide a gender.
  Added by PBardy
- 1.7. Allow a user to delete a profile, this

9 Sprint 2 - Backlog

- 2.5.2. The user can choose to sort the table Chronologically.
  Added by PBardy
- 2.5.1. The user can choose to sort the table Alphabetically.
  Added by PBardy
- 3.5.1. Calculate daily goal given the variables stored in the database (AMBIGUOUS).
  Added by PBardy
- 2.5.3. The user can choose to sort the table in Ascending order.
  Added by PBardy
- 3.1. A counter will be stored in the profile's database, and it will increase by 1 every day that the user reaches their goal, if they miss a day then the counter will be set to 0.
  Added by PBardy
- 2.5.4. The user can choose to sort the table in Descending order.
  Added by PBardy

19 Sprint 2 - In progress features

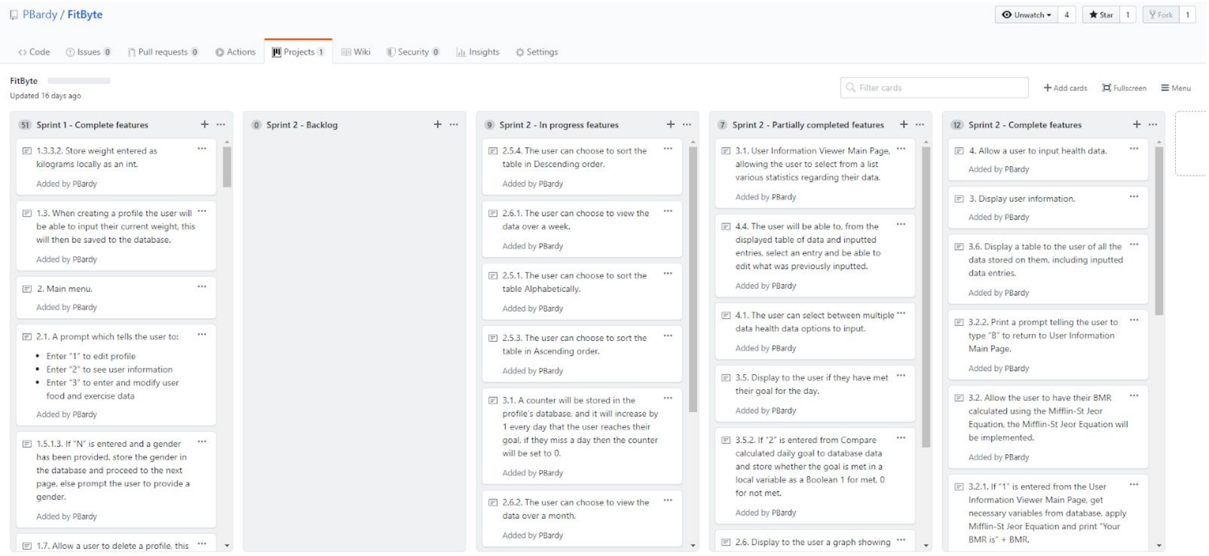
- 4. Allow a user to input health data.
  Added by PBardy
- 3.1. User Information Viewer Main Page, allowing the user to select from a list various statistics regarding their data.
  Added by PBardy
- 3.5.2. If "2" is entered from Compare calculated daily goal to database data and store whether the goal is met in a local variable as a Boolean 1 for met, 0 for not met.
  Added by PBardy
- 3.3. Allow the user to have their BMR calculated using the Revised Harris-Benedict Equation, the Revised Harris-Benedict Equation will be implemented.
  Added by PBardy
- 3. Display user information.
  Added by PBardy
- 3.1.1. A prompt which tells the user to enter "1" to see their BMR, "2" to see their goal status, "3" to see all of their

0 Sprint 2 - Partially completed features

0 Sprint 2 - Complete features

(16/04/2020)

10.5.7 - Project Board, Sprint 2, Scrum Meeting 2



FitByte

Updated 16 days ago

51 Sprint 1 - Complete features

- 1.3.3.2. Store weight entered as kilograms locally as an int.
  Added by PBardy
- 1.3. When creating a profile the user will be able to input their current weight, this will then be saved to the database.
  Added by PBardy
- 2. Main menu.
  Added by PBardy
- 2.1. A prompt which tells the user to:
  - Enter "1" to edit profile
  - Enter "2" to see user information
  - Enter "3" to enter and modify user food and exercise data

  Added by PBardy
- 1.5.1.3. If "N" is entered and a gender has been provided, store the gender in the database and proceed to the next page, else prompt the user to provide a gender.
  Added by PBardy
- 1.7. Allow a user to delete a profile, this

0 Sprint 2 - Backlog

9 Sprint 2 - In progress features

- 2.5.4. The user can choose to sort the table in Descending order.
  Added by PBardy
- 2.6.1. The user can choose to view the data over a week.
  Added by PBardy
- 2.5.1. The user can choose to sort the table Alphabetically.
  Added by PBardy
- 2.5.3. The user can choose to sort the table in Ascending order.
  Added by PBardy
- 3.1. A counter will be stored in the profile's database, and it will increase by 1 every day that the user reaches their goal, if they miss a day then the counter will be set to 0.
  Added by PBardy
- 2.6.2. The user can choose to view the data over a month.
  Added by PBardy

7 Sprint 2 - Partially completed features

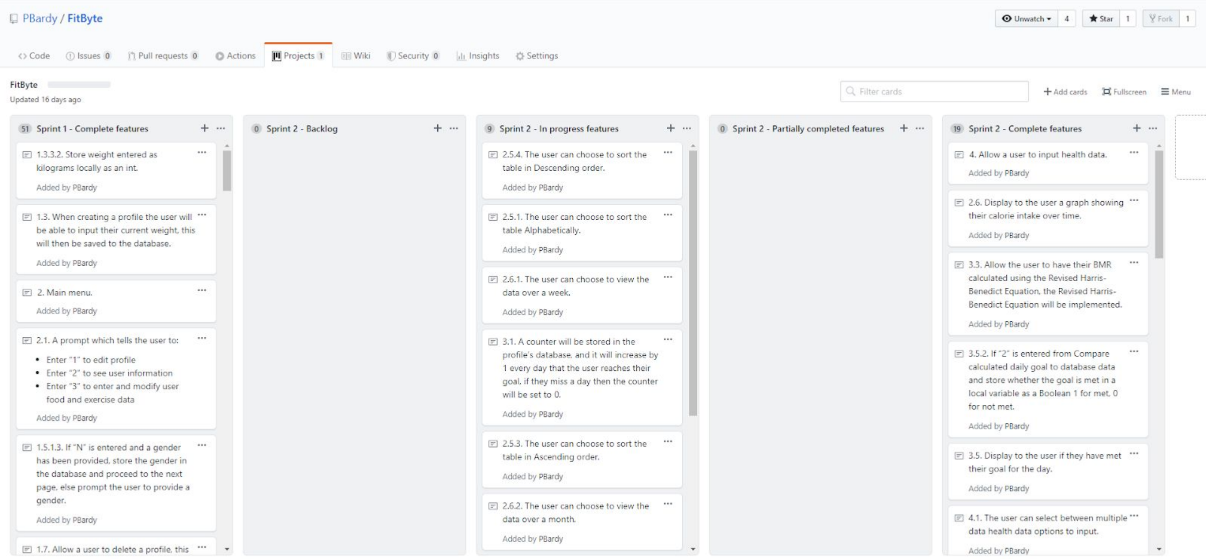
- 3.1. User Information Viewer Main Page, allowing the user to select from a list various statistics regarding their data.
  Added by PBardy
- 4.4. The user will be able to, from the displayed table of data and inputted entries, select an entry and be able to edit what was previously inputted.
  Added by PBardy
- 4.1. The user can select between multiple data health data options to input.
  Added by PBardy
- 3.5. Display to the user if they have met their goal for the day.
  Added by PBardy
- 3.5.2. If "2" is entered from Compare calculated daily goal to database data and store whether the goal is met in a local variable as a Boolean 1 for met, 0 for not met.
  Added by PBardy
- 2.6. Display to the user a graph showing

12 Sprint 2 - Complete features

- 4. Allow a user to input health data.
  Added by PBardy
- 3. Display user information.
  Added by PBardy
- 3.6. Display a table to the user of all the data stored on them, including inputted data entries.
  Added by PBardy
- 3.2.2. Print a prompt telling the user to type "B" to return to User Information Main Page.
  Added by PBardy
- 3.2. Allow the user to have their BMR calculated using the Mifflin-St Jeor Equation, the Mifflin-St Jeor Equation will be implemented.
  Added by PBardy
- 3.2.1. If "1" is entered from the User Information Viewer Main Page, get necessary variables from database, apply Mifflin-St Jeor Equation and print "Your BMR is" + BMR.

(23/04/2020)

10.5.8 - Project Board, Sprint 2, Scrum Meeting 3



51 Sprint 1 - Complete features

1.3.3.2. Store weight entered as kilograms locally as an int.
Added by PBardy

1.3. When creating a profile the user will be able to input their current weight, this will then be saved to the database.
Added by PBardy

2. Main menu.
Added by PBardy

2.1. A prompt which tells the user to:

- Enter "1" to edit profile
- Enter "2" to see user information
- Enter "3" to enter and modify user food and exercise data

Added by PBardy

1.5.1.3. If "N" is entered and a gender has been provided, store the gender in the database and proceed to the next page, else prompt the user to provide a gender.
Added by PBardy

1.7. Allow a user to delete a profile, this

0 Sprint 2 - Backlog

9 Sprint 2 - In progress features

2.5.4. The user can choose to sort the table in Descending order.
Added by PBardy

2.5.1. The user can choose to sort the table Alphabetically.
Added by PBardy

2.6.1. The user can choose to view the data over a week.
Added by PBardy

3.1. A counter will be stored in the profile's database, and it will increase by 1 every day that the user reaches their goal, if they miss a day then the counter will be set to 0.
Added by PBardy

2.5.3. The user can choose to sort the table in Ascending order.
Added by PBardy

2.6.2. The user can choose to view the data over a month.
Added by PBardy

0 Sprint 2 - Partially completed features

19 Sprint 2 - Complete features

4. Allow a user to input health data.
Added by PBardy

2.6. Display to the user a graph showing their calorie intake over time.
Added by PBardy

3.3. Allow the user to have their BMR calculated using the Revised Harris-Benedict Equation, the Revised Harris-Benedict Equation will be implemented.
Added by PBardy

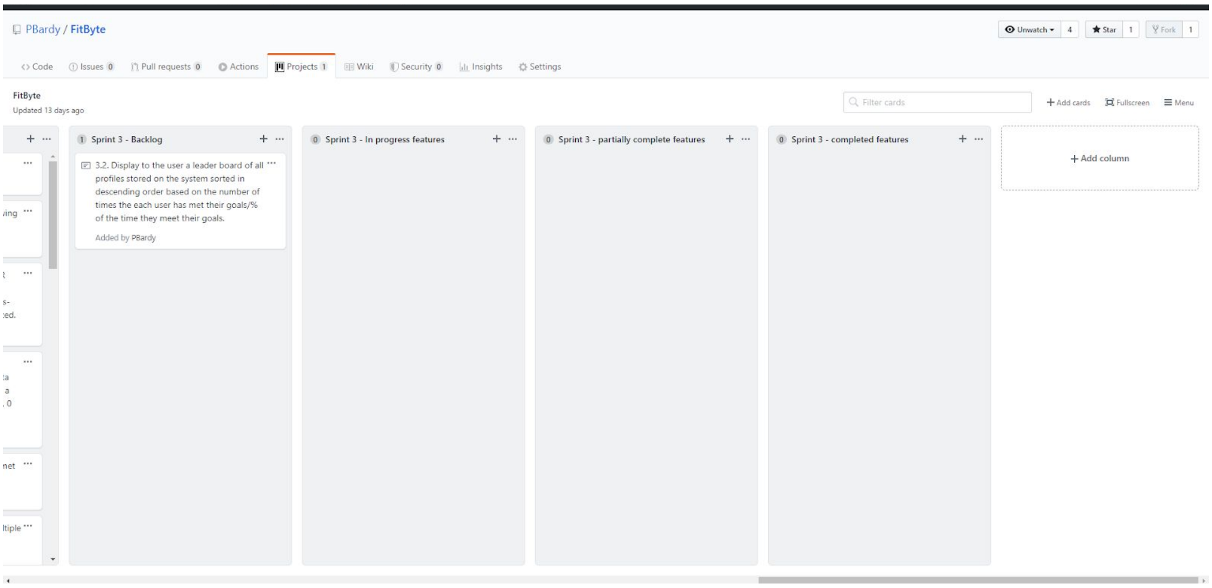
3.5.2. If "2" is entered from Compare calculated daily goal to database data and store whether the goal is met in a local variable as a Boolean 1 for met, 0 for not met.
Added by PBardy

3.5. Display to the user if they have met their goal for the day.
Added by PBardy

4.1. The user can select between multiple data health data options to input.
Added by PBardy

(30/04/2020)

10.5.9 - Project Board, Sprint 3, Start and End



1 Sprint 3 - Backlog

3.2. Display to the user a leader board of all profiles stored on the system sorted in descending order based on the number of times the each user has met their goals/% of the time they meet their goals.
Added by PBardy

0 Sprint 3 - In progress features

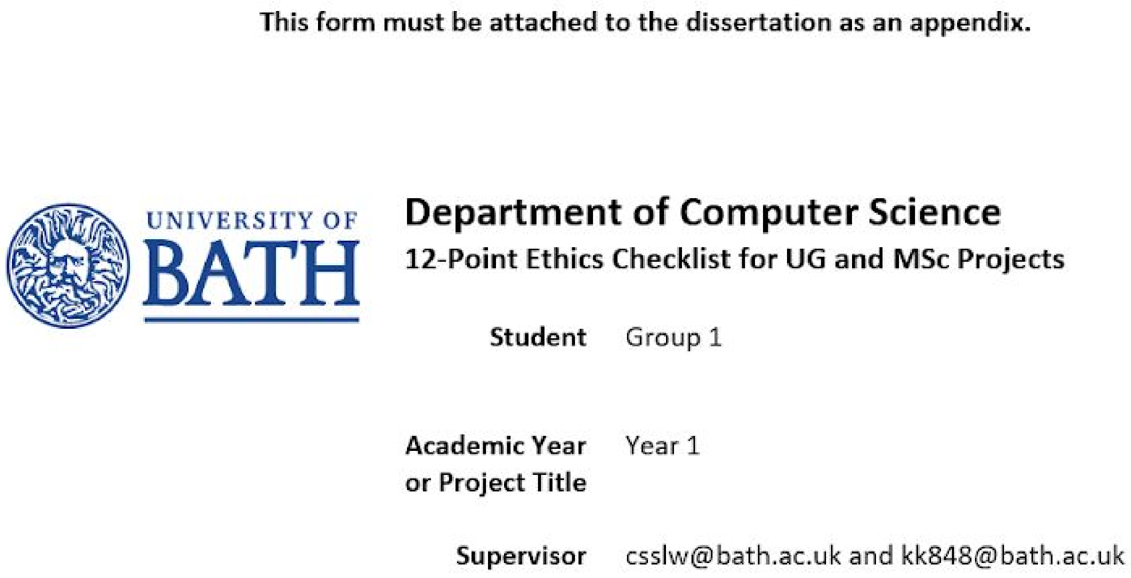
0 Sprint 3 - partially complete features

0 Sprint 3 - completed features

+ Add column

(05/04/2020)

10.6 - Ethics Checklist



This form must be attached to the dissertation as an appendix.

Department of Computer Science

12-Point Ethics Checklist for UG and MSc Projects

Student Group 1

Academic Year or Project Title Year 1

Supervisor csslw@bath.ac.uk and kk848@bath.ac.uk

Does your project involve people for the collection of data other than you and your supervisor(s)? YES

If the answer to the previous question is YES, you need to answer the following questions, otherwise you can ignore them.

This document describes the 12 issues that need to be considered carefully before students or staff involve other people ('participants' or 'volunteers') for the collection of information as part of their project or research. Each answer must be affirmative. Replace the text beneath each question with a statement of how you address the issue in your project.

1. *Have you prepared a briefing script for volunteers?* YES

   We prepared a briefing script informing volunteers that we were conducting a survey seeking to find out the likelihood a university student would want to use our software system and what kind of features they expected. We are also informing them the responses will be anonymous.

2. *Will the participants be informed that they could withdraw at any time?* YES

   All responses to our survey are anonymous, participants can decide not to submit their response after answering all questions.

3. *Is there any intentional deception of the participants?* NO

No participants are to be intentionally deceived. The intention for our survey and how we will use our responses will be clearly communicated to them.

4. *Will participants be de-briefed?* YES

All participants are de-briefed to understand that the purpose of our survey is to help us collect requirements for our personal informatics software system.

5. *Will participants voluntarily give informed consent?* YES

Participants are informed that by submitting a response to the online survey, they have consented for their response to be used in our research.

6. *Will the participants be exposed to any risks greater than those encountered in their normal work life (e.g., through the use of non-standard equipment)?* NO

Participants are simply filling out an online form. They will not be exposed to any great risks.

7. *Are you offering any incentive to the participants?* NO

Participants are submitting responses according to their own will, with no incentive.

8. *Are you in a position of authority or influence over any of your participants?* NO

No, participants for our survey are fellow university students.

9. *Are any of your participants under the age of 16?* NO

No participant is under the age of 16. Our survey has only been sent to university students who are at least 18 years of age.

10. *Do any of your participants have an impairment that will limit Their understanding or communication?* NO

No participant has impairments.

11. *Will the participants be informed of your contact details?* YES

All participants will receive the contact details of the Supervisor and member of the team in charge of the survey.

12. *Do you have a data management plan for all recorded data?* NO

All responses are securely stored on university servers. All responses are completely anonymous. Data will be deleted at an appropriate time.